AM437x and AMIC120 ARM® Cortex™-A9 Processors

Technical Reference Manual

Literature Number: SPRUHL7H
April 2014–Revised January 2018

Contents

15 Ethernet Subsystem 2214

16 Universal Serial Bus (USB) 2543

List of Figures

List of Tables

Read This First

About This Manual

Related Documentation From Texas Instruments

For a complete listing of related documentation and development-support tools, visit www.ti.com.

Terms and Abbreviations

Terms and abbreviations will be added to a future version of this document.

For example: Woco - Woco is a read / write type access defined as "Write - One - Change - Only, and a bit with this access type can be modified, but only once for a POR cycle".

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Introduction

1.1 AM437x Family

1.1.1 Device Features

This architecture is configured with different sets of features in different devices. This technical reference manual details all of the features available in current and future AM437x devices. Some features may not be available or supported in your particular device. The features supported across different AM437x devices are detailed in your device-specific data manual.

1.1.2 Device Identification

Several registers help identify the type and available features of the device. The DEV_FEATURE register in the control module is summarized in the Device Comparison section of your device-specific data manual, and Table 1-1 summarizes the Device_ID registers.

Table 1-1. Device Identification Registers

Bit	Field	Value	Description
31-28	DEVREV		Device revision 0001b - Silicon Revision 1.1 0010b - Silicon Revision 1.2 See device errata for detailed information on functionality in each device revision. Reset value is revision-dependent.
27-12	PARTNUM		Device part number (unique JTAG) 0xB98C
11-1	MFGR		Manufacturer's ID 0x017
0	ID_LSB		Reserved - always 1.

Chapter 2

SPRUHL7H–April 2014–Revised January 2018

Memory Map

This sections describes the memory map for the device.

Topic **Page**

2.1 ARM Cortex-A9 Memory Map

2.1.1 L3 Memory Map

Table 2-1. L3 Memory Map

Block Name	Start_address (hex)	End_address (hex)	Size	Description
GPMC	0x0000_0000	0x1FFF_FFFF	512MB	8-/16-bit External Memory (Ex/R/W) [1][2]
Reserved	0x2000_0000	0x2FFF_FFFF	256MB	
QSPI	0x3000_0000	0x33FF_FFFF	64MB	QSPI CS0 Maddrspace 1 space
Reserved	0x3400_0000	0x3FFF_FFFF	192MB	
Reserved	0x4000_0000	0x4002_FFFF	192KB	
MPU_ROM_PUBLIC	0x4003_0000	0x4003_FFFF	64KB	32-bit Ex/R [2] – Public Boot ROM
Reserved	0x4004_0000	0x400F_FFFF	768KB	
Reserved	0x4010_0000	0x401F_FFFF	1MB	
Reserved	0x4020_0000	0x402E_FFFF	960KB	
MPU_RAM	0x402F_0000	0x402F_FFFF	64KB	32-bit Ex/R/W[2] – SRAM [3]
OCMCRAM	0x4030_0000	0x4033_FFFF	256KB	32-bit Ex/R/W [2] – L3 OCMC SRAM
Reserved	0x4034_0000	0x403F_FFFF	768KB	
Reserved	0x4040_0000	0x4041_FFFF	128KB	
Reserved	0x4042_0000	0x404F_FFFF	896KB	
MPU_L2_CACHE	0x4050_0000	0x4053_FFFF	256KB	MPU L2 Cache
Reserved	0x4054_0000	0x405F_FFFF	768KB	
Reserved	0x4060_0000	0x407F_FFFF	2MB	
Reserved	0x4080_0000	0x4083_FFFF	256KB	
Reserved	0x4084_0000	0x40DF_FFFF	5888KB	
Reserved	0x40E0_0000	0x40E0_7FFF	32KB	
Reserved	0x40E0_8000	0x40EF_FFFF	992KB	
Reserved	0x40F0_0000	0x40F0_7FFF	32KB	
Reserved	0x40F0_8000	0x40FF_FFFF	992KB	
Reserved	0x4100_0000	0x41FF_FFFF	16MB	
Reserved	0x4200_0000	0x43FF_FFFF	32MB	
L3F_CFG	0x4400_0000	0x443F_FFFF	4MB	L3Fast configuration registers
Reserved	0x4440_0000	0x447F_FFFF	4MB	
L3S_CFG	0x4480_0000	0x44BF_FFFF	4MB	L3Slow configuration registers
L4_WKUP	0x44C0_0000	0x44FF_FFFF	4MB	L4 Wakeup Peripheral (see L4_WKUP table)
Reserved	0x4500_0000	0x45FF_FFFF	16MB	
MCASP0_DATA	0x4600_0000	0x463F_FFFF	4MB	McASP0 Data Registers
MCASP1_DATA	0x4640_0000	0x467F_FFFF	4MB	McASP1 Data Registers
Reserved	0x4680_0000	0x46BF_FFFF	4MB	
Reserved	0x46C0_0000	0x46FF_FFFF	4MB	
Reserved	0x4700_0000	0x473F_FFFF	4MB	
Reserved	0x4740_0000	0x477F_FFFF	4MB	
Reserved	0x4780_0000	0x4780_FFFF	64KB	
MMCSD2	0x4781_0000	0x4781_FFFF	64KB	MMCSD2
Reserved	0x4782_0000	0x478F_FFFF	896KB	
QSPI	0x4790_0000	0x479F_FFFF	1MB	QSPI MMR Maddrspace 0 space
Reserved	0x47A0_0000	0x47BF_FFFF	2MB	
Reserved	0x47C0_0000	0x47FF_FFFF	4MB	
L4_PER	0x4800_0000	0x48FF_FFFF	16MB	L4 Peripheral (see L4_PER table)

Table 2-1. L3 Memory Map (continued)

Block Name	Start_address (hex)	End_address (hex)	Size	Description
EDMA3CC	0x4900_0000	0x490F_FFFF	1MB	EDMA3 Channel Controller (TPCC) Registers
Reserved	0x4910_0000	0x497F_FFFF	7MB	
EDMA3TC0	0x4980_0000	0x498F_FFFF	1MB	EDMA3 Transfer Controller 0 (TPTC0) Registers
EDMA3TC1	0x4990_0000	0x499F_FFFF	1MB	EDMA3 Transfer Controller 1 (TPTC1) Registers
EDMA3TC2	0x49A0_0000	0x49AF_FFFF	1MB	EDMA3 Transfer Controller 2 (TPTC2) Registers
Reserved	0x49B0_0000	0x49BF_FFFF	1MB	
Reserved	0x49C0_0000	0x49FF_FFFF	4MB	
L4_FAST	0x4A00_0000	0x4AFF_FFFF	16MB	L4 Fast Peripheral (see L4_FAST table)
DEBUGSS	0x4B00_0000	0x4BFF_FFFF	16MB	Debug Subsystem region
EMIF	0x4C00_0000	0x4CFF_FFFF	16MB	EMIF0 Configuration registers
Reserved	0x4D00_0000	0x4DFF_FFFF	16MB	
Reserved	0x4E00_0000	0x4FFF_FFFF	32MB	
GPMC	0x5000_0000	0x50FF_FFFF	16MB	GPMC Configuration registers
Reserved	0x5100_0000	0x51FF_FFFF	16MB	
Reserved	0x5200_0000	0x52FF_FFFF	16MB	
Reserved	0x5300_0000	0x530F_FFFF	1MB	
Reserved	0x5310_0000	0x531F_FFFF	1MB	
Reserved	0x5320_0000	0x533F_FFFF	2MB	
Reserved	0x5340_0000	0x534F_FFFF	1MB	
Reserved	0x5350_0000	0x535F_FFFF	1MB	
Reserved	0x5360_0000	0x536F_FFFF	1MB	
Reserved	0x5370_0000	0x537F_FFFF	1MB	
Reserved	0x5380_0000	0x543F_FFFF	12MB	
PRU_ICSS1	0x5440_0000	0x547F_FFFF	4MB	PRU-ICSS1 Instruction/Data/Control Space [4]
ADC1	0x5480_0000	0x54BF_FFFF	4MB	ADC1 DMA Port
ADC0	0x54C0_0000	0x54FF_FFFF	4MB	ADC0 DMA Port
Reserved	0x5500_0000	0x55FF_FFFF	16MB	
GFX	0x5600_0000	0x56FF_FFFF	16MB	SGX530 Slave Port
Reserved	0x5700_0000	0x57FF_FFFF	16MB	
Reserved	0x5800_0000	0x58FF_FFFF	16MB	
Reserved	0x5900_0000	0x59FF_FFFF	16MB	
Reserved	0x5A00_0000	0x5AFF_FFFF	16MB	
Reserved	0x5B00_0000	0x5BFF_FFFF	16MB	
Reserved	0x5C00_0000	0x5DFF_FFFF	32MB	
Reserved	0x5E00_0000	0x5FFF_FFFF	32MB	
Reserved	0x6000_0000	0x7FFF_FFFF	512MB	
EMIF	0x8000_0000	0xFFFF_FFFF	2048MB	8-/16-/32-bit External Memory (Ex/R/W) [2]
Reserved	0x1_0000_0000	0x1_FFFF_FFFF	4096MB	

(1) The first 1MB of address space 0x0-0xFFFFF is inaccessible externally.

(2) Ex/R/W – Execute/Read/Write.

(3) Address 0x402F_0000-0x402F_03FF is not available on general purpose (GP) devices.

(4) For PRU-ICSS0/1, the PRU can access the other PRU-ICSS memory space through internal expansion ports. The PRU can access the neighbor PRU-ICSS memory starting at 256KB/0x0004_0000 range. The address seen by the 2nd PRU-ICSS will get translated by hardware logic in PRU-ICSS, 0x0004_0000 will get subtracted.

2.1.2 L4_WKUP Memory Map

Table 2-2. L4_WKUP Memory Map

Region Name	Start_address (hex)	End_address (hex)	Size	Description
Reserved	0x0000_0000	0x44BF_FFFF	1100MB	
L4_WKUP	0x44C0_0000	0x44C0_07FF	2KB	Address/Protection (AP)
L4_WKUP	0x44C0_0800	0x44C0_0FFF	2KB	Link Agent (LA)
L4_WKUP_REG	0x44C0_1000	0x44C0_13FF	1KB	Initiator Port (IP0)
L4_WKUP_REG	0x44C0_1400	0x44C0_17FF	1KB	Initiator Port (IP1)
Reserved	0x44C0_1800	0x44C0_1FFF	2KB	
Reserved	0x44C0_2000	0x44CF_FFFF	1016KB	
Reserved	0x44D0_0000	0x44D0_3FFF	16KB	
Reserved	0x44D0_4000	0x44D0_4FFF	4KB	
Reserved	0x44D0_5000	0x44D7_FFFF	492KB	
Reserved	0x44D8_0000	0x44D8_1FFF	8KB	
Reserved	0x44D8_2000	0x44D8_2FFF	4KB	
Reserved	0x44D8_3000	0x44DE_FFFF	436KB	
PRCM	0x44DF_0000	0x44DF_FFFF	64KB	Module
PRM_IRQ	0x44DF_0000	0x44DF_02FF		Power Reset Module IRQ Registers
PRM_MPU	0x44DF_0300	0x44DF_03FF		Power Reset Module MPU Registers
PRM_GFX	0x44DF_0400	0x44DF_0523		Power Reset Module Graphics Controller Registers
PRM_RTC	0x44DF_0524	0x44DF_0623		Power Reset Module RTC Registers
Reserved	0x44DF_0624	0x44DF_06FF		
PRM_CEFUSE	0x44DF_0700	0x44DF_07FF		Power Reset Module Efuse Registers
PRM_PER	0x44DF_0800	0x44DF_1FFF		Power Reset Module Peripheral Registers
PRM_WKUP	0x44DF_2000	0x44DF_27FF		Power Reset Module Wakeup Registers
CM_WKUP	0x44DF_2800	0x44DF_3FFF		Clock Module Wakeup Registers
PRM_DEVICE	0x44DF_4000	0x44DF_40FF		Power Reset Module Device Registers
CM_DEVICE	0x44DF_4100	0x44DF_41FF		Clock Module Device Registers
CM_DPLL	0x44DF_4200	0x44DF_82FF		Clock Module PLL Registers
CM_MPU	0x44DF_8300	0x44DF_83FF		Clock Module MPU Registers
CM_GFX	0x44DF_8400	0x44DF_84FF		Clock Module Graphics Controller Registers
CM_RTC	0x44DF_8500	0x44DF_85FF		Clock Module RTC Registers
Reserved	0x44DF_8600	0x44DF_86FF		
CM_CEFUSE	0x44DF_8700	0x44DF_87FF		Clock Module Efuse Registers
CM_PER	0x44DF_8800	0x44DF_FFFF		Clock Module Peripheral Registers
Reserved	0x44E0_0000	0x44E0_0FFF	4KB	
Reserved	0x44E0_1000	0x44E0_2FFF	8KB	
Reserved	0x44E0_3000	0x44E0_3FFF	4KB	
Reserved	0x44E0_4000	0x44E0_4FFF	4KB	
DMTIMER0	0x44E0_5000	0x44E0_5FFF	4KB	DMTimer0 Registers
Reserved	0x44E0_6000	0x44E0_6FFF	4KB	
GPIO0	0x44E0_7000	0x44E0_7FFF	4KB	GPIO0 Registers
Reserved	0x44E0_8000	0x44E0_8FFF	4KB	
UART0	0x44E0_9000	0x44E0_9FFF	4KB	UART0 Registers
Reserved	0x44E0_A000	0x44E0_AFFF	4KB	
I2C0	0x44E0_B000	0x44E0_BFFF	4KB	I2C0 Registers
Reserved	0x44E0_C000	0x44E0_CFFF	4KB	
ADC0	0x44E0_D000	0x44E0_DFFF	4KB	ADC0 Registers

Table 2-2. L4_WKUP Memory Map (continued)

Region Name	Start_address (hex)	End_address (hex)	Size	Description
Reserved	0x44E0_E000	0x44E0_EFFF	4KB	
Reserved	0x44E0_F000	0x44E0_FFFF	4KB	
CONTROL_MODULE	0x44E1_0000	0x44E1_FFFF	64KB	Control Module Registers
Reserved	0x44E2_0000	0x44E2_FFFF	64KB	
Reserved	0x44E3_0000	0x44E3_0FFF	4KB	
DMTIMER1_1MS	0x44E3_1000	0x44E3_1FFF	4KB	DMTimer1_1ms (accurate 1ms timer) Registers
Reserved	0x44E3_2000	0x44E3_2FFF	4KB	
Reserved	0x44E3_3000	0x44E3_3FFF	4KB	
Reserved	0x44E3_4000	0x44E3_4FFF	4KB	
WDT1	0x44E3_5000	0x44E3_5FFF	4KB	Watchdog Timer1 Registers
Reserved	0x44E3_6000	0x44E3_6FFF	4KB	
Reserved	0x44E3_7000	0x44E3_7FFF	4KB	
Reserved	0x44E3_8000	0x44E3_8FFF	4KB	
Reserved	0x44E3_9000	0x44E3_9FFF	4KB	
Reserved	0x44E3_A000	0x44E3_AFFF	4KB	
Reserved	0x44E3_B000	0x44E3_DFFF	12KB	
RTCSS	0x44E3_E000	0x44E3_EFFF	4KB	RTC Registers
Reserved	0x44E3_F000	0x44E3_FFFF	4KB	
DEBUGSS	0x44E4_0000	0x44E7_FFFF	256KB	Debug Registers
Reserved	0x44E8_0000	0x44E8_0FFF	4KB	
Reserved	0x44E8_1000	0x44E8_1FFF	4KB	
Reserved	0x44E8_2000	0x44E8_3FFF	8KB	
Reserved	0x44E8_4000	0x44E8_4FFF	4KB	
Reserved	0x44E8_5000	0x44E8_5FFF	4KB	
SYNCTIMER	0x44E8_6000	0x44E8_6FFF	4KB	SyncTimer Registers
Reserved	0x44E8_7000	0x44E8_7FFF	4KB	
Reserved	0x44E8_8000	0x44E8_FFFF	32KB	
Reserved	0x44E9_0000	0x44E9_0FFF	4KB	
Reserved	0x44E9_1000	0x44E9_1FFF	4KB	
Reserved	0x44E9_2000	0x44E9_2FFF	4KB	
Reserved	0x44E9_3000	0x44E9_FFFF	52KB	
Reserved	0x44F0_0000	0x44FF_FFFF	1MB	
Reserved	0x4500_0000	0xFFFF_FFFF	2992MB	

2.1.3 L4_PER Peripheral Memory Map

Table 2-3. L4_PER Peripheral Memory Map

Device Name	Start_address (hex)	End_address (hex)	Size	Description
L4_PER	0x4800_0000	0x4800_07FF	2KB	Address/Protection (AP)
L4_PER	0x4800_0800	0x4800_0FFF	2KB	Link Agent (LA)
L4_PER_REG	0x4800_1000	0x4800_13FF	1KB	Initiator Port (IP0)
L4_PER_REG	0x4800_1400	0x4800_17FF	1KB	Initiator Port (IP1)
L4_PER_REG	0x4800_1800	0x4800_1BFF	1KB	Initiator Port (IP2)
L4_PER_REG	0x4800_1C00	0x4800_1FFF	1KB	Initiator Port (IP3)
Reserved	0x4800_2000	0x4800_3FFF	8KB	

Table 2-3. L4_PER Peripheral Memory Map (continued)

Device Name	Start_address (hex)	End_address (hex)	Size	Description
Reserved	0x4800_4000	0x4800_7FFF	16KB	
Reserved	0x4800_8000	0x4800_8FFF	4KB	
Reserved	0x4800_9000	0x4800_9FFF	4KB	
Reserved	0x4800_A000	0x4800_FFFF	24KB	
Reserved	0x4801_0000	0x4801_0FFF	4KB	
Reserved	0x4801_1000	0x4801_1FFF	4KB	
Reserved	0x4801_2000	0x4801_FFFF	56KB	
Reserved	0x4802_0000	0x4802_0FFF	4KB	
Reserved	0x4802_1000	0x4802_1FFF	4KB	
UART1	0x4802_2000	0x4802_2FFF	4KB	UART1 Registers
Reserved	0x4802_3000	0x4802_3FFF	4KB	
UART2	0x4802_4000	0x4802_4FFF	4KB	UART2 Registers
Reserved	0x4802_5000	0x4802_5FFF	4KB	
Reserved	0x4802_6000	0x4802_7FFF	8KB	
Reserved	0x4802_8000	0x4802_8FFF	4KB	
Reserved	0x4802_9000	0x4802_9FFF	4KB	
I2C1	0x4802_A000	0x4802_AFFF	4KB	I2C1 Registers
Reserved	0x4802_B000	0x4802_BFFF	4KB	
Reserved	0x4802_C000	0x4802_CFFF	4KB	
Reserved	0x4802_D000	0x4802_DFFF	4KB	
Reserved	0x4802_E000	0x4802_EFFF	4KB	
Reserved	0x4802_F000	0x4802_FFFF	4KB	
MCSPI0	0x4803_0000	0x4803_0FFF	4KB	McSPI0 Registers
Reserved	0x4803_1000	0x4803_1FFF	4KB	
Reserved	0x4803_2000	0x4803_2FFF	4KB	
Reserved	0x4803_3000	0x4803_3FFF	4KB	
Reserved	0x4803_4000	0x4803_4FFF	4KB	
Reserved	0x4803_5000	0x4803_5FFF	4KB	
Reserved	0x4803_6000	0x4803_6FFF	4KB	
Reserved	0x4803_7000	0x4803_7FFF	4KB	
MCASP0_CFG	0x4803_8000	0x4803_9FFF	8KB	McASP0 CFG Registers
Reserved	0x4803_A000	0x4803_AFFF	4KB	
Reserved	0x4803_B000	0x4803_BFFF	4KB	
MCASP1_CFG	0x4803_C000	0x4803_DFFF	8KB	McASP1 CFG Registers
Reserved	0x4803_E000	0x4803_EFFF	4KB	
Reserved	0x4803_F000	0x4803_FFFF	4KB	
DMTIMER2	0x4804_0000	0x4804_0FFF	4KB	DMTimer2 Registers
Reserved	0x4804_1000	0x4804_1FFF	4KB	
DMTIMER3	0x4804_2000	0x4804_2FFF	4KB	DMTimer3 Registers
Reserved	0x4804_3000	0x4804_3FFF	4KB	
DMTIMER4	0x4804_4000	0x4804_4FFF	4KB	DMTimer4 Registers
Reserved	0x4804_5000	0x4804_5FFF	4KB	
DMTIMER5	0x4804_6000	0x4804_6FFF	4KB	DMTimer5 Registers
Reserved	0x4804_7000	0x4804_7FFF	4KB	
DMTIMER6	0x4804_8000	0x4804_8FFF	4KB	DMTimer6 Registers
Reserved	0x4804_9000	0x4804_9FFF	4KB	
DMTIMER7	0x4804_A000	0x4804_AFFF	4KB	DMTimer7 Registers

Table 2-3. L4_PER Peripheral Memory Map (continued)

Device Name	Start_address (hex)	End_address (hex)	Size	Description
Reserved	0x4804_B000	0x4804_BFFF	4KB	
GPIO1	0x4804_C000	0x4804_CFFF	4KB	GPIO1 Registers
Reserved	0x4804_D000	0x4804_DFFF	4KB	
Reserved	0x4804_E000	0x4804_FFFF	8KB	
Reserved	0x4805_0000	0x4805_1FFF	8KB	
Reserved	0x4805_2000	0x4805_2FFF	4KB	
Reserved	0x4805_3000	0x4805_FFFF	52KB	
MMCSD0	0x4806_0000	0x4806_0FFF	4KB	MMCSD0 Registers
Reserved	0x4806_1000	0x4806_1FFF	4KB	
Reserved	0x4806_2000	0x4807_FFFF	120KB	
ELM	0x4808_0000	0x4808_FFFF	64KB	ELM Registers
Reserved	0x4809_0000	0x4809_0FFF	4KB	
Reserved	0x4809_1000	0x4809_FFFF	60KB	
Reserved	0x480A_0000	0x480A_FFFF	64KB	
Reserved	0x480B_0000	0x480B_0FFF	4KB	
Reserved	0x480B_1000	0x480B_FFFF	60KB	
Reserved	0x480C_0000	0x480C_0FFF	4KB	
Reserved	0x480C_1000	0x480C_1FFF	4KB	
Reserved	0x480C_2000	0x480C_2FFF	4KB	
Reserved	0x480C_3000	0x480C_3FFF	4KB	
Reserved	0x480C_4000	0x480C_7FFF	16KB	
MAILBOX0	0x480C_8000	0x480C_8FFF	4KB	Mailbox Registers
Reserved	0x480C_9000	0x480C_9FFF	4KB	
SPINLOCK	0x480C_A000	0x480C_AFFF	4KB	Spinlock Registers
Reserved	0x480C_B000	0x480C_BFFF	4KB	
Reserved	0x480C_C000	0x480C_CFFF	4KB	
Reserved	0x480C_D000	0x480C_DFFF	4KB	
Reserved	0x480C_E000	0x480F_FFFF	200KB	
Reserved	0x4810_0000	0x4811_FFFF	128KB	
Reserved	0x4812_0000	0x4812_0FFF	4KB	
Reserved	0x4812_1000	0x4812_1FFF	4KB	
Reserved	0x4812_2000	0x4812_2FFF	4KB	
Reserved	0x4812_3000	0x4812_3FFF	4KB	
Reserved	0x4812_4000	0x4813_FFFF	112KB	
Reserved	0x4814_0000	0x4815_FFFF	128KB	
Reserved	0x4816_0000	0x4816_0FFF	4KB	
Reserved	0x4816_1000	0x4817_FFFF	124KB	
Reserved	0x4818_0000	0x4818_2FFF	12KB	
Reserved	0x4818_3000	0x4818_3FFF	4KB	
Reserved	0x4818_4000	0x4818_7FFF	16KB	
Reserved	0x4818_8000	0x4818_8FFF	4KB	
Reserved	0x4818_9000	0x4818_9FFF	4KB	
Reserved	0x4818_A000	0x4818_AFFF	4KB	
Reserved	0x4818_B000	0x4818_BFFF	4KB	
OCP_WP_NOC	0x4818_C000	0x4818_CFFF	4KB	OCP Watchpoint Registers
Reserved	0x4818_D000	0x4818_DFFF	4KB	
Reserved	0x4818_E000	0x4818_EFFF	4KB	

Table 2-3. L4_PER Peripheral Memory Map (continued)

Device Name	Start_address (hex)	End_address (hex)	Size	Description
Reserved	0x4818_F000	0x4818_FFFF	4KB	
Reserved	0x4819_0000	0x4819_0FFF	4KB	
Reserved	0x4819_1000	0x4819_1FFF	4KB	
Reserved	0x4819_2000	0x4819_2FFF	4KB	
Reserved	0x4819_3000	0x4819_3FFF	4KB	
Reserved	0x4819_4000	0x4819_BFFF	32KB	
I2C2	0x4819_C000	0x4819_CFFF	4KB	I2C2 Registers
Reserved	0x4819_D000	0x4819_DFFF	4KB	
Reserved	0x4819_E000	0x4819_EFFF	4KB	
Reserved	0x4819_F000	0x4819_FFFF	4KB	
MCSPI1	0x481A_0000	0x481A_0FFF	4KB	McSPI1 Registers
Reserved	0x481A_1000	0x481A_1FFF	4KB	
MCSPI2	0x481A_2000	0x481A_2FFF	4KB	McSPI2 Registers
Reserved	0x481A_3000	0x481A_3FFF	4KB	
MCSPI3	0x481A_4000	0x481A_4FFF	4KB	McSPI3 Registers
Reserved	0x481A_5000	0x481A_5FFF	4KB	
UART3	0x481A_6000	0x481A_6FFF	4KB	UART3 Registers
Reserved	0x481A_7000	0x481A_7FFF	4KB	
UART4	0x481A_8000	0x481A_8FFF	4KB	UART4 Registers
Reserved	0x481A_9000	0x481A_9FFF	4KB	
UART5	0x481A_A000	0x481A_AFFF	4KB	UART5 Registers
Reserved	0x481A_B000	0x481A_BFFF	4KB	
GPIO2	0x481A_C000	0x481A_CFFF	4KB	GPIO2 Registers
Reserved	0x481A_D000	0x481A_DFFF	4KB	
GPIO3	0x481A_E000	0x481A_EFFF	4KB	GPIO3 Registers
Reserved	0x481A_F000	0x481A_FFFF	4KB	
Reserved	0x481B_0000	0x481B_FFFF	64KB	
Reserved	0x481C_0000	0x481C_0FFF	4KB	
DMTIMER8	0x481C_1000	0x481C_1FFF	4KB	DMTimer8 Registers
Reserved	0x481C_2000	0x481C_2FFF	4KB	
Reserved	0x481C_3000	0x481C_9FFF	28KB	
Reserved	0x481C_A000	0x481C_AFFF	4KB	
Reserved	0x481C_B000	0x481C_BFFF	4KB	
DCAN0	0x481C_C000	0x481C_DFFF	8KB	DCAN0 Registers
Reserved	0x481C_E000	0x481C_FFFF	8KB	
DCAN1	0x481D_0000	0x481D_1FFF	8KB	DCAN1 Registers
Reserved	0x481D_2000	0x481D_3FFF	8KB	
Reserved	0x481D_4000	0x481D_4FFF	4KB	
Reserved	0x481D_5000	0x481D_5FFF	4KB	
Reserved	0x481D_6000	0x481D_6FFF	4KB	
Reserved	0x481D_7000	0x481D_7FFF	4KB	
MMCSD1	0x481D_8000	0x481D_8FFF	4KB	MMCSD1 Registers
Reserved	0x481D_9000	0x481D_9FFF	4KB	
Reserved	0x481D_A000	0x481F_FFFF	152KB	
Reserved	0x4820_0000	0x4820_0FFF	4KB	
Reserved	0x4820_1000	0x4823_FFFF	252KB	
MPU_SCU	0x4824_0000	0x4824_00FF	256B	MPU SCU Registers

Table 2-3. L4_PER Peripheral Memory Map (continued)

Device Name	Start_address (hex)	End_address (hex)	Size	Description
MPU_INTC	0x4824_0100	0x4824_01FF	256B	MPU Interrupt Controller Interfaces
MPU_GBL_TIMER	0x4824_0200	0x4824_02FF	256B	MPU Global Timer
Reserved	0x4824_0300	0x4824_05FF	768B	
MPU_PRV_TIMERS	0x4824_0600	0x4824_06FF	256B	MPU Private Timers and Watchdog
Reserved	0x4824_0700	0x4824_0FFF	2304B	
MPU_INT_DIST	0x4824_1000	0x4824_1FFF	4KB	MPU Interrupt Distributor
MPU_PL310	0x4824_2000	0x4824_2FFF	4KB	MPU PL310 Programming Registers
Reserved	0x4824_3000	0x4824_3FFF	4KB	
Reserved	0x4824_4000	0x4827_FFFF	240KB	
Reserved	0x4828_0000	0x4828_0FFF	4KB	
MPU_WAKEUP_GEN	0x4828_1000	0x4828_1FFF	4KB	MPU Wakeup Generator
Reserved	0x4828_2000	0x4828_FFFF	56KB	
Reserved	0x4829_0000	0x4829_FFFF	64KB	
MPU_AXI2OCP	0x482A_0000	0x482A_FFFF	64KB	MPU AXI2OCP Registers
Reserved	0x482B_0000	0x482F_FFFF	320KB	
PWMSS0	0x4830_0000	0x4830_00FF	256B	PWMSS0 Configuration Registers
PWMSS0_ECAP	0x4830_0100	0x4830_017F	128B	PWMSS eCAP0 Registers
PWMSS0_EQEP	0x4830_0180	0x4830_01FF	128B	PWMSS eQEP0 Registers
PWMSS0_EPWM	0x4830_0200	0x4830_025F	96B	PWMSS ePWM0 Registers
Reserved	0x4830_0260	0x4830_0FFF	3488B	
Reserved	0x4830_1000	0x4830_1FFF	4KB	
PWMSS1	0x4830_2000	0x4830_20FF	256B	PWMSS1 Configuration Registers
PWMSS1_ECAP	0x4830_2100	0x4830_217F	128B	PWMSS eCAP1 Registers
PWMSS1_EQEP	0x4830_2180	0x4830_21FF	128B	PWMSS eQEP1 Registers
PWMSS1_EPWM	0x4830_2200	0x4830_225F	96B	PWMSS ePWM1 Registers
Reserved	0x4830_2260	0x4830_2FFF	3488B	
Reserved	0x4830_3000	0x4830_3FFF	4KB	
PWMSS2	0x4830_4000	0x4830_40FF	256B	PWMSS2 Configuration Registers
PWMSS2_ECAP	0x4830_4100	0x4830_417F	128B	PWMSS eCAP2 Registers
PWMSS2_EQEP	0x4830_4180	0x4830_41FF	128B	PWMSS eQEP2 Registers
PWMSS2_EPWM	0x4830_4200	0x4830_425F	96B	PWMSS ePWM2 Registers
Reserved	0x4830_4260	0x4830_4FFF	3488B	
Reserved	0x4830_5000	0x4830_5FFF	4KB	
PWMSS3	0x4830_6000	0x4830_60FF	256B	PWMSS3 Configuration Registers
Reserved	0x4830_6100	0x4830_61FF	256B	
PWMSS3_EPWM	0x4830_6200	0x4830_625F	96B	PWMSS ePWM3 Registers
Reserved	0x4830_6260	0x4830_6FFF	3488B	
Reserved	0x4830_7000	0x4830_7FFF	4KB	
PWMSS4	0x4830_8000	0x4830_80FF	256B	PWMSS4 Configuration Registers
Reserved	0x4830_8100	0x4830_81FF	256B	
PWMSS4_EPWM	0x4830_8200	0x4830_825F	96B	PWMSS ePWM4 Registers
Reserved	0x4830_8260	0x4830_8FFF	3488B	
Reserved	0x4830_9000	0x4830_9FFF	4KB	
PWMSS5	0x4830_A000	0x4830_A0FF	256B	PWMSS5 Configuration Registers
Reserved	0x4830_A100	0x4830_A1FF	256B	
PWMSS5_EPWM	0x4830_A200	0x4830_A25F	96B	PWMSS ePWM5 Registers
Reserved	0x4830_A260	0x4830_AFFF	3488B	

Table 2-3. L4_PER Peripheral Memory Map (continued)

Device Name	Start_address (hex)	End_address (hex)	Size	Description
Reserved	0x4830_B000	0x4830_BFFF	4KB	
Reserved	0x4830_C000	0x4830_DFFF	8KB	
Reserved	0x4830_E000	0x4830_EFFF	4KB	
Reserved	0x4830_F000	0x4830_FFFF	4KB	
Reserved	0x4831_0000	0x4831_1FFF	8KB	
Reserved	0x4831_2000	0x4831_2FFF	4KB	
Reserved	0x4831_3000	0x4831_3FFF	4KB	
Reserved	0x4831_4000	0x4831_4FFF	4KB	
Reserved	0x4831_5000	0x4831_7FFF	12KB	
Reserved	0x4831_8000	0x4831_BFFF	16KB	
Reserved	0x4831_C000	0x4831_CFFF	4KB	
Reserved	0x4831_D000	0x4831_FFFF	12KB	
GPIO4	0x4832_0000	0x4832_0FFF	4KB	GPIO4 Registers
Reserved	0x4832_1000	0x4832_1FFF	4KB	
GPIO5	0x4832_2000	0x4832_2FFF	4KB	GPIO5 Registers
Reserved	0x4832_3000	0x4832_3FFF	4KB	
Reserved	0x4832_4000	0x4832_5FFF	8KB	
VPFE0	0x4832_6000	0x4832_6FFF	4KB	VPFE0 (Camera) Registers
Reserved	0x4832_7000	0x4832_7FFF	4KB	
VPFE1	0x4832_8000	0x4832_8FFF	4KB	VPFE1 (Camera) Registers
Reserved	0x4832_9000	0x4832_9FFF	4KB	
DSS_TOP	0x4832_A000	0x4832_A3FF	1KB	Display Subsystem Top Registers
DSS_DISPC	0x4832_A400	0x4832_A7FF	1KB	Display Controller Registers
DSS_RFBI	0x4832_A800	0x4832_ABFF	1KB	RFBI Registers
Reserved	0x4832_AC00	0x4832_AFFF	1KB	
Reserved	0x4832_B000	0x4832_BFFF	4KB	
Reserved	0x4832_C000	0x4832_CFFF	4KB	
Reserved	0x4832_D000	0x4833_CFFF	64KB	
DMTIMER9	0x4833_D000	0x4833_DFFF	4KB	DMTimer9 Registers
Reserved	0x4833_E000	0x4833_EFFF	4KB	
DMTIMER10	0x4833_F000	0x4833_FFFF	4KB	DMTimer10 Registers
Reserved	0x4834_0000	0x4834_0FFF	4KB	
DMTIMER11	0x4834_1000	0x4834_1FFF	4KB	DMTimer11 Registers
Reserved	0x4834_2000	0x4834_2FFF	4KB	
Reserved	0x4834_3000	0x4834_3FFF	4KB	
Reserved	0x4834_4000	0x4834_4FFF	4KB	
MCSPI4	0x4834_5000	0x4834_5FFF	4KB	McSPI4 Registers
Reserved	0x4834_6000	0x4834_6FFF	4KB	
HDQ1W	0x4834_7000	0x4834_7FFF	4KB	HDQ1W Registers
Reserved	0x4834_8000	0x4834_8FFF	4KB	
Reserved	0x4834_9000	0x4834_AFFF	8KB	
Reserved	0x4834_B000	0x4834_BFFF	4KB	
ADC1	0x4834_C000	0x4834_DFFF	8KB	ADC1 Registers
Reserved	0x4834_E000	0x4834_EFFF	4KB	
Reserved	0x4834_F000	0x4837_FFFF	196KB	
USB0_CONTROL	0x4838_0000	0x4839_FFFF	128KB	USB0 Controller Registers
Reserved	0x483A_0000	0x483A_0FFF	4KB	

Table 2-3. L4_PER Peripheral Memory Map (continued)

Device Name	Start_address (hex)	End_address (hex)	Size	Description
Reserved	0x483A_1000	0x483A_7FFF	28KB	
USB0_PHY	0x483A_8000	0x483A_FFFF	32KB	USB0 PHY Registers
Reserved	0x483B_0000	0x483B_0FFF	4KB	
Reserved	0x483B_1000	0x483B_FFFF	60KB	
USB1_CONTROL	0x483C_0000	0x483D_FFFF	128KB	USB1 Controller Registers
Reserved	0x483E_0000	0x483E_0FFF	4KB	
Reserved	0x483E_1000	0x483E_7FFF	28KB	
USB1_PHY	0x483E_8000	0x483E_FFFF	32KB	USB1 PHY Registers
Reserved	0x483F_0000	0x483F_0FFF	4KB	
Reserved	0x483F_1000	0x483F_1FFF	4KB	
Reserved	0x483F_2000	0x483F_3FFF	8KB	
Reserved	0x483F_4000	0x483F_4FFF	4KB	
Reserved	0x483F_5000	0x483F_FFFF	44KB	
Reserved	0x4840_0000	0x48FF_FFFF	12MB	

2.1.4 L4 Fast Peripheral Memory Map

Table 2-4. L4 Fast Peripheral Memory Map

Device Name	Start_address (hex)	End_address (hex)	Size	Description
L4_FAST	0x4A00_0000	0x4A00_07FF	2KB	Address/Protection(AP)
L4_FAST	0x4A00_0800	0x4A00_0FFF	2KB	Link Agent(LA)
L4_FAST_REG	0x4A00_1000	0x4A00_13FF	1KB	Initiator Port(IPO)
Reserved	0x4A00_1400	0x4A00_17FF	1KB	
Reserved	0x4A00_1800	0x4A00_1FFF	2KB	
Reserved	0x4A00_2000	0x4A07_FFFF	504KB	
Reserved	0x4A08_0000	0x4A09_FFFF	128KB	
Reserved	0x4A0A_0000	0x4A0A_0FFF	4KB	
Reserved	0x4A0A_1000	0x4A0F_FFFF	380KB	
CPSW	0x4A10_0000	0x4A10_7FFF	32KB	Registers
CPSW_PORT	0x4A10_0100	0x4A10_07FF		Ethernet Switch Port Control
CPSW_CPDMA	0x4A10_0800	0x4A10_08FF		CPPI DMA Controller Module
CPSW_STATS	0x4A10_0900	0x4A10_09FF		Ethernet Statistics
CPSW_STATERAM	0x4A10_0A00	0x4A10_0BFF		CPPI DMA State RAM
CPSW_CPTS	0x4A10_0C00	0x4A10_0CFF		Ethernet Time Sync Module
CPSW_ALE	0x4A10_0D00	0x4A10_0D7F		Ethernet Address Lookup Engine
CPSW_SL1	0x4A10_0D80	0x4A10_0DBF		Ethernet Sliver for Port 1
CPSW_SL2	0x4A10_0DC0	0x4A10_0DFF		Ethernet Sliver for Port 2
Reserved	0x4A10_0E00	0x4A10_0FFF		
CPSW_MDIO	0x4A10_1000	0x4A10_10FF		Ethernet MDIO Controller
Reserved	0x4A10_1100	0x4A10_11FF		
CPSW_WR	0x4A10_1200	0x4A10_1FFF		Ethernet Subsystem Wrapper for RMII/RGMII
CPSW_CPPI_RAM	0x4A10_2000	0x4A10_3FFF		Communications Port Programming Interface
Reserved	0x4A10_8000	0x4A10_8FFF	4KB	
Reserved	0x4A10_9000	0x4A13_FFFF	220KB	
Reserved	0x4A14_0000	0x4A14_FFFF	64KB	
Reserved	0x4A15_0000	0x4A15_0FFF	4KB	

Table 2-4. L4 Fast Peripheral Memory Map (continued)

Device Name	Start_address (hex)	End_address (hex)	Size	Description
Reserved	0x4A15_1000	0x4A17_FFFF	188KB	
Reserved	0x4A18_0000	0x4A19_FFFF	128KB	
Reserved	0x4A1A_0000	0x4A1A_0FFF	4KB	
Reserved	0x4A1A_1000	0x4A1A_1FFF	4KB	
Reserved	0x4A1A_2000	0x4A1A_3FFF	8KB	
Reserved	0x4A1A_4000	0x4A1A_4FFF	4KB	
Reserved	0x4A1A_5000	0x4A1A_5FFF	4KB	
Reserved	0x4A1A_6000	0x4A1A_6FFF	4KB	
Reserved	0x4A1A_7000	0x4A1A_7FFF	4KB	
Reserved	0x4A1A_8000	0x4A1A_9FFF	8KB	
Reserved	0x4A1A_A000	0x4A1A_AFFF	4KB	
Reserved	0x4A1A_B000	0x4A1A_BFFF	4KB	
Reserved	0x4A1A_C000	0x4A1A_CFFF	4KB	
Reserved	0x4A1A_D000	0x4A1A_DFFF	4KB	
Reserved	0x4A1A_E000	0x4A1A_FFFF	8KB	
Reserved	0x4A1B_0000	0x4A1B_0FFF	4KB	
Reserved	0x4A1B_1000	0x4A1B_1FFF	4KB	
Reserved	0x4A1B_2000	0x4A1B_2FFF	4KB	
Reserved	0x4A1B_3000	0x4A1B_3FFF	4KB	
Reserved	0x4A1B_4000	0x4A1B_4FFF	4KB	
Reserved	0x4A1B_5000	0x4A1B_5FFF	4KB	
Reserved	0x4A1B_6000	0x4A1B_6FFF	4KB	
Reserved	0x4A1B_7000	0x4A1F_FFFF	292KB	
Reserved	0x4A20_0000	0x4A27_FFFF	512KB	
Reserved	0x4A28_0000	0x4A28_0FFF	4KB	
Reserved	0x4A28_1000	0x4A2F_FFFF	508KB	
Reserved	0x4A30_0000	0x4A37_FFFF	512KB	
Reserved	0x4A38_0000	0x4A38_0FFF	4KB	
Reserved	0x4A38_1000	0x4A3F_FFFF	508KB	
Reserved	0x4A40_0000	0x4A40_1FFF	8KB	
Reserved	0x4A40_2000	0x4A40_2FFF	4KB	
Reserved	0x4A40_3000	0x4AFF_FFFF	12276KB	

ARM MPU Subsystem

This chapter describes the MPU Subsystem for the device.

Topic **Page**

3.1 Introduction

The microprocessor unit (MPU) subsystem of the device handles transactions between the ARM core (ARM Cortex-A9 Processor) and the L3 interconnect.

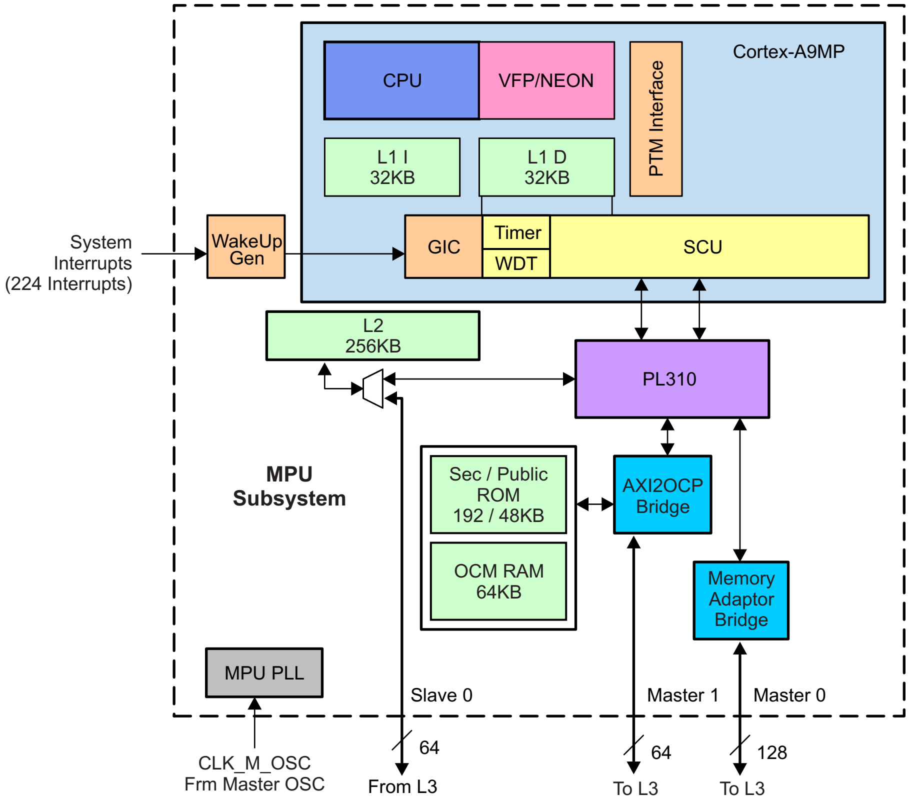
The Cortex-A9 is an ARMv7 compatible, 2-issue, in-order execution pipeline with integrated L1 and L2 caches with a NEON Single Instruction, Multiple Data (SIMD) Media Processing Unit.

The MPU subsystem includes CoreSight compliant logic to allow the debug subsystem access to the Cortex-A8 debug and emulation resources, including the Embedded Trace Macrocell.

The MPU subsystem includes 256KB of L2 cache. The L2 cache is controlled by a PL310. The CPU is configured to have 32KB of instruction cache and 32KB of data cache. The CPU includes a NEON and vector floating point (VFP) (FPU) coprocessors.

Figure 3-1 shows the MPU subsystem top-level block diagram and the power domain partitions.

Figure 3-1. MPU Subsystem Block Diagram



3.1.1 Features

This section outlines the key features of the MPU subsection:

- ARM Microprocessor
  - Cortex-A9 revision R2P10.
  - Symmetric Multi-Processor Architecture (Single CPU Configuration).
  - Superscalar, dynamic multi-issue technology with an efficient 8-stage pipeline.
  - Continuous fetch and decoding of two instructions per clock cycle.
  - Out-of-order (OoO) instruction dispatch and completion.
  - Integrated Cortex-A9 NEON Processing Engine (NPE) to enhance the capabilities of the FPU to include the ARM NEON Advanced SIMD support for accelerated media and signal processing computation.
  - VFPv3-D16 hardware to support single- and double-precision add, subtract, multiply, divide, multiply and accumulate and square root operations.
  - 32KB (L1) instruction and 32KB (L1) data cache with 32B line size and four-way set associative.
  - Memory Management Unit (MMU) with a two-level translation lookaside buffer (TLB) organization.
    - First level is a 32-entry, fully-associative micro-TLB implemented on each of the instruction and data sides.
    - Second level is a unified, two-way associative, 128-entry main TLB with support for a hardware TLB table walk.
  - Snoop Control Unit (SCU) ensures memory coherency in the system between CPU and other masters sharing cached data.
  - Integrated Timer and Watchdog Timer.
  - Integrated symmetric multiprocessing (SMP)-capable generic interrupt controller with 224 shared peripheral interrupts (SPI).
  - Two 64-bit AXI master ports to interface to the L2 controller.
    - Supports multiple outstanding transactions
    - Supports out-of-order data return
- L2 Cache Controller (PL310)
  - 256KB L2 Cache.
  - 16-way set associative.
  - 32B line size.
  - PL310 address filtering function used to split accesses between MA and AXI2OCP
  - Two slave ports
  - Two master ports.
  - Lockdown format C (way locking) for instruction and data.
  - Includes four 256-bit line fill buffers (LFB) shared by the master ports.
  - Each slave port includes two 256-bit line read buffers (LRB).
  - Includes four 256-bit store buffers with merge capability.
  - Support for 64-byte line fills issued to L3.
  - Support for using the 256KB L2 cache as SRAM. SRAM interface is memory mapped to chip level L3.

- Debug and Emulation
  - CPU debug requirements handled through chip-level debug subsystem.
  - Cross trigger interface (CTI) connects to a cross trigger matrix (CTM).
  - Includes a trigger interface to convert the TI trigger format to the ARM CTI format.
  - Program trace macrocell (PTM) to perform real-time instruction flow tracing based on program flow trace (PFT) architecture.
  - Generates trace only at certain points (waypoints) in program execution to reduce the amount of trace data generated.
  - Implemented in the emulation power domain (WKUP voltage domain).
  - A debug bridge connects the external debug OCP port to all the internal APB targets.
- AXI2OCP Bridge
  - Supports OCP 2.2.
  - Connected to PL310 slave port M1.
  - Single request, multiple data protocol on port.
  - Multiple targets, including two OCP ports (64-bit and 32-bit).
- Memory Adaptor
  - Connected to PL310 slave port M0.
  - Standard OCP 2.3 interface to chip-level L3 interconnect.
  - Full-speed interface to the PL310.

3.2 Integration

3.2.1 *Clocking, Reset and Power Management*

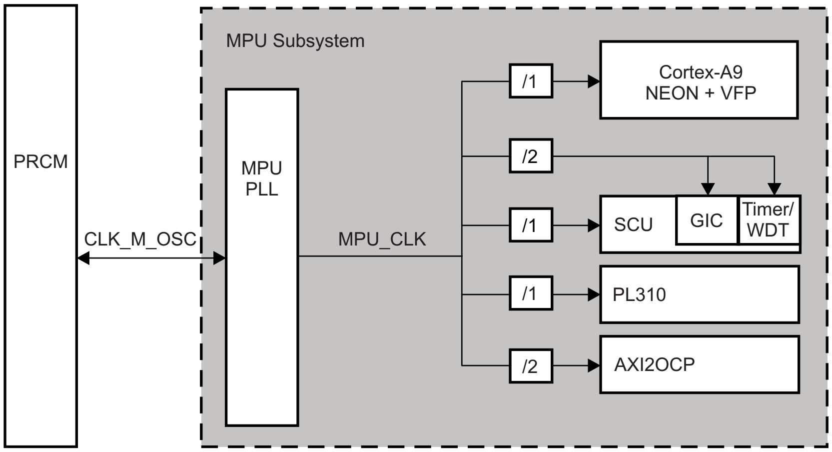
3.2.1.1 Power Management

All power domains are controlled by the global PRCM. The power management supports for debug and emulation are also controlled by the global PRCM.

3.2.1.2 Clocking Management

The MPU PLL generates the MPU clock for the MPU subsystem, as shown in Figure 3-2.

Figure 3-2. MPU Subsystem Clocking Scheme



3.2.1.3 WakeUpGen

The WakeUpGen unit is responsible for generating a wakeup event from the incoming interrupts and enable bits. The WakeUpGen is implemented in the MPU always-on power domain.

3.2.1.3.1 WkUpGen Configuration Registers

The WkUpGen unit has configuration registers which can be accessed via an OCP interface. Table 3-1 summarizes the configuration registers in the WkUpGen unit.

The reserved bits in the WkUpGen configuration registers have the following properties: A) return a 0 on read and B) no effect on write.

IRQ8 is disabled by default and cannot be used for wake up. All other interrupts are enabled after reset.

Table 3-1. Summary of Configuration Registers in WkUpGen Unit

Register	Description	RW[1]	Address Offset	Reset
Wkg_control_0	WakeUpGen Control and Status Register	P/S R, SW	0x000	0x0000_0000
WkUpGenEnb_0A	CPU0 WakeUp Enable for Interrupts 32 to 63	P/S RW	0x010	0xFFFF_FEFF
WkUpGenEnb_0B	CPU0 WakeUp Enable for Interrupts 64 to 95	P/S RW	0x014	0xFFFF_FFFF
WkUpGenEnb_0C	CPU0 WakeUp Enable for Interrupts 96 to 127	P/S RW	0x018	0xFFFF_FFFF
WkUpGenEnb_0D	CPU0 WakeUp Enable for Interrupts 128 to 159	P/S RW	0x01C	0xFFFF_FFFF
WkUpGenEnb_0E	CPU0 WakeUp Enable for Interrupts 160 to 191	P/S RW	0x020	0xFFFF_FFFF
WkUpGenEnb_0F	CPU0 WakeUp Enable for Interrupts 192 to 223	P/S RW	0x024	0xFFFF_FFFF
WkUpGenEnb_10	CPU0 WakeUp Enable for Interrupts 224 to 255	P/S RW	0x028	0xFFFF_FFFF
AuxCoreBoot0	Registers used by OS to boot of Aux Core	P/S RW	0x800	0x0000_0000
Reserved	Reserved	R	0x804	0x0000_0000
PTMsyncreq_mask	[31:8] used by syncreq generation logic. [7:0] are read-only and read as 0x0	P/S RW	0xC00	0x0000_0000
PTMsyncreq_en	[0] used by syncreq generation logic. [31:1] are read-only and read as 0x0	P/S RW	0xC04	0x0000_0000
TimestampCycleLo	[31:0] of the 48 bit free running counter. Reset by PIMPUAONRSTN	P/S R	0xC08	0x0000_0000
TimestampCycleHi	[47:32] of the 48 bit free running counter read as [15:0]. [31:16] are read as 0x0. Reset by PIMPUAONRSTN	P/S R	0xC0C	0x0000_0000

[1] P/S = Public/secure (privilege or user mode)
R = Read-only

3.2.1.3.1.1 *WFI/WFE Control_0 and WFI/WFE Control_1*

Figure 3-3 shows the format of the WFI/WFE Control Register. Table 3-2 describes the WFI/WFE Control Register.

Figure 3-3. WFI/WFE Control Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
DOMAINRESET	MPUWARMRESET	MPUCOLDRESET	WDTRESETREQ	RESERVED	EVENTOSTATUS	STANDBYWFESTATUS	STANDBYWFISTATUS
R-0h							

7	6	5	4	3	2	1	0
RESERVED						WDTRESET_ENB	RESERVED
R-0h						R-0h	R-0h

Table 3-2. WFI/WFE Control Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	DOMAINRESET	RO	0h	Set when the PIDOMAINRSTONLY is asserted (read only).
14	MPUWARMRESET	RWc	0h	Set when the PIMPU_RSTN signal is asserted. Cleared by a write.
13	MPUCOLDRESET	RWc	0h	Set when the PIMPU_PWRON_RSTN signal is asserted. Cleared by a write.
12	WDTRESETREQ		0h	Set when the WD timer reset request signal from the SCU is asserted.
11	RESERVED	R	0h	
10	EVENTOSTATUS	RWc	0h	Set when a rising edge of EVENTO from CPU is detected. Given only for visibility/debug purpose.
9	STANDBYWFESTATUS	RWc	0h	Set when a rising edge of StandbyWFE from CPU is detected. Given only for visibility/debug purpose.
8	STANDBYWFISTATUS	RWc	0h	Set when a rising edge of StandbyWFI from CPU is detected. Given only for visibility/debug purpose.
7-2	RESERVED	R	0h	
1	WDTRESET_ENB	RW	0h	Enable Watch Dog Timer Reset assertion. When set to 1'b1, it will enable POMPU_RESETREQ (mapped to PO_SPARE_B_0) to be asserted when the Watch Dog timer (in the SCU) of the corresponding CPU expires.
0	RESERVED	R	0h	

In the RW column, RWc means a register bit is readable and gets cleared by a write (any input data).

3.2.1.3.2 WkupGenEnb Registers

Figure 3-4 shows the format of the WkUpGenEnb registers. Table 3-3 describes the WkUpGenEnb registers.

Each register is 32-bit wide. The LSB is the enable bit for the lowest interrupt line in that group. For example, bit 0 of WakeUGenEnb_0B is the enable bit for interrupt number 32. The enable bits are reset to 0 and can be set to 1 (by OCP configuration access) to enable the interrupt to wake up the CPU.

Figure 3-4. WkupGenEnb Registers

31	30	29	28	27	26	25	24
WkUpGenEnb[n+31]	WkUpGenEnb[n+30]	WkUpGenEnb[n+29]	WkUpGenEnb[n+28]	WkUpGenEnb[n+27]	WkUpGenEnb[n+26]	WkUpGenEnb[n+25]	WkUpGenEnb[n+24]
RW-1h	RW-1h	RW-1h	RW-1h	RW-1h	RW-1h	RW-1h	RW-1h

23	22	21	20	19	18	17	16
WkUpGenEnb[n+23]	WkUpGenEnb[n+22]	WkUpGenEnb[n+21]	WkUpGenEnb[n+20]	WkUpGenEnb[n+19]	WkUpGenEnb[n+18]	WkUpGenEnb[n+17]	WkUpGenEnb[n+16]
RW-1h	RW-1h	RW-1h	RW-1h	RW-1h	RW-1h	RW-1h	RW-1h

15	14	13	12	11	10	9	8
WkUpGenEnb[n+15]	WkUpGenEnb[n+14]	WkUpGenEnb[n+13]	WkUpGenEnb[n+12]	WkUpGenEnb[n+11]	WkUpGenEnb[n+10]	WkUpGenEnb[n+9]	WkUpGenEnb[n+8]
RW-1h	RW-1h	RW-1h	RW-1h	RW-1h	RW-1h	RW-1h	RW-0h

7	6	5	4	3	2	1	0
WkUpGenEnb[n+7]	WkUpGenEnb[n+6]	WkUpGenEnb[n+5]	WkUpGenEnb[n+4]	WkUpGenEnb[n+3]	WkUpGenEnb[n+2]	WkUpGenEnb[n+1]	WkUpGenEnb[n]
RW-1h	RW-1h	RW-1h	RW-1h	RW-1h	RW-1h	RW-1h	RW-1h

Table 3-3. WkupGenEnb Registers Field Descriptions

Bit	Field	Type	Reset	Description
31	WkUpGenEnb[n+31]	RW	1h	WakeUpGen enable for Interrupt line n+31
30	WkUpGenEnb[n+30]	RW	1h	WakeUpGen enable for Interrupt line n+30
29	WkUpGenEnb[n+29]	RW	1h	WakeUpGen enable for Interrupt line n+29
28	WkUpGenEnb[n+28]	RW	1h	WakeUpGen enable for Interrupt line n+28
27	WkUpGenEnb[n+27]	RW	1h	WakeUpGen enable for Interrupt line n+27
26	WkUpGenEnb[n+26]	RW	1h	WakeUpGen enable for Interrupt line n+26
25	WkUpGenEnb[n+25]	RW	1h	WakeUpGen enable for Interrupt line n+25
24	WkUpGenEnb[n+24]	RW	1h	WakeUpGen enable for Interrupt line n+24
23	WkUpGenEnb[n+23]	RW	1h	WakeUpGen enable for Interrupt line n+23
22	WkUpGenEnb[n+22]	RW	1h	WakeUpGen enable for Interrupt line n+22
21	WkUpGenEnb[n+21]	RW	1h	WakeUpGen enable for Interrupt line n+21
20	WkUpGenEnb[n+20]	RW	1h	WakeUpGen enable for Interrupt line n+20
19	WkUpGenEnb[n+19]	RW	1h	WakeUpGen enable for Interrupt line n+19
18	WkUpGenEnb[n+18]	RW	1h	WakeUpGen enable for Interrupt line n+18
17	WkUpGenEnb[n+17]	RW	1h	WakeUpGen enable for Interrupt line n+17
16	WkUpGenEnb[n+16]	RW	1h	WakeUpGen enable for Interrupt line n+16
15	WkUpGenEnb[n+15]	RW	1h	WakeUpGen enable for Interrupt line n+15
14	WkUpGenEnb[n+14]	RW	1h	WakeUpGen enable for Interrupt line n+14
13	WkUpGenEnb[n+13]	RW	1h	WakeUpGen enable for Interrupt line n+13
12	WkUpGenEnb[n+12]	RW	1h	WakeUpGen enable for Interrupt line n+12
11	WkUpGenEnb[n+11]	RW	1h	WakeUpGen enable for Interrupt line n+11
10	WkUpGenEnb[n+10]	RW	1h	WakeUpGen enable for Interrupt line n+10
9	WkUpGenEnb[n+9]	RW	1h	WakeUpGen enable for Interrupt line n+9
8	WkUpGenEnb[n+8]	RW	0h	WakeUpGen enable for Interrupt line n+8
7	WkUpGenEnb[n+7]	RW	1h	WakeUpGen enable for Interrupt line n+7
6	WkUpGenEnb[n+6]	RW	1h	WakeUpGen enable for Interrupt line n+6
5	WkUpGenEnb[n+5]	RW	1h	WakeUpGen enable for Interrupt line n+5
4	WkUpGenEnb[n+4]	RW	1h	WakeUpGen enable for Interrupt line n+4
3	WkUpGenEnb[n+3]	RW	1h	WakeUpGen enable for Interrupt line n+3
2	WkUpGenEnb[n+2]	RW	1h	WakeUpGen enable for Interrupt line n+2
1	WkUpGenEnb[n+1]	RW	1h	WakeUpGen enable for Interrupt line n+1
0	WkUpGenEnb[n]	RW	1h	WakeUpGen enable for Interrupt line n

3.3 Functional Description

3.3.1 Cortex-A9 MPCore

3.3.1.1 Timer and Watchdog Timer

The Cortex-A9 has its own timer and watchdog timer.

3.3.1.1.1 Power Domain of Timer and Watchdog Timer

The timer and watchdog timer are in the system power and clock domain. When the system is in WFI or OFF state, the timer will stop running. Software must depend on the timer only when the system power domain is ON and the DPLL clock is not gated off. If a running timer is required when the system power domain is off, software must use the system timer at the top-level.

Conversely, since the watchdog timer is in the MPU system power domain, the watchdog timer will continue to run while the CPU alone is clock-gated (A9 core internal clock gated during WFi and subsystem (DPLL) clock is running). If the software depends on the watchdog timer to stop when the CPU is in this mode, the software must disable the watchdog timer before the CPU goes into low-power state.

3.3.1.1.2 Watchdog Registers (Private Timer in ARM Documentation)

For register definitions, see the *ARM Cortex-A9 MPCore Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).

Table 3-4. WATCHDOG REGISTERS

Offset	Register Name
0x00	Private Timer Load
0x04	Private Timer Counter
0x08	Private Timer Control
0x0C	Private Timer Interrupt Status
0x20	Watchdog Load
0x24	Watchdog Counter
0x28	Watchdog Control
0x2C	Watchdog Interrupt Status
0x30	Watchdog Reset Status
0x34	Watchdog Disable

3.3.1.1.3 Global Timer Registers

For register definitions, see the *ARM Cortex-A9 MPCore Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).

Table 3-5. GLOBAL TIMER REGISTERS

Offset	Register Name
0x00, 0x04	Global Timer Counter
0x08	Global Timer Control
0x0C	Global Timer Interrupt Status
0x10, 0x14	Comparator Value
0x18	Auto Increment

3.3.1.2 PL310 L2 Cache Controller

For the feature list of the PL310, see Section 3.1.1.

The L2 cache controller on the MPU subsystem is the PL310. The L2 cache controller runs at the full CPU clock speed and is configured to have two slave ports and two master ports. All four ports are AXI interfaces with 64-bit data widths. All four ports run at full CPU speed.

The two master ports use load sharing to distribute the transactions. An address filtering mechanism is implemented but disabled by default. It can be enabled by software, if necessary.

The L2 cache size on the MPU subsystem is 256KB. The cache is configured as 16-way set associative, with a 32B line size. The L2 cache controller performs critical word first refilling with a pseudo-random cache replacement policy.

Parity checking is disabled on the L2 cache and parity bits are not implemented.

By default, the PL310 transforms all “shared” non-cacheable accesses to cacheable no-allocate for reads, or write-through no write-allocate for writes. This is the desired behavior for the MPU subsystem because the share bit must be set for coherent memory. The share attribute override feature (bit 22 of Auxiliary Control Register) must be disabled (that is, left in default state) in the PL310.

Cache lockdown by line and cache lockdown by master are implemented.

The PL310 includes logic to support cache event monitoring. All events that are monitored are routed to the HWDBG port.

The PL310 may be configured to generate interrupts on error conditions or event counter overflow and increment. The PL310 interrupt is routed to interrupt #0. When an interrupt occurs, software may look at register 2 (interrupt register) to determine the source of the interrupt.

3.3.1.2.1 PL310 Registers

For register definitions, see the *CoreLink Level 2 Cache Controller L2C-310 Technical Reference Manual Revision: r3p2* (available at infocenter.arm.com/help/index.jsp).

Table 3-6. PL310 REGISTERS

Offset	Register Name
0x000 - 0x0FC	Cache ID and Cache Type
0x100 - 0x1FC	Control
0x200 - 0x2FC	Interrupts and Counter Control
0x300 - 0x6FC	Reserved
0x700 - 0x7FC	Cache Maintenance Operations
0x800 - 0x8FC	Reserved
0x900 - 0x9FC	Cache Lockdown
0xA00 - 0xBFC	Reserved
0xC00 - 0xCFC	Address Filtering
0xD00 - 0xEFC	Reserved
0xF00 - 0xFFC	Debug, Prefetch, and Power

3.3.1.2.2 L2 as SRAM

The MPU subsystem supports usage of the 256KB L2 cache as general-purpose SRAM. The L2 cache comes up as disabled after reset. Software must ensure that the L2 cache is not enabled in this use case. Part usage of L2 as cache and part as SRAM is not supported.

L2 SRAM is memory mapped as a slave port on the chip-level interconnect. The MPU subsystem supports an OCMC64 module that converts the OCP accesses into SRAM accesses. The SRAM signals are muxed with the signals from PL310.

The input port PIUSEL2SRAM signal controls whether the PL310 path is used or the OCMC path. See the CTRL_MPU_L2 register.

The OCMC module is in the MPU voltage and power domain and is asynchronous to the chip-level L3 clock domain. The OCMC module can be clocked by MPU_CLK/2, MPU_CLK/3, MPU_CLK/4, or MPU_CLK/6. The input pin PIL2SRAMCLKDIV[1:0] determines the divide ratio.

00 - MPU_CLK/2
01 - MPU_CLK/3
10 - MPU_CLK/4
11 - MPU_CLK/6

NOTE: Always ensure that the OCMC (L2) is clocked at a frequency lower or equal to the chip-level L3 clock frequency by using the appropriate divide ratios. The ASYNC bridge on the OCMC path does not support response flow control, and if the above restriction is not posed, it could lead to a hang scenario because the OCMC can give responses faster than the rate at which the L3 can accept them.

In addition, changing the divide ratio (PIL2SRAMCLKDIV) or changing the L2 from cache to SRAM (PIUSEL2SRAM) dynamically is not permitted.

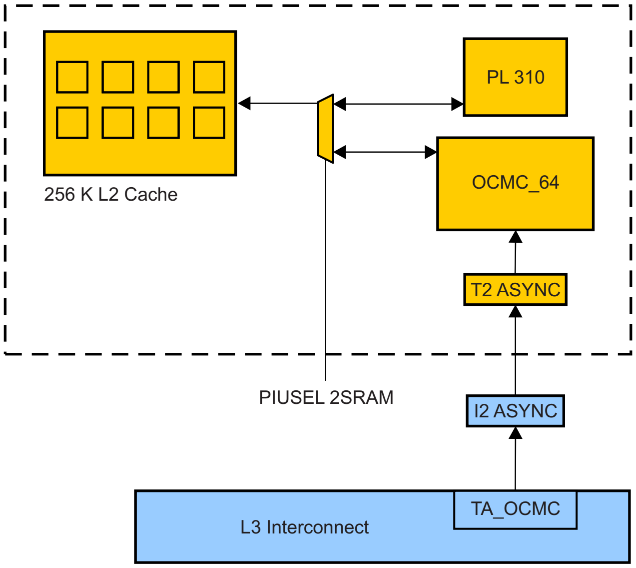
3.3.1.2.3 Power and Clock Gating Concerns

The MPU power domain must be ON while using the L2 as SRAM because the OCMC controller resides in the MPU power domain.

The MPU PLL cannot be clock-gated when OCMC is used for mapping L2 as SRAM because the OCMC controller receives the clock from the MPU PLL (see Figure 3-5). The MPU PLL supplies a clock to all of the MPU subsystem components, including the CPU. To achieve lower power in this use case, you must use CPU-level clock gating (inside the Cortex-A9).

For more details on CPU-level clock gating, see Section 3.2, *Integration*.

Figure 3-5. L2 Usage as SRAM



3.3.1.3 Generalized Interrupt Controller (GIC)

For details about the GIC, see the *ARM Cortex-A9 MPCore Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp). The Cortex-A9 GIC shares the same programmer's model as the GIC-PL390.

3.3.1.3.1 Interrupts

The first 32 interrupts are mapped internally in the ARM MPU subsystem. For details on the available interrupts, see the *ARM Cortex-A9 MPCore Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).

There are also an additional 224 shared peripheral interrupts (SPI) which are external to the ARM MPU subsystem. For SPI interrupt definitions, see Chapter 8, *Interrupts*.

3.3.1.3.2 Distributor Registers

For register definitions, see the *ARM Cortex-A9 MPCore Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).

Table 3-7. DISTRIBUTOR REGISTERS

Offset	Register Name
0x000	Distributor Control
0x004	Interrupt Controller Type
0x008	SCU CPU Power Status
0x00C - 0x7C	Reserved
0x080 - 0x09C	Interrupt Security
0x100 - 0x11C	Interrupt Set-Enable
0x180 - 0x19C	Interrupt Clear-Enable
0x200 - 0x27C	Interrupt Set-Pending
0x280 - 0x29C	Interrupt Clear-Pending
0x300 - 0x31C	Active Bit
0x380 - 0x3FC	Reserved
0x400 - 0x4FC	Interrupt Processor Target
0xBFC	Reserved
0xC00 - 0xC3C	Interrupt Configuration
0xD00	PPI Status
0xD04 - 0xD1C	SPI Status
0xD80 - 0xEFC	Reserved
0xF00	Software Generated Interrupt

3.3.1.3.3 Interrupt Controller (INTC) Interface Registers

For register definitions, see the *ARM Cortex-A9 MPCore Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).

Table 3-8. INTC REGISTERS

Offset	Register Name
0x000	CPU Interface Control
0x004	Interrupt Priority Mask
0x008	Binary Point Register
0x00C	Interrupt Acknowledge
0x010	End of Interrupt
0x014	Running Priority
0x018	Highest Pending Interrupt
0x01C	Aliased Non-Secure Binary Point
0x0FC	CPU Interface Implementer ID

3.3.1.4 Snoop Control Unit (SCU)

The SCU connects the Cortex-A9 processor to the memory system through the AXI interfaces.

The SCU functions is to initiate L2 AXI memory accesses.

NOTE: The Cortex SCU does not support hardware management of coherency of the instruction cache.

3.3.1.4.1 SCU Registers

For register definitions, see the *ARM Cortex-A9 MPCore Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).

Table 3-9. SCU REGISTERS

Offset	Register Name
0x00	SCU Control
0x04	SCU Configuration
0x08	SCU CPU Power Status
0x0C	SCU Invalidate All Registers in Secure State
0x40	Filtering Start Address
0x44	Filtering End Address
0x50	SCU Access Control (SAC)
0x54	SCU Non-secure Access Control

3.3.1.5 AXI2OCP Interface

The AXI2OCP bridge is used to connect the AXI bus on the ARM A9 to the OCP native L3 interconnect (64-bit widths) and interrupt controller. The AXI2OCP bridge converts between AXI and OCP protocols and maintains a mapping of AXI tags to the OCP Tag ID.

3.3.1.6 Memory Adaptor

The memory adaptor bridge is used to connect the AXI bus on the ARM A9 to the OCP native L3 interconnect (128-bit width). The memory adaptor contains logic to minimize cache miss latency.

3.3.1.7 Configuration Options

This section provides the configurable design options for the Cortex-A9.

Feature	Selected Option
Cortex-A9 processors	One
Instruction Cache Size	32KB
Data Cache Size	32KB
TLB size	128-entry
VFP / NEON	Included
Jazelle DBX extension	Included
Program Trace Macrocell (PTM)	Included
Power off and Dormant Wrappers	Included
Support for Parity Error Detection	Not Included
Accelerator Coherency Port	Included
Shared Peripheral Interrupts	224

Interconnects

This chapter describes the interconnects of the device.

Topic **Page**

4.1 Introduction

The system interconnect is based on a 2-level hierarchical architecture (L3, L4) driven by system performance. The L4 interconnect is based on a fully native OCP infrastructure, directly complying with the OCPIP2.2 reference standard.

4.1.1 Terminology

The following is a brief explanation of some terms used in this document:

Initiator: Module able to initiate read and write requests to the chip interconnect (typically: processors, DMA, etc.).

Target: Unlike an initiator, a target module cannot generate read/write requests to the chip interconnect, but it can respond to these requests. However, it may generate interrupts or a DMA request to the system (typically: peripherals, memory controllers). **Note:** A module can have several separate ports; therefore, a module can be an initiator and a target.

Agent: Each connection of one module to one interconnect is done using an agent, which is an adaptation (sometimes configurable) between the module and the interconnect. A target module is connected by a target agent (TA), and an initiator module is connected by an initiator agent (IA).

Interconnect: The decoding, routing, and arbitration logic that enable the connection between multiple initiator modules and multiple target modules connected on it.

Register Target (RT): Special TA used to access the interconnect internal configuration registers.

Data-flow Signal: Any signal that is part of a clearly identified transfer or data flow (typically: command, address, byte enables, etc.). Signal behavior is defined by the protocol semantics.

Sideband Signal: Any signal whose behavior is not associated to a precise transaction or data flow.

Command Slot: A command slot is a subset of the command list. It is the memory buffer for a single command. A total of 32 command slots exist.

Out-of-band Error: Any signal whose behavior is associated to a device error-reporting scheme, as opposed to in-band errors. Note: Interrupt requests and DMA requests are not routed by the interconnect in the device.

ConnID: Any transaction in the system interconnect is tagged by an in-band qualifier ConnID, which uniquely identifies the initiator at a given interconnect point. A ConnID is transmitted in band with the request and is used for error-logging mechanism.

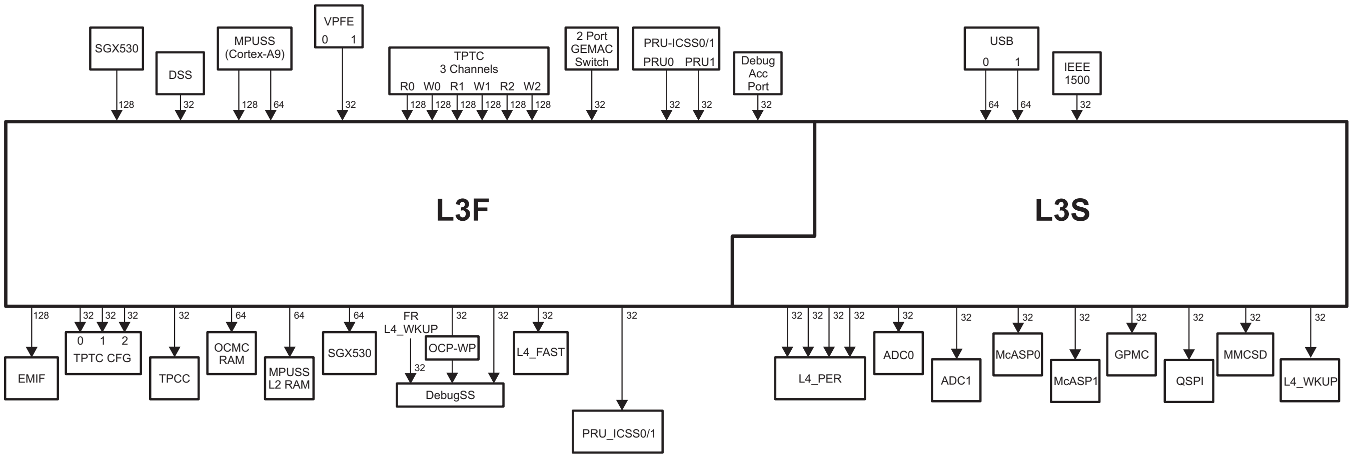
4.1.2 L3 Interconnect

The L3 high-performance interconnect is based on a Network-On-Chip (NoC) interconnect infrastructure. The NoC uses an internal packet-based protocol for forward (read command, write command with data payload) and backward (read response with data payload, write response) transactions. All exposed interfaces of this NoC interconnect, both for Targets and Initiators; comply with the OCPIP2.2 reference standard.

4.1.2.1 L3 Topology

The L3 topology is driven by performance requirements, bus types, and clocking structure. The main L3 paths are shown in Figure 4-1. Arrows indicate the master/slave relationship not data flow. L3 is partitioned into two separate clock domains: L3F corresponds to L3 Fast clock domain and L3S corresponds to L3 Slow clock domain.

Figure 4-1. L3 Topology



4.1.2.2 L3 Port Mapping

Each initiator and target core is connected to the L3 interconnect through a Network Interface Unit (NIU). The NIUs act as entry and exit points to the L3 Network-on-Chip (NoC) – converting between the IP's OCP protocol and the NoC's internal protocol, and also include various programming registers. All ports are single threaded with tags used to enable pipelined transactions. The interconnect includes:

Initiator Ports:

- L3F
  - Cortex A9 MPUSS 128-bit initiator port0 and 64-bit initiator port1
  - SGX530 128-bit initiator port
  - 3 EDMA3TC (TPTC) 128-bit read initiator ports
  - 3 EDMA3TC (TPTC) 128-bit write initiator ports
  - 2 PRU-ICSS0/1 32-bit initiator ports
  - 2 port Gigabit Ethernet Switch (CPGSW) 32-bit initiator port
  - Debug Subsystem 32-bit initiator port
  - VPFE0 32-bit initiator port
  - VPFE1 32-bit initiator port
  - DSS 32-bit initiator port
- L3S
  - 2 USB 64-bit initiator ports
  - P1500 32-bit initiator port

Target Ports:

- L3F
  - EMIF 128-bit target port
  - 3 EDMA3TC (TPTC) CFG 32-bit target ports
  - EDMA3CC (TPCC) CFG 32-bit target port
  - OCMC RAM0 64-bit target port
  - DebugSS 32-bit target port
  - SGX530 64-bit target port
  - L4_FAST 32-bit target port
  - PRU-ICSS0/1 32-bit target port
  - MPUSS L2 RAM

- L3S
  - 4 L4_PER peripheral 32-bit target ports
  - GPMC 32-bit target port
  - McASP0 32-bit target port
  - McASP1 32-bit target port
  - QSPI 32-bit target port
  - ADC0 32-bit target port
  - MMCSD 32-bit target port
  - L4_WKUP wakeup 32-bit target port
  - ADC1 FIFO 32-bit target port

4.1.2.3 Interconnect Connections

The L3 connections between bus masters and slave ports are shown in Table 4-1. The L3 interconnect will return an address-hole error if any initiator attempts to access a target to which it has no connection.

Table 4-1. L3 Master — Slave Connectivity

Masters	Slaves																				
	EMIF	OCMC-RAM	MPUSS L2 RAM	TPTC0–2 CFG	TPCC	SGX530	L4_Fast	L4_PER Port0	L4_PER Port1	L4_PER Port2	L4_PER Port3	McASP0-1	GPMC	ADC0 FIFO / DMA	MMCSD2 / DMA FIFO	L4_WKUP	DebugSS	NOC Regs	ADC1 FIFO / DMA	QSPI	PRU-ICSS1
MPUSS M1 (128-bit)	X																				
MPUSS M2 (64-bit)	X	X	X	X	X	X	X	X				X	X	X	X	X	X	X	X	X	X
TPTC0 RD	X	X	X		X		X			X		X	X	X	X	X			X	X	X
TPTC0 WR	X	X	X		X		X			X		X	X	X	X	X	X		X	X	X
TPTC1 RD	X	X	X		X		X			X		X	X	X	X	X			X	X	X
TPTC1 WR	X	X	X		X		X			X		X	X	X	X	X	X		X	X	X
TPTC2 RD	X	X	X		X		X				X	X	X	X	X	X			X	X	X
TPTC2 WR	X	X	X		X		X				X	X	X	X	X	X	X		X	X	X
PRU-ICSS (PRU0)	X	X	X	X	X		X		X			X	X	X	X	X	X	X	X	X	X
PRU-ICSS (PRU1)	X	X	X	X	X		X		X			X	X	X	X	X	X	X	X	X	X
GEMAC	X	X	X										X							X	
SGX530	X	X											X								
USB0	X	X	X										X								
USB1	X	X	X										X							X	
EMU (DAP)	X	X	X	X	X	X	X				X	X	X	X	X	X	X	X	X	X	X
IEEE1500	X	X	X	X	X	X	X				X	X	X	X	X	X	X	X	X	X	X
DSS	X	X	X										X								
VPFE0	X	X	X										X								
VPFE1	X	X	X										X								

4.1.2.4 ConnID Assignment

Each L3 initiator includes a unique 6-bit master connection identifier (MConnID) that is used to indentify the source of a transfer request.

Table 4-2. MConnID Assignment

Initiator	6-bit MConnID (Debug)	Instrumentation	Comment
MPUSS M1 (128-bit)	0x00	0	Connects only to EMIF
MPUSS M2 (64-bit)	0x01	SW	
DAP	0x04	SW	
P1500	0x05	SW	
PRU-ICSS0/1 (PRU0)	0x0C	SW	
PRU-ICSS0/1 (PRU1)	0x0D	SW	
Wakeup Processor	0x14	SW	Connects only to L4_WKUP
TPTC0 Read	0x18	0	
TPTC0 Write	0x19	SW	One WR port for data logging
TPTC1 Read	0x1A	0	
TPTC1 Write	0x1B	0	
TPTC2 Read	0x1C	0	
TPTC2 Write	0x1D	0	
SGX530	0x20	0	
OCP WP Traffic Probe	0x21 (1)	HW	Direct connect to DebugSS
OCP WP DMA Profiling	0x22 (1)	HW	Direct connect to DebugSS
OCP-WP Event Trace	0x23 (1)	HW	Direct connect to DebugSS
DSS	0x25	0	
VPFE0	0x2C	0	
VPFE1	0x2D	0	
GEMAC	0x30	0	
USB0_RD	0x34	0	
USB0_WR	0x35	0	
USB1_RD	0x36	0	
USB1_WR	0x37	0	
Stat Collector 0	0x3C	HW	
Stat Collector 1	0x3D	HW	
Stat Collector 2	0x3E	HW	
Stat Collector 3	0x3F	HW	

(1) These MConnIDs are generated within the OCP-WP module based on the H0, H1, and H2 configuration parameters.

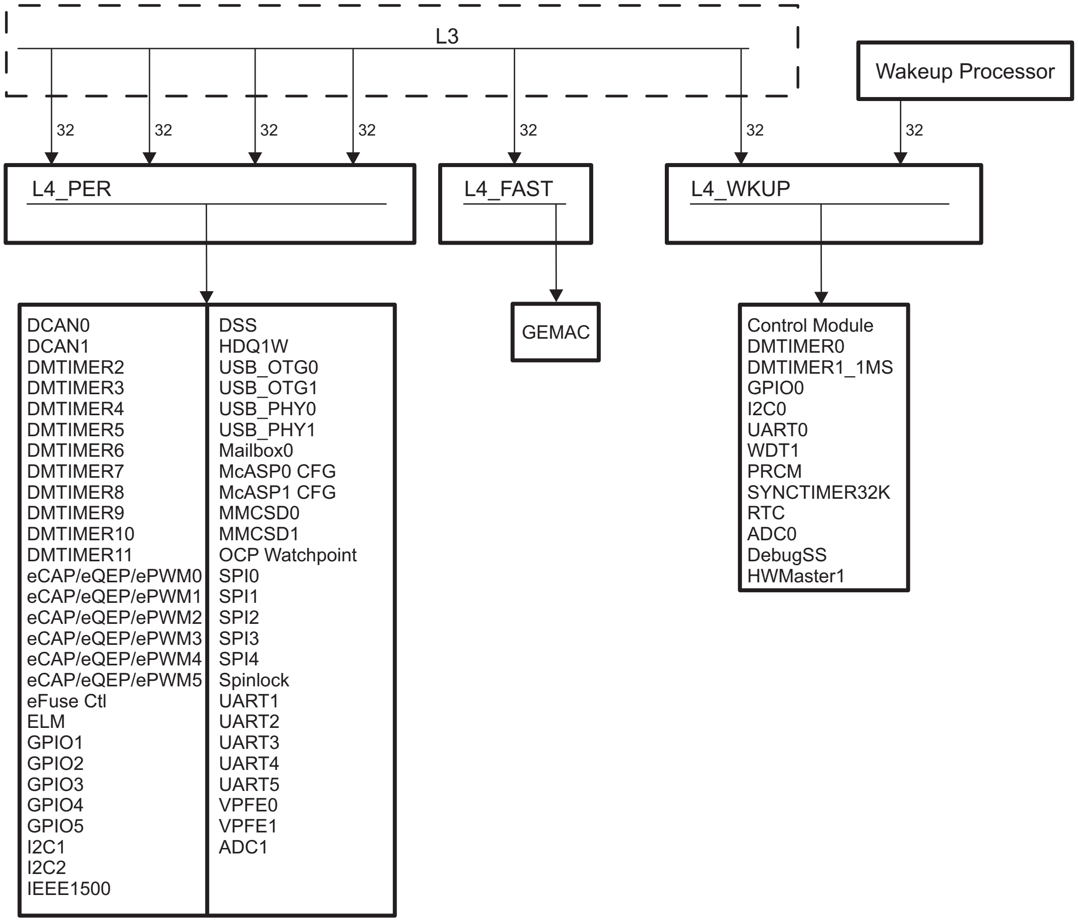
NOTE: Instrumentation refers to debug type. SW instrumentation means that the master can write data to be logged to the STM (similar to a printf()). HW indicates debug data captured automatically by hardware. A '0' entry indicates no debug capability.

4.1.3 L4 Interconnect

The L4 interconnect is a non-blocking peripheral interconnect that provides low latency access to a large number of low bandwidth, physically dispersed target cores. Each L4 can handle incoming traffic from up to four initiators and can distribute those communication requests to and collect related responses from up to 63 targets.

This device provides three interfaces with L3 interconnect for High Speed Peripheral, Standard Peripheral, and Wakeup Peripherals. Figure 4-2 shows the L4 bus architecture and memory-mapped peripherals.

Figure 4-2. L4 Topology



Chapter 5

SPRUHL7H – April 2014 – Revised January 2018

Initialization

This chapter describes the initialization of the device.

Topic **Page**

5.1 Introduction

This section describes the booting functionality of the device, referred hereafter as ROM Code. The booting functionality covers the following features:

- Memory Booting: booting the device by executing firmware stored on permanent memories like flash-memory or memory cards. This process usually occurs after a cold or warm reset of the device.
- Peripheral Booting: booting the device by downloading the executable code over a communication interface such as UART, USB, or Ethernet. This process is can also be used to flash a device.

The device always starts in secure mode. The Secure ROM Code handles early initialization. The Secure ROM code switches the device into public mode. Hence the Public ROM Code provides run-time services for cache maintenance.

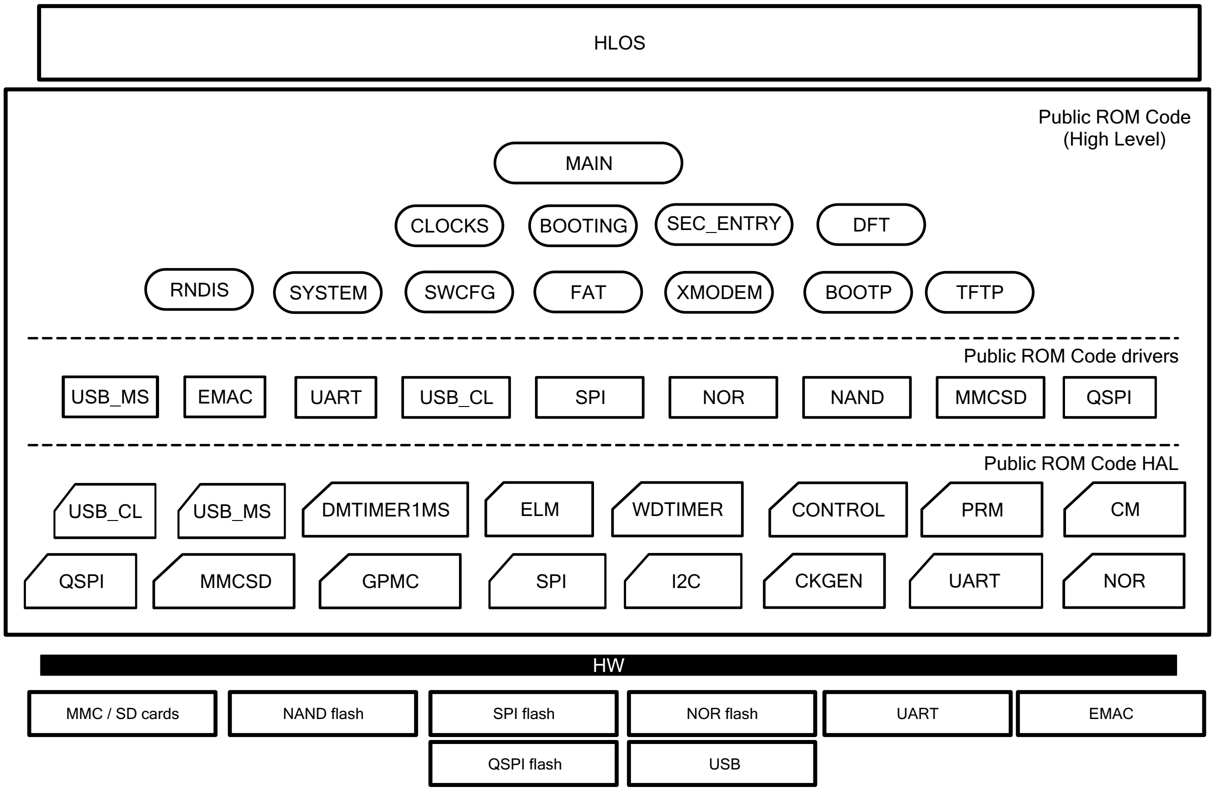
5.2 Functional Description

5.2.1 Architecture

Figure 5-1 shows the architecture of the Public ROM Code. It is split into three main layers with a top-down approach: high-level, drivers, and hardware abstraction layer (HAL). One layer communicates with a lower level layer through a unified interface.

- The high-level layer implements the main tasks of the Public ROM Code: watchdog and clocks configuration and main booting routine.
- The driver layer implements the logical and communication protocols for any booting device in accordance with the interface specification.
- The HAL implements the lowest level code for interacting with the hardware infrastructure IPs. End booting devices attach to device IO pads.

Figure 5-1. Public ROM Code Architecture



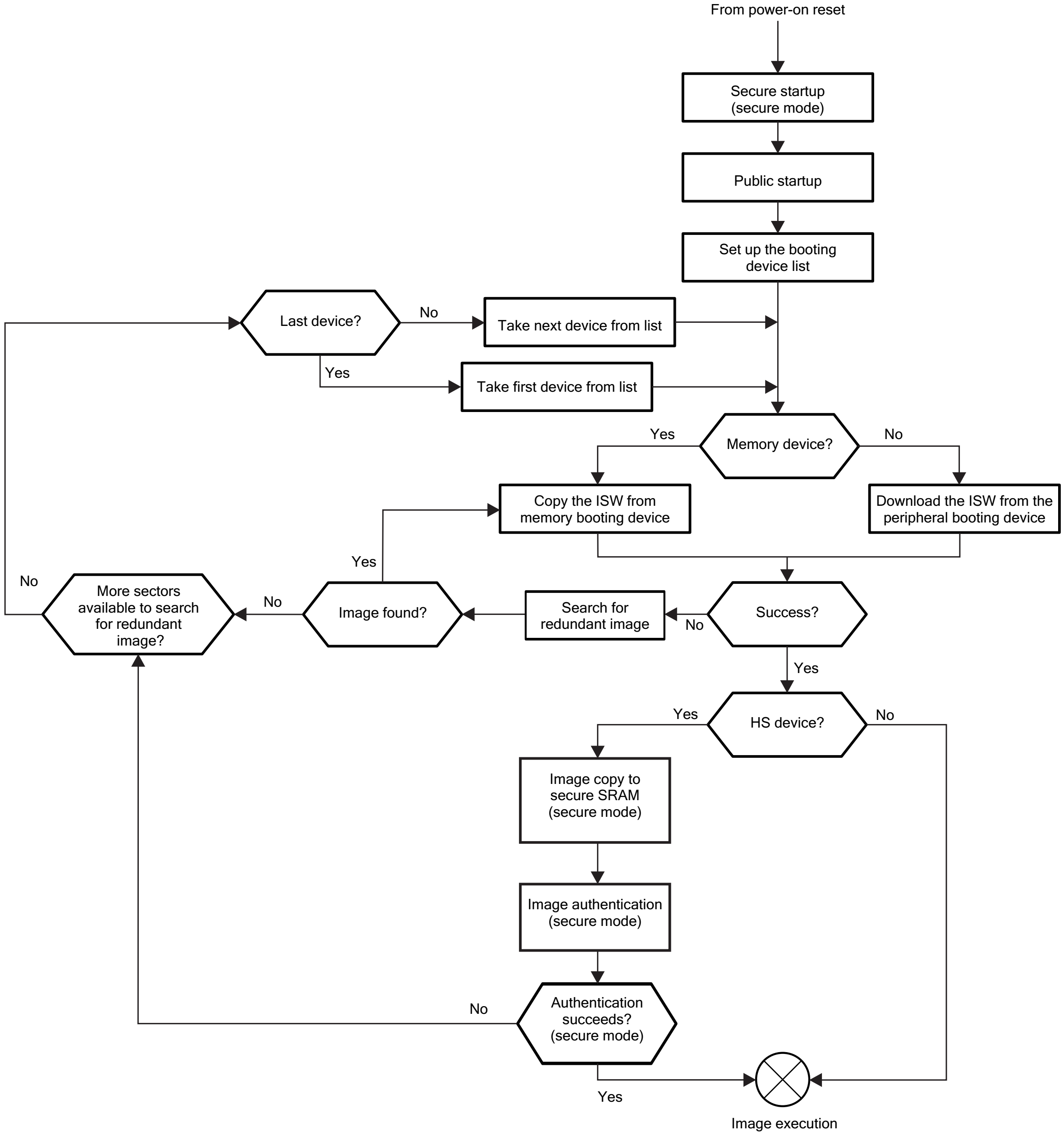
5.2.2 Functionality

Figure 5-2 shows the high-level flow for the Public ROM Code booting procedure. The Public ROM Code starts after the secure startup. The ROM Code then performs platform configuration and initialization as part of the public start-up procedure.

The booting device list is based on the SYSBOOT pins. A booting device can be a memory booting device (soldered flash memory or temporarily booting device like memory card) or a peripheral interface connected to a host.

The main loop of the booting procedure searches the booting device list for an image from the currently selected booting device. This loop exits if a valid booting image is found and successfully executed or if the watchdog expires.

Figure 5-2. Public ROM Code Boot Procedure



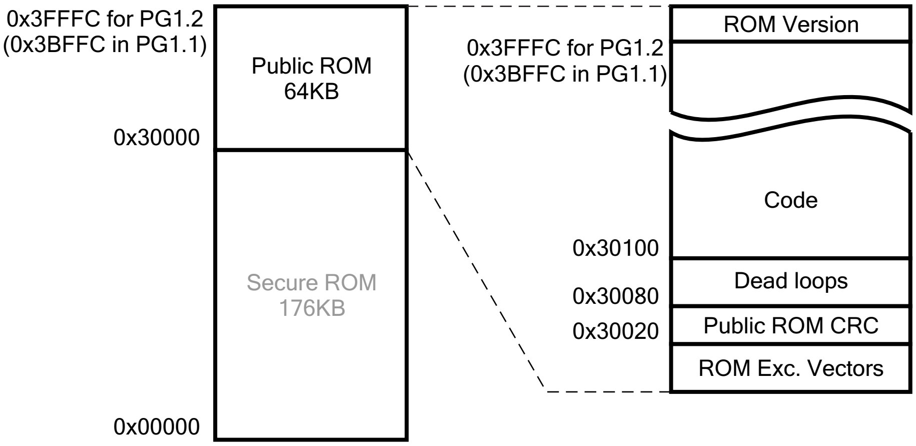
5.2.3 Memory Map

5.2.3.1 Public ROM Memory Map

Figure 5-3 shows the on-chip ROM memory map. The top holds the Public ROM Code. The Public ROM Code mapping includes:

- Exception vectors
- CRC
- Dead loops collection
- Code and const data sections
- ROM Version

Figure 5-3. ROM Memory Map



5.2.3.1.1 Public ROM Exception Vectors

Table 5-1 lists the Public ROM exception vectors. These vectors handle the standard exceptions that occur during the code execution. For example, if there is an issue accessing the memory region, it will generate a data abort exception when trying to read or write into that memory. The reset exception is redirected to the Public ROM Code startup. Other exceptions are redirected to their RAM handlers by loading appropriate addresses into the PC register.

Table 5-1. Public ROM Exception Vectors

Address	Exception	Content
30000h	Reset	Branch to the Public ROM Code startup
30004h	Undefined	PC = 40338E04h
30008h	SWI	PC = 40338E08h
3000Ch	Pre-fetch abort	PC = 40338E0Ch
30010h	Data abort	PC = 40338E10h
30014h	Unused	PC = 40338E14h
30018h	IRQ	PC = 40338E18h
3001Ch	FIQ	PC = 40338E1Ch

5.2.3.1.2 Public ROM Code CRC

The Public ROM Code CRC is calculated as 32-bit CRC code (CRC-32-IEEE 802.3) for the address range 30000h – 3FFFCh for PG1.2 (0x3BFFC for PG1.1). The 4-byte CRC code is stored at location 30020h.

5.2.3.1.3 Dead Loops

Table 5-2 lists the built-in dead loops used for different purposes. All dead loops are branch instructions coded in ARM mode. The fixed location of these dead loops facilitates debugging and testing. The first seven dead loops are linked to default ROM exception handlers mentioned in Table 5-1 using RAM exception vectors mentioned in Table 5-4 and Table 5-5.

Table 5-2. Dead Loops

Address	Purpose
30080h	Undefined exception default handler
30084h	SWI exception default handler
30088h	Pre-fetch abort exception default handler
3008Ch	Data abort exception default handler
30090h	Unused exception default handler
30094h	IRQ exception default handler
30098h	FIQ exception default handler
3009Ch	Validation tests PASS
300A0h	Validation tests FAIL
300A4h	Reserved
300A8h	Image not executed or returned.
300Ach	Reserved
300B0h	Reserved
300B4h	Reserved
300B8h	Reserved
300BCh	Reserved

5.2.3.1.4 Code

This space is used to hold code and constant data.

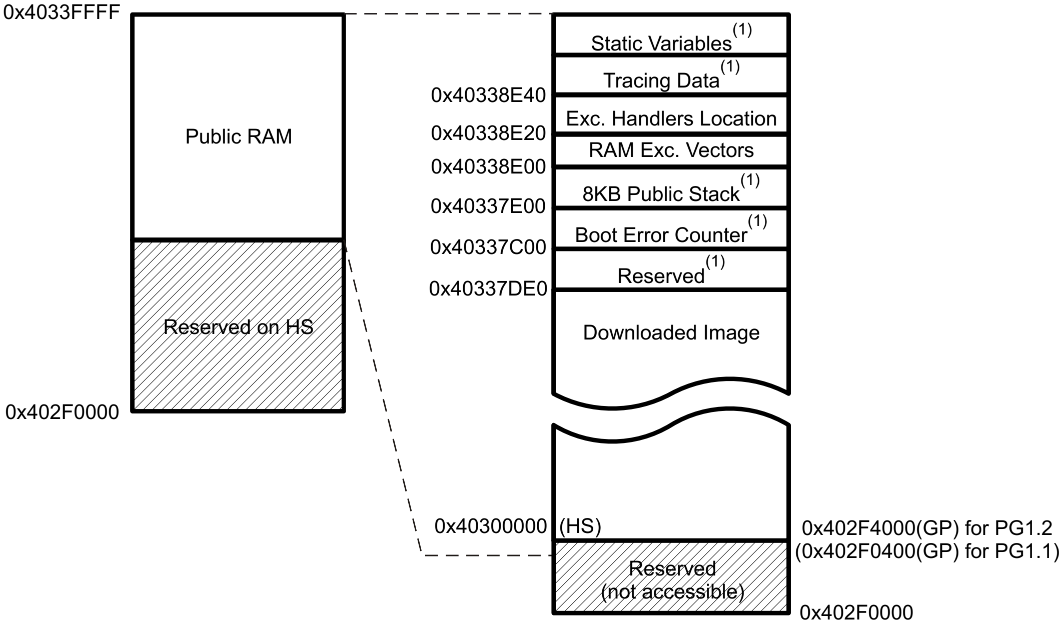
5.2.3.1.5 Public ROM Code Version

The ROM Code version includes two decimal numbers: major and minor. The version identifies the ROM Code release version in a given IC. The ROM Code version is a 32-bit hexadecimal value located at address 3FFFCh for PG1.2 (0x3BFFC for PG1.1). The minor Version is represented by the 1st byte and the major version is represented by the 2nd byte.

5.2.3.2 Public L3 RAM Memory Map

The Public ROM Code makes use of the on chip RAM module connected to the L3 interconnect (referred hereafter as L3 RAM). Its usage is shown in Figure 5-4. The Public L3 RAM memory map ranges from address 402F0000h to 4033FFFFh.

Figure 5-4. Public L3 RAM Memory Map



(1) Reserved for ROM use.

5.2.3.2.1 *Downloaded Image (ISW)*

This area is used by the Public ROM Code to store the downloaded Initial SW. The downloaded image can be up to 271KB on GP Device. It resides from 0x402F4000 (0x402F0400 for PG1.1) to 0x40337C00 in GP device.

Note: If ROM boots using USB_MS boot mode, then the image can reside only from 0x40300000. This limits the maximum image size while using USB_MS boot mode to 220KB.

5.2.3.2.2 *Boot Error Counters*

Boot error counter registers contain the number of attempts a particular boot mode has failed after a cold reset. Each boot mode error counter is 32 bits wide The device numbers 1, 2, 3, 4 mentioned are the sequential number in which boot modes are tried with a particular set of sysboot pin configuration. The SYSBOOT pin configuration is given in Table 5-10.

Table 5-3. Boot Error Counters

Address	Boot Mode
40337DE0h	Device No. 1
40337DE4h	Device No. 2
40337DE8h	Device No. 3
40337DECh	Device No. 4

5.2.3.2.3 *Public Stack*

Space reserved for stack.

5.2.3.2.4 *RAM Exception Vectors and Exception Handlers Location*

The RAM exception vectors enable a simple means to redirect exceptions to custom handlers. Table 5-4 shows content of the RAM space reserved for RAM vectors. The first seven addresses are ARM instructions which load the value located in the subsequent seven addresses into the PC register. These instructions execute when an exception occurs since they are called from the ROM exception vectors.

Table 5-4. RAM Exception Vectors

Address	Exception	Content
40338E00h	Reserved	Reserved
40338E04h	Undefined	PC = [40338E24h]
40338E08h	SWI	PC = [40338E28h]
40338E0Ch	Pre-fetch abort	PC = [40338E2Ch]
40338E10h	Data abort	PC = [40338E30h]
40338E14h	Unused	PC = [40338E34h]
40338E18h	IRQ	PC = [40338E38h]
40338E1Ch	FIQ	PC = [40338E3Ch]

As Table 5-5 shows, Undefined, SWI, Unused and FIQ exceptions are redirected to a hardcoded dead loop. Pre-fetch abort, data abort, and IRQ exception are redirected to pre-defined ROM handlers. User code can redirect any exception to a custom handler by writing its address to the appropriate location from 4033D024h to 4033D03Ch or by overriding the branch (load into PC) instruction between addresses from 4033D004h to 4033D01Ch.

Table 5-5. RAM Exception Handlers Location

Address	Exception	Content
40338E20h	Reserved	30090h
40338E24h	Undefined	30080h
40338E28h	SWI	30084h
40338E2Ch	Pre-fetch abort	Address of default pre-fetch abort handler(1)
40338E30h	Data abort	Address of default data abort handler(1)
40338E34h	Unused	30090h
40338E38h	IRQ	Address of default IRQ handler
40338E3Ch	FIQ	30098h

(1) For more details, see Section 5.2.3.1.1, *Public ROM Exception Vectors*.

5.2.3.2.5 Tracing Data

This area contains trace vectors reflecting the execution path of the public boot. Section 5.2.12, *Tracing*, describes the different trace vectors and lists all the possible trace codes.

Table 5-6. Tracing Data

Address	Size (bytes)	Description
40338E40h	4	Current tracing vector, word 1
40338E44h	4	Current tracing vector, word 2
40338E48h	4	Current tracing vector, word 3
40338E4Ch	4	Current tracing vector, word 4
40338E50h	4	Current tracing vector, word 5
40338E54h	4	Reserved
40338E58h	4	Reserved
40338E5Ch	4	Reserved
40338E60h	4	Reserved
40338E64h	4	Reserved

5.2.3.2.6 Static Variables

This area contains the ROM Code static variables used during boot time.

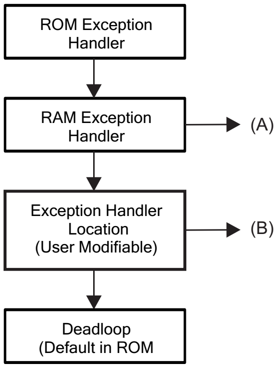
5.2.3.3 Public ROM Exception Handling

When an exception occurs, ROM Code branches to the corresponding ROM Exception Handler. The ROM exception handler changes the PC value to the RAM Exception Vectors (ROM specific). The instruction at RAM Exception Vectors (ROM specific) contains an instruction to change the PC to the value written at Exception Handlers Location (User Modifiable). In this location, the default value is the location of deadloops. The user can overwrite these values to point to their specific Exception Handlers.

For example, if Undefined Instruction Abort, the flow is:

30004h → 40338E04h (Load PC with value at 40338E24h) → 30080h {Deadloop}

Figure 5-5. The ROM Exception Handling Flow



A Instruction to load the PC with the value at the exception handler location.

B User code can update values here to point to their specific exception handler.

5.2.3.3.1 Specific Cases of Abort Handlers

The default handlers for pre-fetch and data abort perform reads from CP15 debug registers to retrieve the reason of the abort

5.2.3.3.1.1 Prefetch Abort

If pre-fetch abort: the IFAR register is read from CP15 and stored in R0. The IFSR register is read and stored in the R1 register. Then the ROM Code jumps to the pre-fetch abort dead loop (30088h).

5.2.3.3.1.2 Data Abort

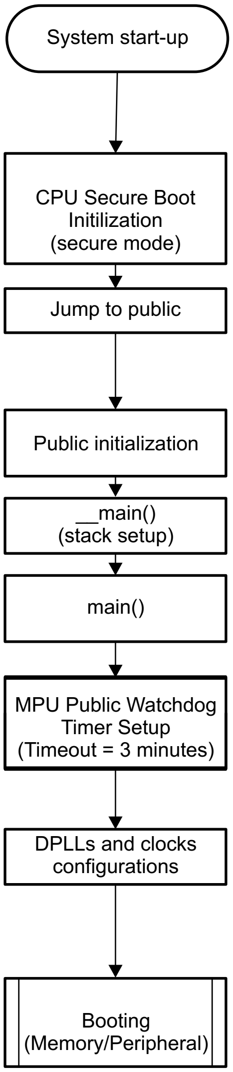
If data abort: the DFAR register is read from CP15 and stored in R0. The DFSR register is read and stored in the R1 register. Then the ROM Code jumps to the data abort dead loop (3008Ch).

5.2.4 Start-up and Configuration

5.2.4.1 ROM Code Startup

The Public ROM Code is physically located at the address 30000h that is immediately next to the Secure ROM Code.

Figure 5-6. ROM Code Startup Sequence



As Figure 5-6 shows, the CPU jumps to the Public ROM Code reset vector once it has completed the secure boot initialization.

Once in public mode, the CPU:

1. Performs the public-side initialization and stack setup (compiler auto generated C- initialization or “scatter loading”)
2. Configures the public watchdog timer (set to three minutes)
3. Performs system clocks configuration
4. Jumps to the booting routine

5.2.4.2 CPU State at Public Startup

The CPU L1 **instruction cache** and **branch prediction** mechanisms are not activated as part of the public boot process. The public vector base address is configured to the reset vector of Public ROM Code (30000h). MMU is left switched off during the public boot (hence L1 data cache off).

5.2.4.3 Clocking Configuration

Support for the following frequencies are based on SYSBOOT[15:14].

Table 5-7. Crystal Frequencies Supported

SYSBOOT[15:14]	Crystal Frequency
00b	19.2 MHz
01b	24 MHz
10b	25 MHz
11b	26 MHz

The ROM Code configures the clocks and DPLLs which are necessary for ROM Code execution:

- CORE ADPLLS is locked at 2 GHz and divided further to provide 100 MHz clocks for L3 functional clock and 100 MHz for Ethernet Module. The CORE ADPLLS is not reconfigured on warm reset if Ethernet isolation is enabled.
- MPU ADPLLS is locked to provide 300 MHz for the Cortex-A9.
- PER ADPLLJ is locked to provide 960 MHz and 192 MHz for peripheral blocks.

Table 5-8 summarizes the ROM Code default settings for clocks. This default configuration enables all the ROM Code functions with minimized needs on power during boot.

Table 5-8. ROM Code Default Clock Settings

Clock	Frequency (MHz)	Source (Source Frequency)
L3_FCLK	100	CORE_CLKOUTM4 (100 MHz)
SPI_CLK	48	PER_CLKOUTM2 (192 MHz)
MMC_CLK	96	PER_CLKOUTM2 (192 MHz)
UART_CLK	48	PER_CLKOUTM2 (192 MHz)
I2C_CLK	48	PER_CLKOUTM2 (192 MHz)
MPU_CLK	300	MPU_PLL (300 MHz)
USB_PHY_CLK	960	PER_CLKDCOLDO (960 MHz)
QSPI_CLK	12	PER_CLKOUTM2 (192 MHz)
MHZ_250_CLK (Ethernet)	100	CORE_CLKOUTM5 (100 MHz)
MHZ_50_CLK (Ethernet)	50	CORE_CLKOUTM5 (100 MHz)
MHZ_5_CLK (Ethernet)	5	CORE_CLKOUTM5 (100 MHz)

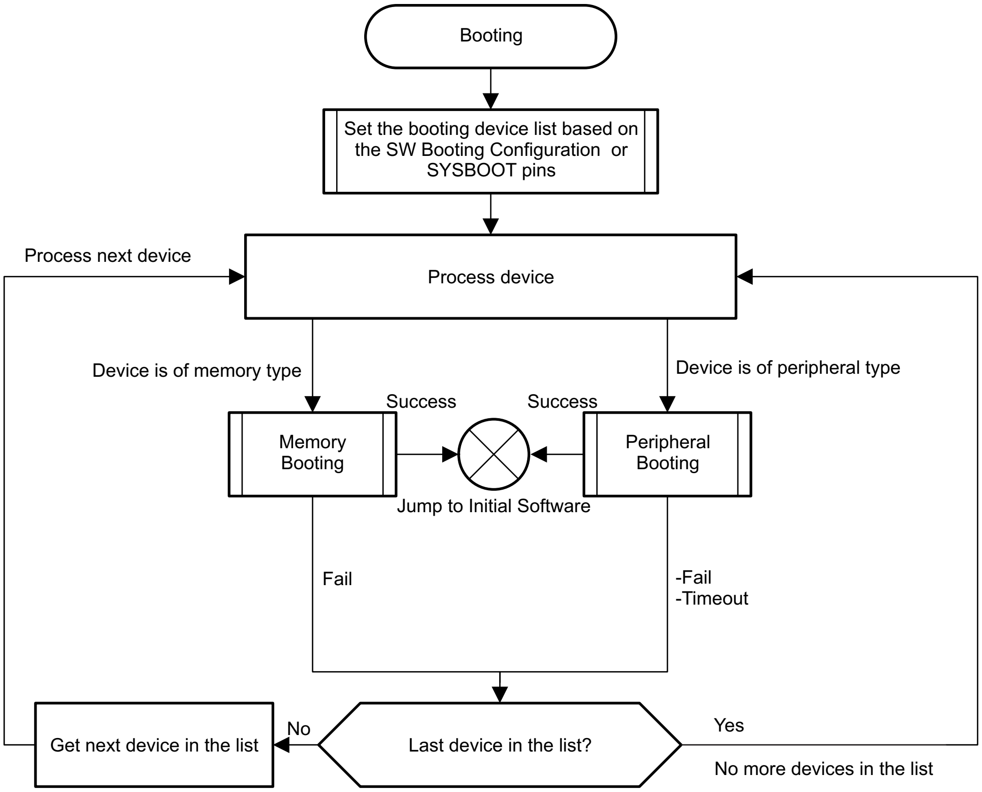
The DPLLs and PRCM clock dividers are configured with the ROM Code default values after cold or warm reset in order to give the same working conditions to the Public ROM Code sequence.

5.2.5 Booting

5.2.5.1 Overview

Figure 5-7 shows the booting procedure. First a booting device list is created. The list consists of all devices which will be searched for a booting image. The list is filled in based on the SYSBOOT pins.

Figure 5-7. ROM Code Booting Procedure



Once the booting device list is set up, the booting routine examines the devices enumerated in the list sequentially and executes either the memory booting or peripheral booting procedure depending on the booting device type.

- The memory booting procedure executes when the booting device type is one of NOR, NAND, MMC, SPI-EEPROM, QSPI-EEPROM or USB Pen drive (Mass Storage Class).

  The memory booting procedure reads data from a memory type device. If a valid booting image is found and successfully read from the external memory device:

  - When ROM transfers control to the ISW, it passes a parameter to a Boot Parameter Structure in R0. The Boot Parameter Structure can be used to determine the boot device, reset reason, etc. The fields of this structure are described in Table 5-9.

- The peripheral booting executes when the booting device type is Ethernet, USB Client Mode or UART.

  The peripheral booting procedure downloads data from a host (commonly a PC) to the device by Ethernet, USB Client Mode, or UART links. The ROM Code uses a host-slave logical protocol for synchronization. Upon successful UART, USB Client Mode or Ethernet connection the host sends the image binary contents. The peripheral booting procedure is described in Section 5.2.7, *Peripheral Booting*.

Table 5-9. Booting Parameters Structure

Offset	Field	Size (bytes)	Description
00h	Reserved	4	Reserved
04h	Device Descriptor Address	4	Pointer to the memory device descriptor that has been used during the memory booting process (used internally for ROM testing).
08h	Current Booting Device	1	Code of device used for booting: 00h – void, no device 01h – NOR 02h – NOR (wait monitoring on) 03h – NOR2 04h – NOR2 (wait monitoring on) 05h – NAND 06h – NAND with I2C 07h – MMC/SD port 0 08h – MMC/SD port 1 0Ah – SPI 0Bh - QSPI 0Ch - SPI2 0Dh - USB_MS 41h – UART0 45h – USB_CL 47h – CPGMAC0
09h	Reset Reason	1	Current reset reason bit mask (bit=1-event present): [0] – Power-on reset [1] – Global software reset [2] – Security violation reset [4] – Watchdog1 reset [5] – External warm reset [9] – IcePick reset Other bits – Reserved
0Ah	Reserved	10	Reserved

5.2.5.1.1 Device List

The ROM Code creates the device list based on information gathered from the SYSBOOT configuration pins sensed in the control module. The pins are used to index the device table from which the list of devices is extracted

5.2.5.2 SYSBOOT Configuration Pins

Table 5-10. SYSBOOT Configuration Pins

SYS BOOT [18](1)(2)	SYS BOOT [17]	SYS BOOT [16](3)	SYS BOOT [15:14]	SYS BOOT [13:12]	SYS BOOT [11]	SYS BOOT [9]	SYS BOOT [8]	SYS BOOT [7]	SYS BOOT [6]	SYS BOOT [5]	SYS BOOT [4:0]				
ALL boot modes: CLKOUT2 output	**ALL boot modes:** CLKOUT1 output enabled or disabled on XDMA_EVENT_INTR0	**USB_MS and USB_CL:** DP/DM swapping	**ALL boot modes:** Crystal Frequency (MHz)	**ALL boot modes:** Set to 00b for normal operation (4)	**Fast NOR or NOR:** Muxed or non-muxed device	**Fast NOR or NOR:** WAIT enable	**Fast NOR:** Must be 1 NAND or NAND_I2C or NOR: Wait mux option	**QSPI** Width Selection or **NOR** Pinmux	**NOR** Pinmux or **QSPI** Pinmux or **NAND/ NAND_I2C** ECC	**EMAC:** PHY interface Type		Boot Sequence			
CTRL_STS[26]	CTRL_STS[25]	CTRL_STS[24]	CTRL_STS [23:22]	CTRL_STS [21:20]	CTRL_STS[19]	CTRL_STS[17]	CTRL_STS[16]	CTRL_STS[7:6]		CTRL_STS[5]	CTRL_STS [4:0]				
												1	2	3	4
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	USB_MS or USB_CL: 0b = USB DP/DM not swapped 1b = USB DP/DM swapped	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	NAND: 0b = Wait mux option 0 1b = Wait mux option 1		NAND: 0b = ECC done by ROM 1b = ECC done by NAND	don't care	00000b	NAND	USB_MS (USB1)	MMC0	USB_CL (USB0)
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	USB_MS or USB_CL: 0b = USB DP/DM not swapped 1b = USB DP/DM swapped	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	don't care	don't care		don't care	00001b	MMC0	MMC1	USB_MS (USB1)	USB_CL (USB0)

---

(1) SYSBOOT[18] can be used to supply a clock for external ethernet PHY devices. If SYSBOOT[5]=0, EXTDEV_PLL will be configured to 25MHz, if SYSBOOT[5]=1, EXTDEV_PLL will be configured to 50MHz.

(2) The functionality provided by SYSBOOT[18] is only available in PG1.2 silicon.

(3) The functionality provided by SYSBOOT[16] is only available in PG1.2 silicon

(4) All other values reserved.

Table 5-10. SYSBOOT Configuration Pins (continued)

SYS BOOT [18][1][2]	SYS BOOT [17]	SYS BOOT [16][3]	SYS BOOT [15:14]	SYS BOOT [13:12]	SYS BOOT [11]	SYS BOOT [9]	SYS BOOT [8]	SYS BOOT [7]	SYS BOOT [6]	SYS BOOT [5]	SYS BOOT [4:0]				
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	USB_MS or USB_CL: 0b = USB DP/DM not swapped 1b = USB DP/DM swapped	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	don't care	don't care		don't care	00010b	SPI	USB_MS (USB1)	MMC0	USB_CL (USB0)
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	USB_MS or USB_CL: 0b = USB DP/DM not swapped 1b = USB DP/DM swapped	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	NAND_I2C: 0b = Wait mux option 0 1b = Wait mux option 1		NAND_I2C: 0b = ECC done by ROM 1b = ECC done by NAND	don't care	00011b	NAND_I2C	USB_MS (USB1)	MMC0	USB_CL (USB0)
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	USB_MS or USB_CL: 0b = USB DP/DM not swapped 1b = USB DP/DM swapped	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	don't care	don't care		don't care	00100b	MMC1	MMC0	USB_MS (USB1)	USB_CL (USB0)
0b = Do not route EXTDEV_PLL to CLKOUT2 1b = Route EXTDEV_PLL to CLKOUT2	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	USB_MS or USB_CL: 0b = USB DP/DM not swapped 1b = USB DP/DM swapped	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	0b = non-muxed device 1b = muxed device	NOR: 0 = WAIT signal not monitored during boot 1 = WAIT signal monitored during boot	NOR: 0b = Wait mux option 0 1b = Wait mux option 1	NOR: 00b = pinmux option 0 01b = pinmux option 1 10b = pinmux option 2 11b = reserved		EMAC: 0b = MII 1b = RMII	00101b	NOR	USB_MS (USB1)	EMAC1	USB_CL (USB0)

Table 5-10. SYSBOOT Configuration Pins (continued)

SYS BOOT [18](1)(2)	SYS BOOT [17]	SYS BOOT [16](3)	SYS BOOT [15:14]	SYS BOOT [13:12]	SYS BOOT [11]	SYS BOOT [9]	SYS BOOT [8]	SYS BOOT [7]	SYS BOOT [6]	SYS BOOT [5]	SYS BOOT [4:0]				
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	don't care	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	NAND: 0b = Wait mux option 0 1b = Wait mux option 1		NAND: 0b = ECC done by ROM 1b = ECC done by NAND	don't care	00110b	NAND			
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	USB_MS or USB_CL: 0b = USB DP/DM not swapped 1b = USB DP/DM swapped	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	0b = non-muxed device 1b = muxed device	NOR: 0 = WAIT signal not monitored during boot 1 = WAIT signal monitored during boot	NOR: 0b = Wait mux option 0 1b = Wait mux option 1	NOR: 00b = pinmux option 0 01b = pinmux option 1 10b = pinmux option 2 11b = reserved		don't care	00111b	NOR	USB_MS (USB1)	UART0	
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	USB_MS or USB_CL: 0b = USB DP/DM not swapped 1b = USB DP/DM swapped	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	don't care	QSPI: 0b = quad read 1b = single read	QSPI: 0b = pinmux option 0 1b = pinmux option 1	don't care	01000b	QSPI	USB_MS (USB1)	MMC0	USB_CL (USB0)
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	don't care	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	don't care	don't care		don't care	01001b	SPI			

Table 5-10. SYSBOOT Configuration Pins (continued)

SYSBOOT [18](1)(2)	SYSBOOT [17]	SYSBOOT [16](3)	SYSBOOT [15:14]	SYSBOOT [13:12]	SYSBOOT [11]	SYSBOOT [9]	SYSBOOT [8]	SYSBOOT [7]	SYSBOOT [6]	SYSBOOT [5]	SYSBOOT [4:0]				
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	don't care	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	don't care	QSPI: 0b = quad read 1b = single read	QSPI: 0b = pinmux option 0 1b = pinmux option 1	don't care	01010b	QSPI			
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	don't care	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	NAND_I2C: 0b = Wait mux option 0 1b = Wait mux option 1		NAND_I2C: 0b = ECC done by ROM 1b = ECC done by NAND	don't care	01011b	NAND_I2C			
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	don't care	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	don't care	don't care		don't care	01100b	MMC0			
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	don't care	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	don't care	don't care		don't care	01101b	MMC1			
0b = Do not route EXTDEV_PLL to CLKOUT2 1b = Route EXTDEV_PLL to CLKOUT2	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	USB_MS or USB_CL: 0b = USB DP/DM not swapped 1b = USB DP/DM swapped	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	NAND_I2C: 0b = Wait mux option 0 1b = Wait mux option 1		NAND_I2C: 0b = ECC done by ROM 1b = ECC done by NAND	EMAC: 0b = MII 1b = RMII	01110b	NAND_I2C	USB_MS (USB1)	EMAC1	UART0

Table 5-10. SYSBOOT Configuration Pins (continued)

SYS BOOT [18](1)(2)	SYS BOOT [17]	SYS BOOT [16](3)	SYS BOOT [15:14]	SYS BOOT [13:12]	SYS BOOT [11]	SYS BOOT [9]	SYS BOOT [8]	SYS BOOT [7]	SYS BOOT [6]	SYS BOOT [5]	SYS BOOT [4:0]				
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	don't care	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	0b = non-muxed device 1b = muxed device	fast_NOR: 0 = WAIT signal not monitored during boot 1 = WAIT signal monitored during boot	fast NOR: must be 1	don't care		don't care	01111b	FAST_NOR			
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	USB_MS or USB_CL: 0b = USB DP/DM not swapped 1b = USB DP/DM swapped	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	NAND: 0b = Wait mux option 0 1b = Wait mux option 1		NAND_I2C: 0b = ECC done by ROM 1b = ECC done by NAND	don't care	10000b	MMC0	USB_MS (USB1)	USB_CL (USB0)	NAND
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	USB_MS or USB_CL: 0b = USB DP/DM not swapped 1b = USB DP/DM swapped	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	don't care	don't care		don't care	10001b	MMC1	USB_MS (USB1)	USB_CL (USB0)	MMC0
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	USB_MS or USB_CL: 0b = USB DP/DM not swapped 1b = USB DP/DM swapped	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	don't care	don't care		don't care	10010b	MMC0	USB_MS (USB1)	USB_CL (USB0)	SPI

Table 5-10. SYSBOOT Configuration Pins (continued)

SYS BOOT [18][1][2]	SYS BOOT [17]	SYS BOOT [16][3]	SYS BOOT [15:14]	SYS BOOT [13:12]	SYS BOOT [11]	SYS BOOT [9]	SYS BOOT [8]	SYS BOOT [7]	SYS BOOT [6]	SYS BOOT [5]	SYS BOOT [4:0]				
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	USB_MS or USB_CL: 0b = USB DP/DM not swapped 1b = USB DP/DM swapped	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	NAND_I2C: 0b = Wait mux option 0 1b = Wait mux option 1		NAND_I2C: 0b = ECC done by ROM 1b = ECC done by NAND	don't care	10011b	MMC0	USB_MS (USB1)	USB_CL (USB0)	NAND_I2C
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	USB_MS or USB_CL: 0b = USB DP/DM not swapped 1b = USB DP/DM swapped	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	don't care	don't care		don't care	10100b	MMC0	USB_MS (USB1)	USB_CL (USB0)	MMC1
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	USB_MS or USB_CL: 0b = USB DP/DM not swapped 1b = USB DP/DM swapped	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	0b = non-muxed device 1b = muxed device	NOR: 0b = WAIT signal not monitored during boot 1b = WAIT signal monitored during boot	NOR: 0b = wait mux option 0 1b = wait mux option 1	NOR: 00b = pinmux option 0 01b = pinmux option 1 10b = pinmux option 2 11b = reserved		don't care	10101b	USB_MS (USB1)	USB_CL (USB0)	UART0	NOR
0b = Do not route EXTDEV_PLL to CLKOUT2 1b = Route EXTDEV_PLL to CLKOUT2	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	don't care	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	NAND or NAND_I2C: 0b = Wait mux option 0 1b = Wait mux option 1		NAND or NAND_I2C: 0b = ECC done by ROM 1b = ECC done by NAND	EMAC: 0b = MII 1b = RMII	10110b	EMAC1	NAND_I2C	NAND	MMC0

Table 5-10. SYSBOOT Configuration Pins (continued)

SYS BOOT [18](1)(2)	SYS BOOT [17]	SYS BOOT [16](3)	SYS BOOT [15:14]	SYS BOOT [13:12]	SYS BOOT [11]	SYS BOOT [9]	SYS BOOT [8]	SYS BOOT [7]	SYS BOOT [6]	SYS BOOT [5]	SYS BOOT [4:0]				
0b = Do not route EXTDEV_PLL to CLKOUT2 1b = Route EXTDEV_PLL to CLKOUT2	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	don't care	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	don't care	QSPI: 0b = quad read 1b = single read	QSPI: 0b = pinmux option 0 1b = pinmux option 1	EMAC: 0b = MII 1b = RMII	10111b	EMAC1	SPI	QSPI	MMC1
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	USB_MS or USB_CL: 0b = USB DP/DM not swapped 1b = USB DP/DM swapped	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	don't care	QSPI: 0b = quad read 1b = single read	QSPI: 0b = pinmux option 0 1b = pinmux option 1	don't care	11000b	MMC0	USB_MS (USB1)	USB_CL (USB0)	QSPI
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	don't care	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	NAND or NAND_I2C: 0b = Wait mux option 0 1b = Wait mux option 1		NAND or NAND_I2C: 0b = ECC done by ROM 1b = ECC done by NAND	don't care	11001b	UART0	NAND_I2C	NAND	MMC0
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	don't care	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	don't care	QSPI: 0b = quad read 1b = single read	QSPI: 0b = pinmux option 0 1b = pinmux option 1	don't care	11010b	UART0	SPI	QSPI	MMC1

Table 5-10. SYSBOOT Configuration Pins (continued)

SYS BOOT [18](1)(2)	SYS BOOT [17]	SYS BOOT [16](3)	SYS BOOT [15:14]	SYS BOOT [13:12]	SYS BOOT [11]	SYS BOOT [9]	SYS BOOT [8]	SYS BOOT [7]	SYS BOOT [6]	SYS BOOT [5]	SYS BOOT [4:0]				
0b = Do not route EXTDEV_PLL to CLKOUT2 1b = Route EXTDEV_PLL to CLKOUT2	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	don't care	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	0b = non-muxed device 1b = muxed device	NOR: 0 = WAIT signal not monitored during boot 1 = WAIT signal monitored during boot	NOR: 0b = Wait mux option 0 1b = Wait mux option 1	NOR: 00b = pinmux option 0 01b = pinmux option 1 10b = pinmux option 2 11b = reserved		EMAC: 0b = MII 1b = RMII	11011b	EMAC1	UART0	NOR	
0b = Do not route EXTDEV_PLL to CLKOUT2 1b = Route EXTDEV_PLL to CLKOUT2	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	don't care	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	don't care	don't care		EMAC: 0b = MII 1b = RMII	11100b	EMAC1			
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	USB_MS or USB_CL: 0b = USB DP/DM not swapped 1b = USB DP/DM swapped	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	don't care	don't care		don't care	11101b	USB_CL (USB0)			
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	USB_MS or USB_CL: 0b = USB DP/DM not swapped 1b = USB DP/DM swapped	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	don't care	don't care	NAND_I2C: 0b = Wait mux option 0 1b = Wait mux option 1		NAND_I2C: 0b = ECC done by ROM 1b = ECC done by NAND_I2C	don't care	11110b	USB_MS (USB1)	NAND_I2C		

Table 5-10. SYSBOOT Configuration Pins (continued)

SYS BOOT [18][1][2]	SYS BOOT [17]	SYS BOOT [16][3]	SYS BOOT [15:14]	SYS BOOT [13:12]	SYS BOOT [11]	SYS BOOT [9]	SYS BOOT [8]	SYS BOOT [7]	SYS BOOT [6]	SYS BOOT [5]	SYS BOOT [4:0]				
don't care	0b = CLKOUT1 disabled 1b = CLKOUT1 enabled	don't care	00b = 19.2 01b = 24 10b = 25 11b = 26	00b	0b = non-muxed device 1b = muxed device	fast_NOR: 0 = WAIT signal not monitored during boot 1 = WAIT signal monitored during boot	fast NOR: must be 1	don't care		don't care	11111b	FAST_NOR			

General notes for SYSBOOT configuration pins:

- SYSBOOT[17] selects the default mux mode of XDMA_EVENT_INTR0. When SYSBOOT[17]=0, default mux mode of XDMA_EVENT_INTR0 is mode0. When SYSBOOT[17]=1, default mux mode of XDMA_EVENT_INTR0 is mode3, which enables CLKOUT1 on the terminal.
- DSS_HSYNC terminal is SYSBOOT[17] input. DSS_VSYNC terminal is SYSBOOT[16]. DSS_DATA[15:0] terminals are respectively SYSBOOT[15:0] inputs. All are latched on the rising edge of PWRONTSTn. DSS_AC_BIAS_EN terminal is SYSBOOT[18].
- Note that even though some bits may be a don't care for ROM code, all SYSBOOT values are latched into the CTRL_STS register and may be used by software after ROM execution has completed.
- SYSBOOT[10] should always be set to 0 for normal operation. SYSBOOT[10] = 1 is reserved.

The ROM Code uses the row pointed by the SYSBOOT pins value. The device list is filled in with the 1st to 4th devices.

Table 5-10 is the decoding table for SYSBOOT pin configuration. The following shortcuts are used in the table:

- MMC1: MMC or SD card (MMC port 1)
- MMC0: MMC or SD card (MMC port 0)
- NAND_I2C: NAND flash memory / read geometry from EEPROM on I2C0
- NOR: NOR device with or without wait monitoring (5)
- UART: UART interface (UART port 0)
- EMAC: Ethernet interface (EMAC port 1)
- SPI: SPI EEPROM (SPI 0, CS0)
- USB_CL: USB Client Mode (USB0 interface)
- USB_MS: USB Mass Storage Class (USB1 interface)
- QSPI: Quad SPI interface (QSPI, CS0)

NOTE: For any SYSBOOT value that is selected, please be aware of the pin muxing implications. For example, if the boot mode selected is EMAC, NAND, SPI, NANDI2C, the SOC will drive EMAC, GPMC, SPI and I2C pins, in that order, depending on which boot device finally succeeds. For specific details of the pins driven by each device, see the description of the boot device in this document.

To extend the boot flow to boot from devices that are not natively supported by the ROM, or if the user needs a different booting configuration, use USB_MS boot. For example, if a customer wants to boot from an unsupported NAND device or UART with different configuration, the system can be configured to boot from another inexpensive boot media like SPI flash and the code for configuring and booting from the unsupported NAND device can be loaded into the SPI flash. This is known as a secondary boot.

A “Low Latency NOR boot mechanism” (Section 5.2.8, *Low Latency NOR Booting*) is provided where minimal execution is performed from ROM Code for configuring the GPMC interface and then directly jump to the code contained in the connected NOR flash device.

5.2.6 Memory Booting

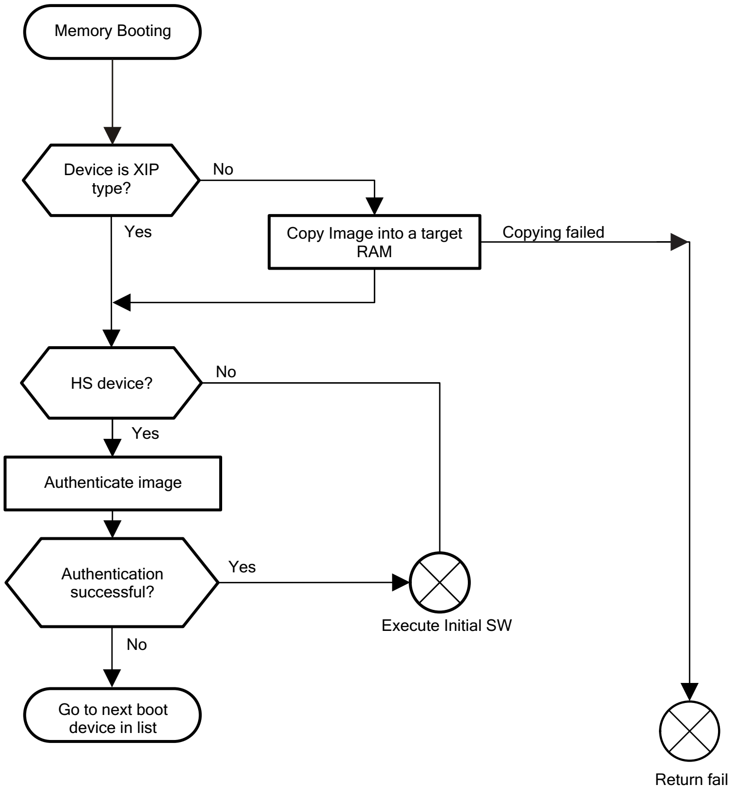
5.2.6.1 Overview

The memory booting procedure executes external code located in memory device types.

---

(5) Wait monitoring is enabled or disabled based on SYSBOOT pins in Table 5-10.

Figure 5-8. Memory Booting



The following memory booting devices are supported:

- MMC/SD cards
- NOR flash
- NAND flash
- SPI EEPROMs
- Quad SPI EEPROMs
- USB Mass Storage device

Two groups of permanent booting devices are distinguished by the need of code shadowing. The code shadowing means copying a code from an indirectly addressable device into a location (typically a RAM area) from where the code can be executed. Devices which are directly addressable are called eXecute In Place (XIP) devices.

Figure 5-8 shows the memory booting flowchart. The Image execution step is about performing the shadowing of the image that is copying the image from external mass storage (non-XIP) into internal RAM. The next sections detail procedures for device initialization and detection in addition to the description of the sector read routine for each supported device type. A sector is a logical unit of 512 bytes.

The detection of whether an image is present on a selected device differs by device type:

- On a GP Device type, a booting image (bootloader) is considered to be present when the first four bytes of the sector are not equal to 00000000h or FFFFFFFFh.

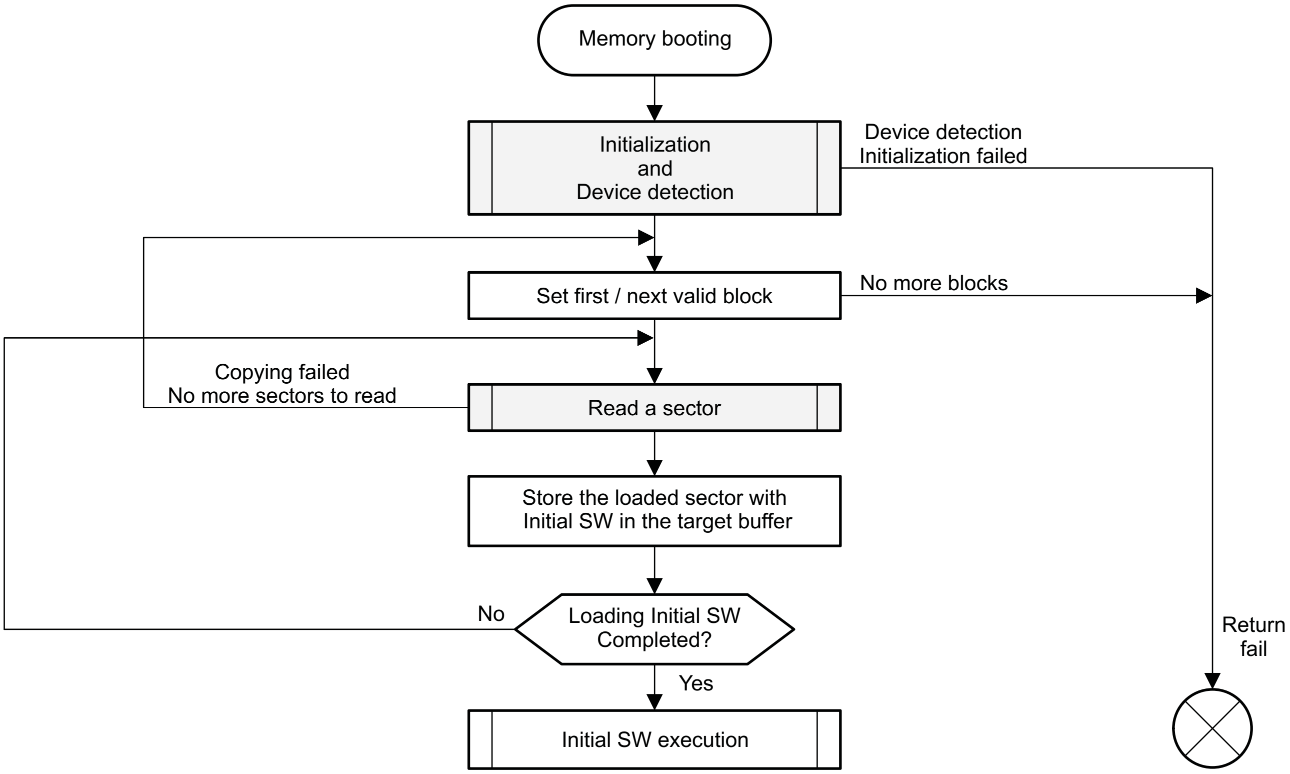
During the first read sector call, the first sector is copied to a temporary RAM buffer. Once the image is found and the destination address is known from the decoding the header, the content of the temporary buffer is moved to the target RAM location so it is not needed to re-read the first image sector. On a GP device the GP header is discarded, and only executable code is located in RAM with the first executable instruction located at the destination address.

MMC/SD cards (in raw mode), SPI, and NAND devices hold up to four copies of the booting image. The ROM Code searches for one valid image out of the four, if present, by walking over the first four blocks of the mass storage space. Other XIP devices (NOR) use only one copy of the booting image.

5.2.6.2 Image Shadowing for non-XIP Memories

The GP device shadowing uses the following approach.

Figure 5-9. Image Shadowing on GP Device



5.2.6.3 NOR Memory

The ROM Code can boot directly from NOR devices. A NOR flash memory behaves as an eXecute In Place (XIP) device. NOR flash devices are supported under the following assumptions:

- GPMC is the communication interface.
- Connect up to 1Gb (128MB) memories.
- Only x16 data bus width (x8 not supported).
- Asynchronous protocol.
- Supports address / data multiplexed mode and non-muxed mode.
- GPMC clock is 50 MHz.
- Device connected to CS0 mapped to address 0x08000000h.
- Wait pin signal WAIT0 is monitored or ignored depending on the SYSBOOT pin configuration mentioned in Table 5-10.
- Flexible muxing options for gpmc address lines for non-muxed and muxed NOR devices based on SYSBOOT pin configuration mentioned in Table 5-10.

Depending on the SYSBOOT pins, the GPMC is configured to use the WAIT signal connected on the WAIT pin or not. Wait pin polarity is set to stall accessing memory when the WAIT pin is low. The wait monitoring is intended to be used with memories which require long time for initialization after reset or need to pause while reading data.

The boot procedure from a NOR device is:

1. Configure GPMC for NOR device access.
2. Set the image location to 0x08000000h
3. Verify if a bootable image is present at the image location.
4. If the image is not found, return from NOR booting to the main booting loop.

5.2.6.3.1 NOR Initialization and Detection

5.2.6.3.1.1 GPMC Initialization

Figure 5-10 and Table 5-11 describe the GPMC timing settings configured for NOR boot and other address-data accessible devices.

Figure 5-10. GPMC NOR Timings

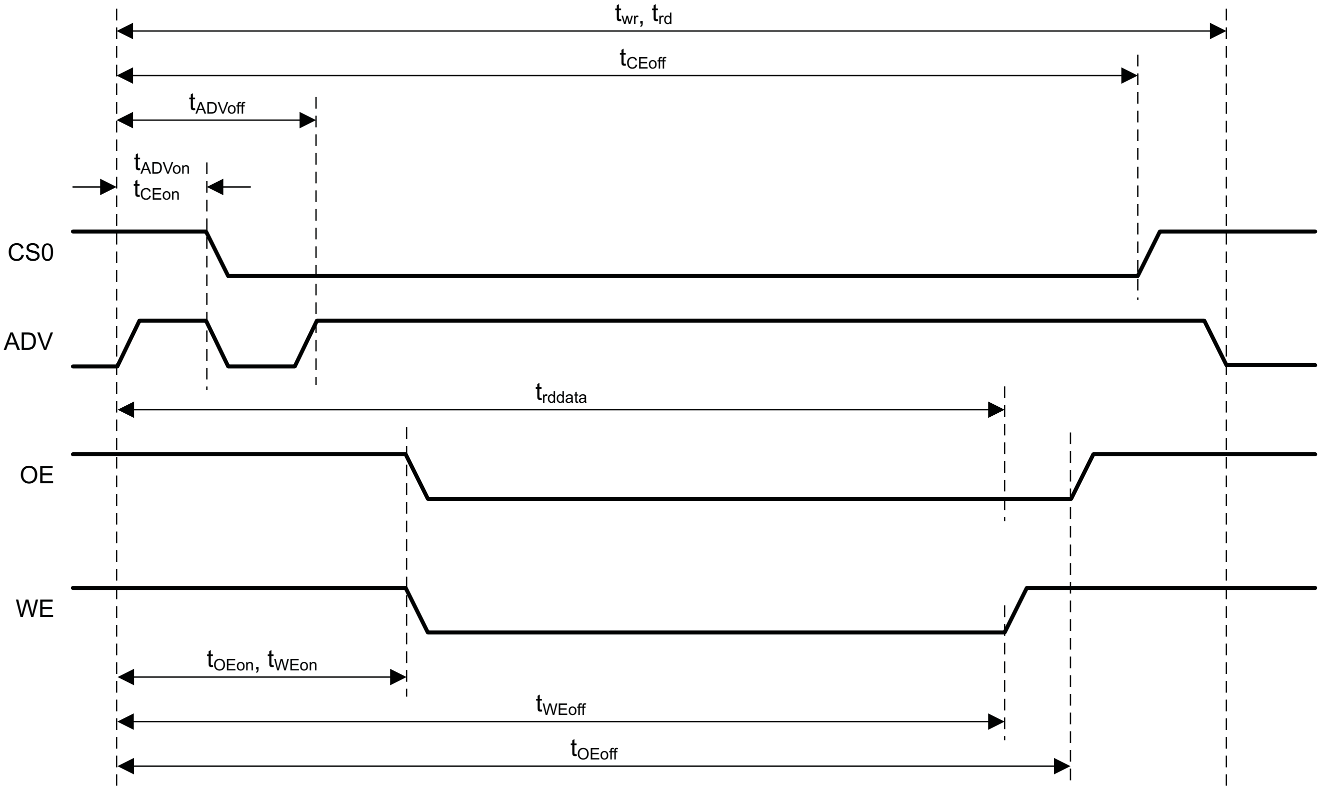


Table 5-11. NOR Timings Parameters

Parameter	Description	Value (clock cycles)
t_{wr}	write cycle period	17
t_{rd}	read cycle period	17
t_{CEon}	CE low time	1
t_{CEoff}	CE high time	16
t_{ADVon}	ADV low time	1
t_{ADVoff}	ADV high time	2
t_{OEon}	OE low time	3
t_{WEon}	WE low time	3
t_{rddata}	data latch time	15
t_{OEoff}	OE high time	16
t_{WEoff}	WE high time	15

5.2.6.3.1.2 Device Detection

There is no specific identification routine executed prior to booting from an NOR device.

5.2.6.3.2 Pins Used

Table 5-12 lists the pins configured by the ROM for NOR boot mode. Not all pins are driven at boot time. The decision as to which pins need to be driven is based on the type of NOR flash selected.

Table 5-12. Pins Used for NOR Boot Common Signals

Signal Name	Pin Used in NOR	CTRL_CONF Register	Register Setting
cs0	gpmc_cs0	CTRL_CONF_GPMC_CS0	0x00010000
advn_ale	gpmc_advn_ale	CTRL_CONF_GPMC_ADVN_ALE	0x00010000
oen_ren	gpmc_oen_ren	CTRL_CONF_GPMC_OEN_REN	0x00010000
be1n	gpmc_be1n	CTRL_CONF_GPMC_BE1N	0x00010000
be0n_cle	gpmc_be0n_cle	CTRL_CONF_GPMC_BE0N_CLE	0x00010000
wen	gpmc_wen	CTRL_CONF_GPMC_WEN	0x00010000
clk	gpmc_clk	CTRL_CONF_GPMC_CLK	0x00060000

Table 5-13. Pins Used for NOR Boot Wait Pin Selection

Signal name	Pin used pinmux option 0	CTRL_CONF Register	Register Setting	Pin used pinmux option 1	CTRL_CONF Register	Register Setting
wait	gpmc_wait0	CTRL_CONF_ GPMC_WAIT	0x00060000	gpmc_csn3	CTRL_CONF_ GPMC_CSN3	0x00060001

Table 5-14. Pins Used for non-Mux NOR Boot

Signal Name	Pin Used in non-Mux-NOR Pinmux Option 0	CTRL_CONF Register	Register Setting	Pin Used in non-Mux-NOR Pinmux Option 1	CTRL_CONF Register	Register Setting
a0	dss_data0	CTRL_CONF_ DSS_DATA0	0x00000001	gpmc_a0	CTRL_CONF_ GPMC_A0	0x00000000
a1	dss_data1	CTRL_CONF_ DSS_DATA1	0x00000001	gpmc_a1	CTRL_CONF_ GPMC_A1	0x00000000
a2	dss_data2	CTRL_CONF_ DSS_DATA2	0x00000001	gpmc_a2	CTRL_CONF_ GPMC_A2	0x00000000
a3	dss_data3	CTRL_CONF_ DSS_DATA3	0x00000001	gpmc_a3	CTRL_CONF_ GPMC_A3	0x00000000

Table 5-14. Pins Used for non-Mux NOR Boot (continued)

Signal Name	Pin Used in non-Mux-NOR Pinmux Option 0	CTRL_CONF Register	Register Setting	Pin Used in non-Mux-NOR Pinmux Option 1	CTRL_CONF Register	Register Setting
a4	dss_data4	CTRL_CONF_ DSS_DATA4	0x00000001	gpmc_a4	CTRL_CONF_ GPMC_A4	0x00000000
a5	dss_data5	CTRL_CONF_ DSS_DATA5	0x00000001	gpmc_a5	CTRL_CONF_ GPMC_A5	0x00000000
a6	dss_data6	CTRL_CONF_ DSS_DATA6	0x00000001	gpmc_a6	CTRL_CONF_ GPMC_A6	0x00000000
a7	dss_data7	CTRL_CONF_ DSS_DATA7	0x00000001	gpmc_a7	CTRL_CONF_ GPMC_A7	0x00000000
a8	dss_vsync	CTRL_CONF_ DSS_VSYNC	0x00000001	gpmc_a8	CTRL_CONF_ GPMC_A8	0x00000000
a9	dss_hsync	CTRL_CONF_ DSS_HSYNC	0x00000001	gpmc_a9	CTRL_CONF_ GPMC_A9	0x00000000
a10	dss_pclk	CTRL_CONF_ DSS_PCLK	0x00000001	gpmc_a10	CTRL_CONF_ GPMC_A10	0x00000000
a11	dss_ac_bias_en	CTRL_CONF_ DSS_AC_BIAS_ EN	0x00000001	gpmc_a11	CTRL_CONF_ GPMC_A11	0x00000000
a12	dss_data8	CTRL_CONF_ DSS_DATA8	0x00000001	dss_data8	CTRL_CONF_ DSS_DATA8	0x00000001
a13	dss_data9	CTRL_CONF_ DSS_DATA9	0x00000001	dss_data9	CTRL_CONF_ DSS_DATA9	0x00000001
a14	dss_data10	CTRL_CONF_ DSS_DATA10	0x00000001	dss_data10	CTRL_CONF_ DSS_DATA10	0x00000001
a15	dss_data11	CTRL_CONF_ DSS_DATA11	0x00000001	dss_data11	CTRL_CONF_ DSS_DATA11	0x00000001
a16	dss_data12	CTRL_CONF_ DSS_DATA12	0x00000001	dss_data12	CTRL_CONF_ DSS_DATA12	0x00000001
a17	dss_data13	CTRL_CONF_ DSS_DATA13	0x00000001	dss_data13	CTRL_CONF_ DSS_DATA13	0x00000001
a18	dss_data14	CTRL_CONF_ DSS_DATA14	0x00000001	dss_data14	CTRL_CONF_ DSS_DATA14	0x00000001
a19	dss_data15	CTRL_CONF_ DSS_DATA15	0x00000001	dss_data15	CTRL_CONF_ DSS_DATA15	0x00000001
a20	mmc0_dat3	CTRL_CONF_ MMC0_DAT3	0x08040004	mmc0_dat3	CTRL_CONF_ MMC0_DAT3	0x08040001
a21	mmc0_dat2	CTRL_CONF_ MMC0_DAT2	0x08040004	mmc0_dat2	CTRL_CONF_ MMC0_DAT2	0x08040001
a22	mmc0_dat1	CTRL_CONF_ MMC0_DAT1	0x08040001	mmc0_dat1	CTRL_CONF_ MMC0_DAT1	0x08040001
a23	mmc0_dat0	CTRL_CONF_ MMC0_DAT0	0x08040001	mmc0_dat0	CTRL_CONF_ MMC0_DAT0	0x08040001
a24	mmc0_clk	CTRL_CONF_ MMC0_CLK	0x08040001	mmc0_clk	CTRL_CONF_ MMC0_CLK	0x08040001
a25	mmc0_cmd	CTRL_CONF_ MMC0_CMD	0x08040001	mmc0_cmd	CTRL_CONF_ MMC0_CMD	0x08040001
a26	gpmc_a10 (pinmux mode 4)	CTRL_CONF_ GPMC_A10	0x0804004	-		
a27	gpmc_a11 (pinmux mode 4)	CTRL_CONF_ GPMC_A11	0x0804004	-		
d0 - d15	gpmc_ad0 – gpmc_ad15	CTRL_CONF_ GPMC_AD0 to CTRL_CONF_ GPMC_AD15	0x00060000	gpmc_ad0 - gpmc_ad15	CTRL_CONF_ GPMC_AD0 to CTRL_CONF_ GPMC_AD15	0x00060000

Table 5-15. Pins Used for Mux NOR Boot

Signal Name	Pin Used in Mux NOR Pinmux Option 0	CTRL_CONF Register	Register Setting	Pin Used in Mux NOR Pinmux Option 1	CTRL_CONF Register	Register Setting	Pin Used in Mux NOR Pinmux Option 2	CTRL_CONF Register	Register Setting
d0 – d15 / a1 – a16	gpmc_ad0 - gpmc_ad15	CTRL_CONF_GPMC_AD0 to CTRL_CONF_GPMC_AD15	0x00060000	gpmc_ad0 - gpmc_ad15	CTRL_CONF_GPMC_AD0 to CTRL_CONF_GPMC_AD15	0x00060000	gpmc_ad0 - gpmc_ad15	CTRL_CONF_GPMC_AD0 to CTRL_CONF_GPMC_AD15	0x00060000
a17	dss_data1	CTRL_CONF_DSS_DATA1	0x00000001	gpmc_a1	CTRL_CONF_GPMC_A1	0x00000000	dss_vsync	CTRL_CONF_DSS_VSYNC	0x00000002
a18	dss_data2	CTRL_CONF_DSS_DATA2	0x00000001	gpmc_a2	CTRL_CONF_GPMC_A2	0x00000000	dss_hsync	CTRL_CONF_DSS_HSYNC	0x00000002
a19	dss_data3	CTRL_CONF_DSS_DATA3	0x00000001	gpmc_a3	CTRL_CONF_GPMC_A3	0x00000000	dss_pclk	CTRL_CONF_DSS_PCLK	0x00000002
a20	dss_data4	CTRL_CONF_DSS_DATA4	0x00000001	gpmc_a4	CTRL_CONF_GPMC_A4	0x00000000	dss_ac_bias_en	CTRL_CONF_DSS_AC_BIAS_EN	0x00000002
a21	dss_data5	CTRL_CONF_DSS_DATA5	0x08040001	gpmc_a5	CTRL_CONF_GPMC_A5	0x08040000	gpmc_be0n_cle	CTRL_CONF_GPMC_BE0N	0x00000002
a22	dss_data6	CTRL_CONF_DSS_DATA6	0x08040001	gpmc_a6	CTRL_CONF_GPMC_A6	0x08040000	gpmc_a6	CTRL_CONF_GPMC_A6	0x08040000
a23	dss_data7	CTRL_CONF_DSS_DATA7	0x08040001	gpmc_a7	CTRL_CONF_GPMC_A7	0x08040000	gpmc_a7	CTRL_CONF_GPMC_A7	0x08040000
a24	dss_vsync	CTRL_CONF_DSS_VSYNC	0x08040001	gpmc_a8	CTRL_CONF_GPMC_A8	0x08040000	gpmc_a8	CTRL_CONF_GPMC_A8	0x08040000
a25	dss_hsync	CTRL_CONF_DSS_HSYNC	0x08040001	gpmc_a9	CTRL_CONF_GPMC_A9	0x08040000	gpmc_a9	CTRL_CONF_GPMC_A9	0x08040000
a26	dss_pclk	CTRL_CONF_DSS_PCLK	0x08040001	gpmc_a10	CTRL_CONF_GPMC_A10	0x08040000	gpmc_a10	CTRL_CONF_GPMC_A10	0x08040000
a27	dss_ac_bias_en	CTRL_CONF_DSS_AC_BIAS_EN	0x08040001	gpmc_a11	CTRL_CONF_GPMC_A11	0x08040000	gpmc_a11	CTRL_CONF_GPMC_A11	0x08040000

5.2.6.3.2.1 SYSBOOT Signals

Table 5-16 describes the SYSBOOT signals relevant to NOR boot.

Table 5-16. SYSBOOT Signals for NOR Boot

SYSBOOT[7:6]	Used for Pinmux option selection in NOR 00b – pinmux option 0 is selected for NOR 01b – pinmux option 1 is selected for NOR 1xb – pinmux option 2 is selected for NOR
SYSBOOT[8]	Decides which pin WAIT needs to be connected to the NOR flash. 0b – Wait mux option 0 1b – Wait mux option 1

Table 5-16. SYSBOOT Signals for NOR Boot (continued)

SYSBOOT[9]	During NOR Boot this pin will be used to determine if Wait is enabled. 0b – Ignore WAIT input 1b – Use WAIT input
SYSBOOT[11]	Address Muxing 0b – No Addr/Data Muxing 1b – Addr/Data Muxing

5.2.6.4 NAND Memory

The NAND flash memory is not XIP and requires shadowing before the code can be executed.

5.2.6.4.1 Features

- GPMC is the communication interface
- Device size from 512Mb (64MB)
- x8 and x16 bus width
- Support for the following page sizes:
  - 2048 bytes + 64 spare bytes
  - 4096 bytes + 128 or 218 spare bytes
- Only supports devices where chip select can be de-asserted during read, program, or erase cycles, without interrupting the operation
- Single Level Cell (SLC) and Multiple Level Cell (MLC) devices
- Device Identification based on ONFI or ROM table
- ECC correction: 8 bits per sector for most devices (16 bits per sector for devices with large spare area)
- Support for disabling ECC correction, so than the in-built ECC correction mechanisms on some NANDs can be used.
- GPMC timings adjusted for NAND access
- GPMC clock is 50 MHz
- Device connected to CS0
- Wait pin signal WAIT0 connected to NAND BUSY output based on SYSBOOT[8] (1)
- Four physical blocks are searched for an image. The block size depends on device.

5.2.6.4.2 Initialization and Detection

The initialization routine for NAND devices includes three parts: GPMC initialization, device detection with parameters determination, and bad block detection.

5.2.6.4.2.1 ONFI Support

The NAND identification starts with ONFI detection.

5.2.6.4.2.2 GPMC Initialization

The GPMC interface is configured to NAND devices. The address bus is released since a NAND device does not use it. The data bus width is initially set to 8 bits and changed to 16 bits if needed after device parameters determination. The following scheme is applied since NAND devices require different timings when compared to regular NOR devices:

---

(1) See Table 5-10, *SYSBOOT Configuration Pins*.

Figure 5-11. GPMC NAND Timings

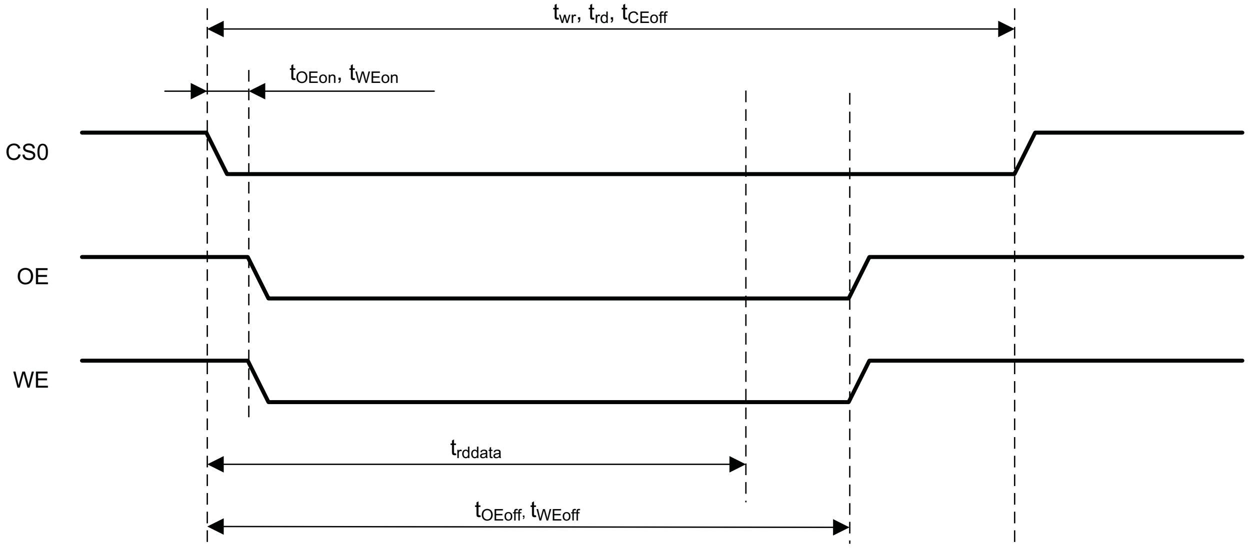


Table 5-17. Parameters for NAND Timings

Parameter	Description	Value (clock cycles)
t_{wr}	write cycle period	30
t_{rd}	read cycle period	30
t_{CEon}	CE low (not marked on the figure)	0
t_{OEon}	CE low to OE low time	7
t_{WEon}	CE low to WE low time	5
t_{rddata}	CE low to data latch time	21
t_{OEoff}	CE low to OE high time	24
t_{WEoff}	CE low to WE high time	22

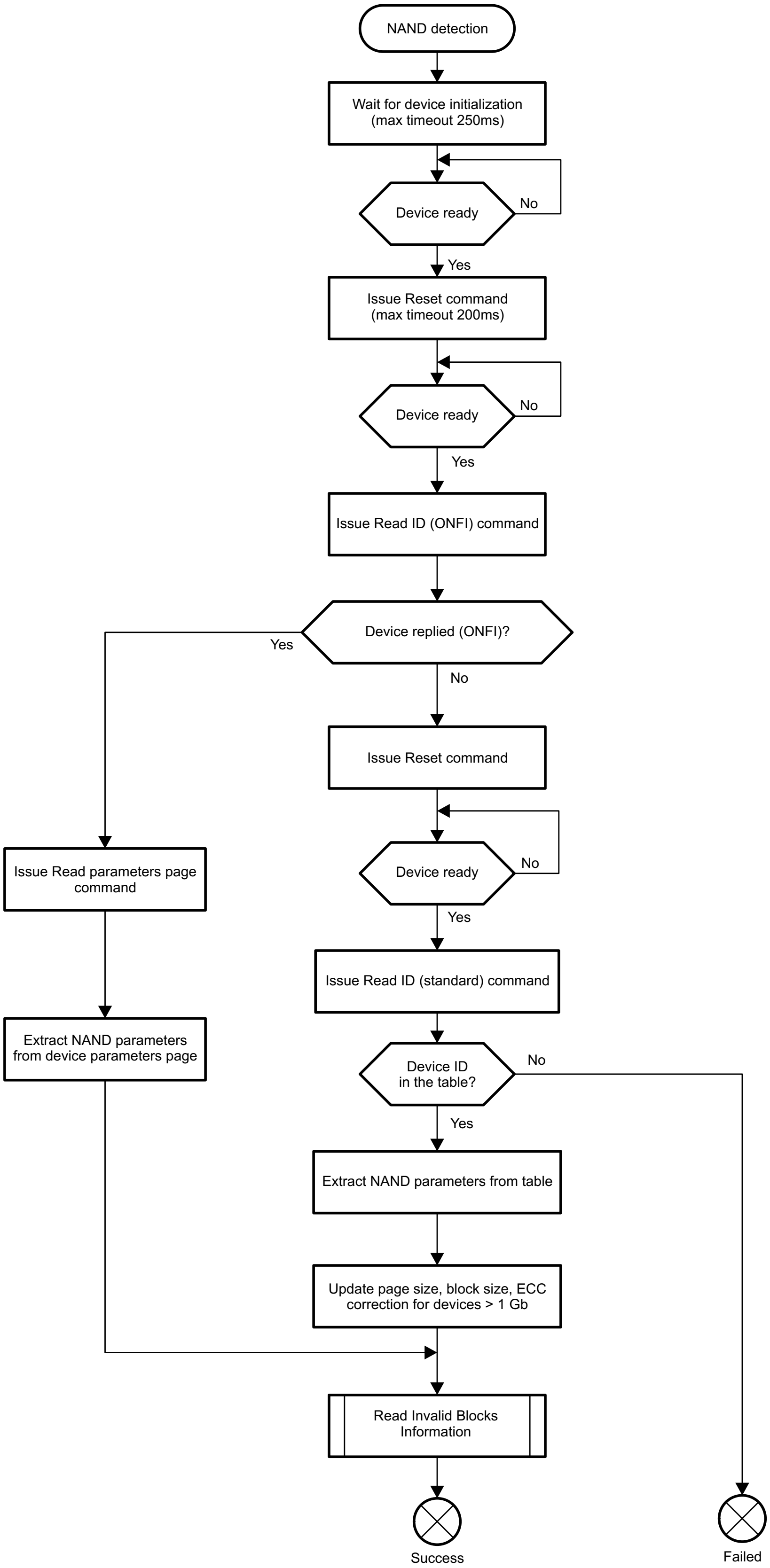
Figure 5-11 and Table 5-17 describe the timings configured for NAND device access.

5.2.6.4.2.3 Device Detection and Parameters

The ROM Code first performs an initial wait for device auto initialization (with a 250 ms timeout) with polling of the ready information. It then identifies the NAND type connected to the GPMC interface. The GPMC is initialized using 8-bit asynchronous mode. The NAND device is reset (command FFh) and its status is polled until ready for operation (with a 200 ms timeout). The ONFI Read ID (command 90h / address 20h) is sent to the NAND device. If it replies with the ONFI signature (4 bytes), then a Read parameters page (command ECh) is sent. If the parameters page does not have the ONFI signature, then the ONFI identification fails.

If the ONFI identification passes, the information shown in Table 5-18 is then extracted: page size, spare area size, number of pages per block, and the addressing mode. The remaining data bytes from the parameters page stream are simply ignored. The detection procedure is described in Figure 5-12. Once the device has been successfully detected, the ROM Code changes GPMC to 16-bit bus width if necessary.

Figure 5-12. NAND Device Detection



NOTE: Timeouts are based on ONFI timing requirements and NAND reset timing specifications.

Table 5-18. ONFI Parameters Page Description

Offset	Description	Size (bytes)
6	Features supported	2
80	Number of data bytes per page	4
84	Number of spare bytes per page	2
92	Number of pages per block	4
101	Number of address cycles	1

If the ONFI Read ID command fails (it will fail with any device not supporting ONFI) then the device is reset again with polling for device to be ready (with a 200 ms timeout). Then, the standard Read ID (command 90h / address 00h) is sent. If the Device ID (2nd byte of the ID byte stream) is recognized as a supported device, then the device parameters are extracted from an internal ROM Code table. The list of supported devices is in Table 5-19.

Table 5-19. Supported NAND Devices

Capacity	Device ID	Bus Width	Page Size
512 Mb	F0	x8	2048
	C0	x16	
	A0	x8	
	B0	x16	
	F2	x8	
	C2	x16	
	A2	x8	
	B2	x16	
1 Gb	F1	x8	2048
	C1	x16	
	A1	x8	
	B1	x16	
2 Gb	DA	x8	2048
	CA	x16	
	AA	x8	
	BA	x16	
	83	x8	
	93	x16	
4 Gb	DC	x8	2048
	CC	x16	
	AC	x8	2048/4096
	BC	x16	
	84	x8	2048
	94	x16	
8 Gb	D3	x8	2048/4096
	C3	x16	
	A3	x8	
	B3	x16	
	85	x8	2048
	95	x16	

Table 5-19. Supported NAND Devices (continued)

Capacity	Device ID	Bus Width	Page Size
16 Gb	D5	x8	2048/4096
	C5	x16	
	A5	x8	
	B5	x16	
	86	x8	2048
	96	x16	
32 Gb	D7	x8	2048/4096
	C7	x16	
	A7	x8	
	B7	x16	
	87	x8	2048
	97	x16	
64 Gb	DE	x8	2048/4096
	CE	x16	
	AE	x8	
	BE	x16	

When the parameters are retrieved from the ROM table: page size and block size are updated based on 4th byte of NAND ID data. Due to inconsistency among manufacturers, only devices of at least 2Gb (included) have these parameters updated. Therefore, the ROM Code supports 4kB page devices but only if their size, according to the table, is at least 2Gb. Devices smaller than 2Gb have the block size parameter fixed to 128kB. Table 5-20 shows the 4th ID Data byte encoding used in ROM Code.

Table 5-20. 4th NAND ID Data Byte

Item	Description	I/O #							
		7	6	5	4	3	2	1	0
Page Size	1kB							0	0
	2kB							0	1
	4kB							1	0
	8kB							1	1
Cell type	2 levels					0	0		
	4 levels					0	1		
	8 levels					1	0		
	16 levels					1	1		
Block Size	64kB			0	0				
	128kB			0	1				
	256kB			1	0				
	512kB			1	1				

5.2.6.4.2.4 *Reading NAND Geometry from I2C EEPROM*

ROM supports a special boot mode called NANDI2C to support NAND devices whose geometry cannot be detected by the ROM automatically using methods described in the previous section (Figure 5-12). If this boot mode is selected, the ROM code reads NAND geometry from an I2C EEPROM. If the read is successful, ROM code moves to the next steps of NAND boot beginning with reading bad blocks information.

If the I2C EEPROM read fails, the ROM will fall back to querying the NAND for the geometry information, as described above.

Note: The NAND bus width configuration mentioned in the I2C EEPROM overrides the BUSWIDTH configuration selected by SYSBOOT pins.

Table 5-21 lists the device pins configured by the ROM for NANDI2C boot mode, in addition to the NAND boot pins described in the previous sections.

Table 5-21. Pins Used for NAND I2C Boot for I2C EEPROM Access

Signal Name	Pin Used	CTRL_CONF Register	Register Setting
I2C SCL	i2c0_scl	CTRL_CONF_I2C0_SCL	0x000e0000
I2C SDA	i2c0_sda	CTRL_CONF_I2C0_SDA	0x000e0000

ROM accesses the I2C EEPROM at I2C slave address 0x50 and reads 7 bytes starting from address offset 0x80. The format of this NAND geometry information follows:

Table 5-22. NAND Geometry Information on I2C EEPROM

<table>
<tr><th rowspan="2">Byte address</th><th colspan="2">Information</th></tr>
<tr><th>Upper Nibble</th><th>Lower Nibble</th></tr>
<tr><td>0x80</td><td colspan="2">Magic Number – 0x10</td></tr>
<tr><td>0x81</td><td colspan="2">Magic Number – 0xb3</td></tr>
<tr><td>0x82</td><td colspan="2">Magic Number – 0x57</td></tr>
<tr><td>0x83</td><td colspan="2">Magic Number – 0xa6</td></tr>
<tr><td>0x84</td><td>NAND column address (word/byte offset within a page) size in bytes, Example: 2</td><td>NAND row address (page offset) size in bytes. Example: 3</td></tr>
<tr><td>0x85</td><td>Page size (2^N) exponent “N”. Example (for page size of 2048): 11</td><td>Pages per block (2^N) exponent “N”
Example (for number of blocks 64): 6</td></tr>
<tr><td>0x86</td><td>NAND bus width
0 → 8-bit, 1 → 16-bit</td><td>ECC Type
0 → No ECC, 1 → BCH8, 2 → BCH16</td></tr>
</table>

5.2.6.4.2.5 *ECC Correction*

The default ECC correction applied is BCH 8 bits per sector using the GPMC and ELM hardware.

For device ID codes D3h, C3h, D5h, C5h, D7h, C7h, DEh, CEh when manufacturer code (first ID byte) is 98h the Cell type information is checked in the 4th byte of ID data. If it is equal to 10b then the ECC correction applied is BCH 16 bits per sector.

In addition, ECC computation done by the ROM can be turned off completely by using SYSBOOT[6] . This is particularly useful when interfacing with NAND devices that have built in ECC engines.

5.2.6.4.2.6 *Bad Block Verification*

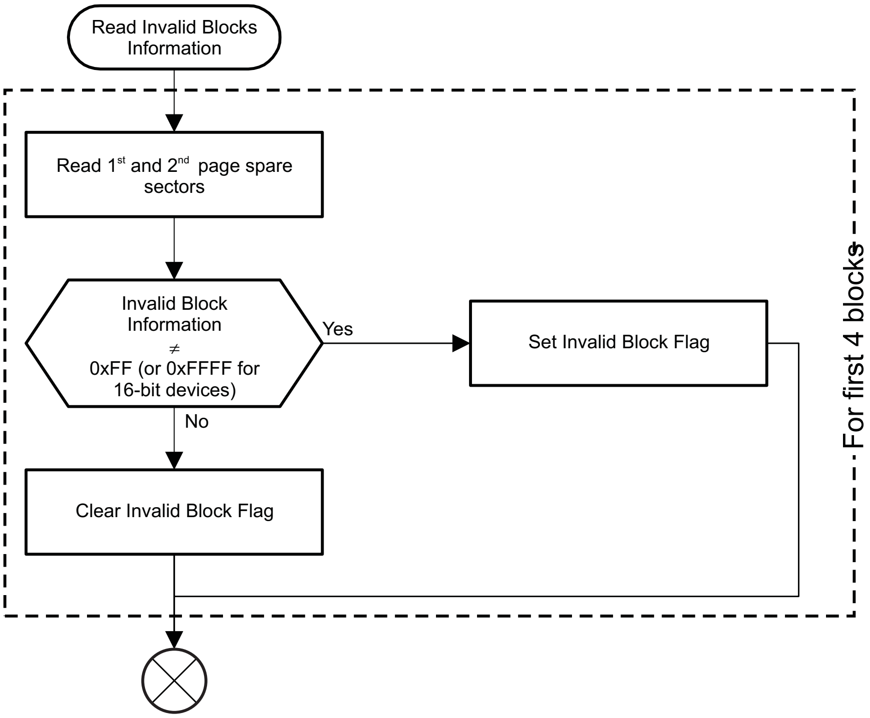
Invalid blocks are blocks with invalid bits whose reliability cannot be guaranteed by the manufacturer. Invalid bits are identified in the factory or during the programming and reported in the initial invalid block information located in the spare area on the 1st and 2nd page of each block. Since the ROM Code is looking for an image in the first four blocks, it must detect block validity status of these blocks. Blocks which are detected as invalid are not accessed later on.

The valid block status is coded in the spare areas of the first two pages of a block:

- 8-bit device: first byte equals FFh in 1st and 2nd pages
- 16-bit device: first word equals FFFFh in 1st and 2nd pages

Figure 5-13 shows the invalid block detection routine. The routine reads spare areas and checks validity data pattern.

Figure 5-13. NAND Invalid Blocks Detection



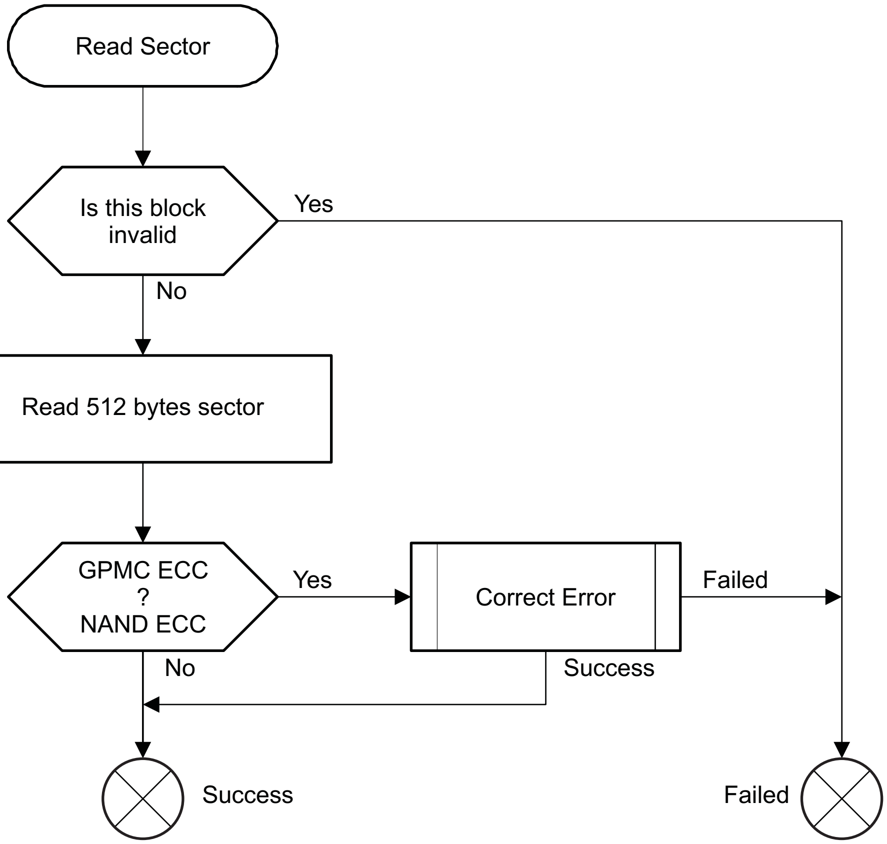
5.2.6.4.3 NAND Read Sector Procedure

The ROM Code reads data from NAND devices in 512-byte sectors. The read function fails in two cases:

- The accessed sector is within a block marked as invalid
- The accessed sector contains an error which cannot be corrected with ECC

Figure 5-14 shows the read sector routine for NAND devices. The ROM Code uses a normal read (command 00h 30h) to read NAND page data.

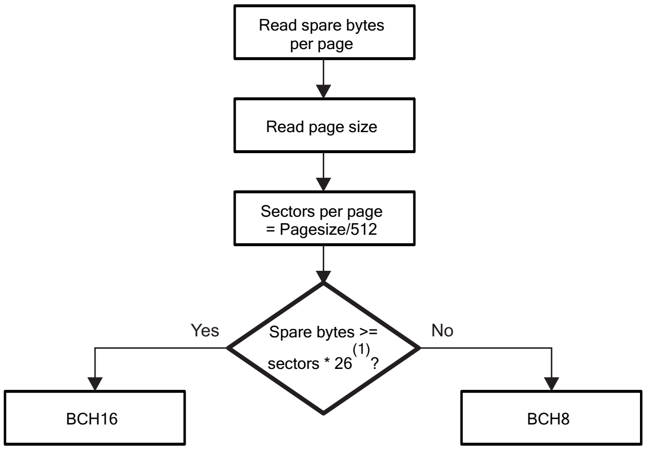
Figure 5-14. NAND Read Sector Procedure



Page data can contain errors due to memory alteration. The ROM Code uses an ECC correction algorithm to detect and possibly correct those errors. The ECC algorithm used is BCH with capability for correcting 8b or 16b errors per sector. Selecting between BCH8 and BCH16 ECC scheme is shown in Figure 5-15. The BCH data is automatically calculated by the GPMC on reading each 512-byte sector. The computed ECC is compared against ECC stored in the spare area for the corresponding page. Depending on the page size, the amount of ECC data bytes stored in the corresponding spare area is different.

Figure 5-16 and Figure 5-17 show the mapping of ECC data inside the spare area for respectively 2KB-page and 4KB-page devices. If both ECC data are equal then the Read Sector function returns the read 512-byte sector without error. Otherwise the ROM Code tries to correct errors in the corresponding sector (this procedure is assisted by the ELM hardware) and returns the data if successful. If errors are uncorrectable, the function returns with FAIL.

Figure 5-15. NAND ECC Scheme Selection Procedure



(1) 26 is the number of ECC bytes needed for the 512-byte sector in the BCH16 scheme.

Figure 5-16. ECC Data Mapping for 2KB Page and 8b BCH Encoding

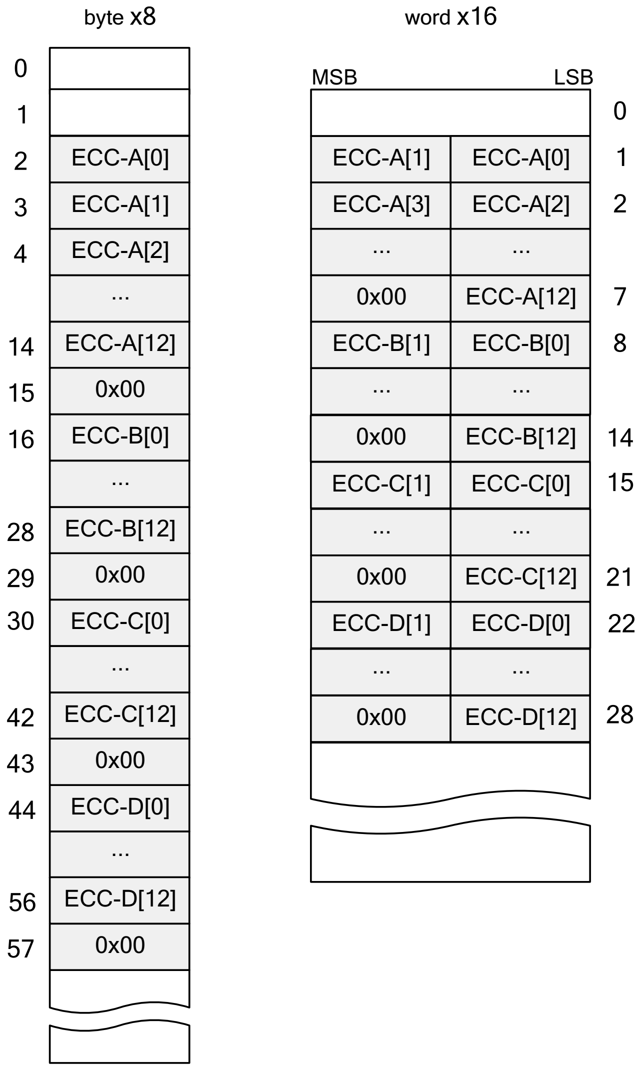
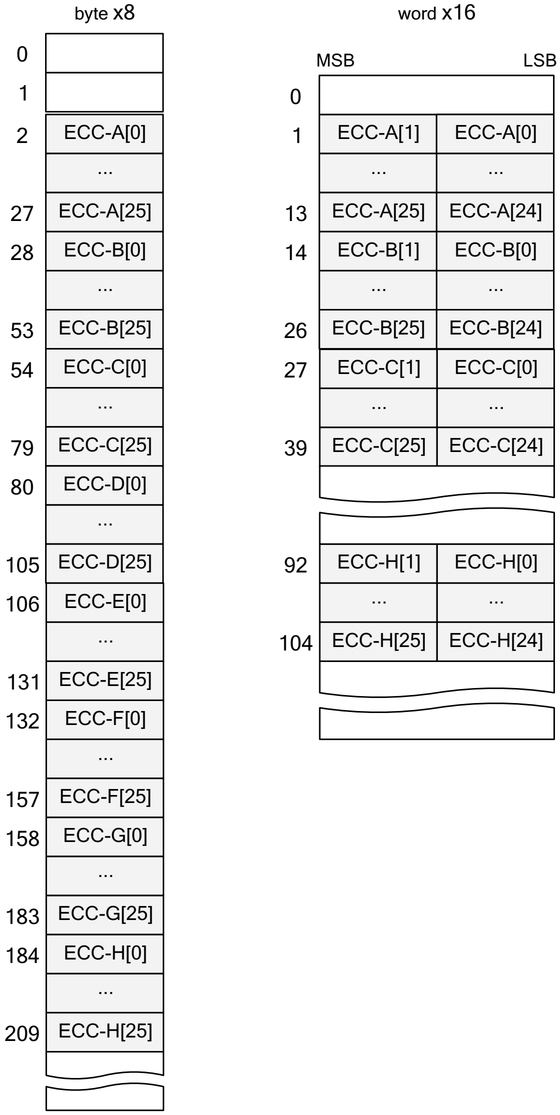


Figure 5-17. ECC Data Mapping for 4KB Page and 16b BCH Encoding



5.2.6.4.4 Pins Used

Table 5-23 lists the device pins configured by the ROM for NAND boot mode. Not all the pins are driven at boot time.

Table 5-23. Pins Used for NAND Boot

Signal Name	Pin Used	CTRL_CONF Register	Register Setting
cs0	gpmc_cs0	CTRL_CONF_GPMC_CS0	0x00010000
advn_ale	gpmc_advn_ale	CTRL_CONF_GPMC_ADVN_ALE	0x00010000
oen_ren	gpmc_oen_ren	CTRL_CONF_GPMC_OEN_REN	0x00010000
be0n_cle	gpmc_be0n_cle	CTRL_CONF_GPMC_BE0N_CLE	0x00010000
Wen	gpmc_wen	CTRL_CONF_GPMC_WEN	0x00010000
Clk	gpmc_clk	CTRL_CONF_GPMC_CLK	0x00060000
ad0 - ad15	gpmc_ad0 - gpmc_ad15	CTRL_CONF_GPMC_AD0 to CTRL_CONF_GPMC_AD15	0x00060000

Table 5-24. Pins Used for NAND Boot Wait Pin Selection

Signal name	Pin Used Pinmux Option 0	CTRL_CONF Register	Register Setting	Pin Used Pinmux Option 1	CTRL_CONF Register	Register Setting
wait	gpmc_wait0	CTRL_CONF_ GPMC_WAIT0	0x00060000	gpmc_csn3	CTRL_CONF_ GPMC_CSN3	0x00600001

5.2.6.4.4.1 SYSBOOT Signals

Table 5-25 lists the SYSBOOT signals for NAND boot.

Table 5-25. SYSBOOT Signals for NAND Boot

SYSBOOT[6]	Used by Boot ROM to determine if NAND ECC is handled by ROM or NAND device. Please note that when ROM is booting from external NAND, WAIT monitoring will be forced by ROM code. SYSBOOT setting is not required to enable/disable Wait monitoring. 0 – ECC done by ROM 1 – ECC done by NAND Device
SYSBOOT[8]	Decides which pin READY/BUSY needs to be connected to when NAND is selected. 0 – Wait mux option 0 1 – Wait mux option 1

5.2.6.5 MMC/SD Cards

5.2.6.5.1 Overview

The ROM Code supports booting from MMC/SD cards in the following conditions:

- MMC/SD Cards compliant to [6], [9] of low and high capacities.
- MMC/SD cards connected to MMC interface #0 and #1.
- Support for 3.3 V or 1.8 V I/O voltages.
- Initial 1-bit MMC Mode and optional 4-bit mode, if MMC/SD card supports it.
- Clock Frequency: identification mode: 240 KHz; data transfer mode up to 12 MHz.
- Only one card connected to the bus.
- Raw mode, image data read directly from sectors in the user area.
- File system mode (FAT16/32 supported with or without Master Boot Record), image data is read from a booting file.

5.2.6.5.2 System Interconnection

An MMC/SD card or eMMC/eSD/managed NAND memory device can connect to MMC0 or MMC1 interface.

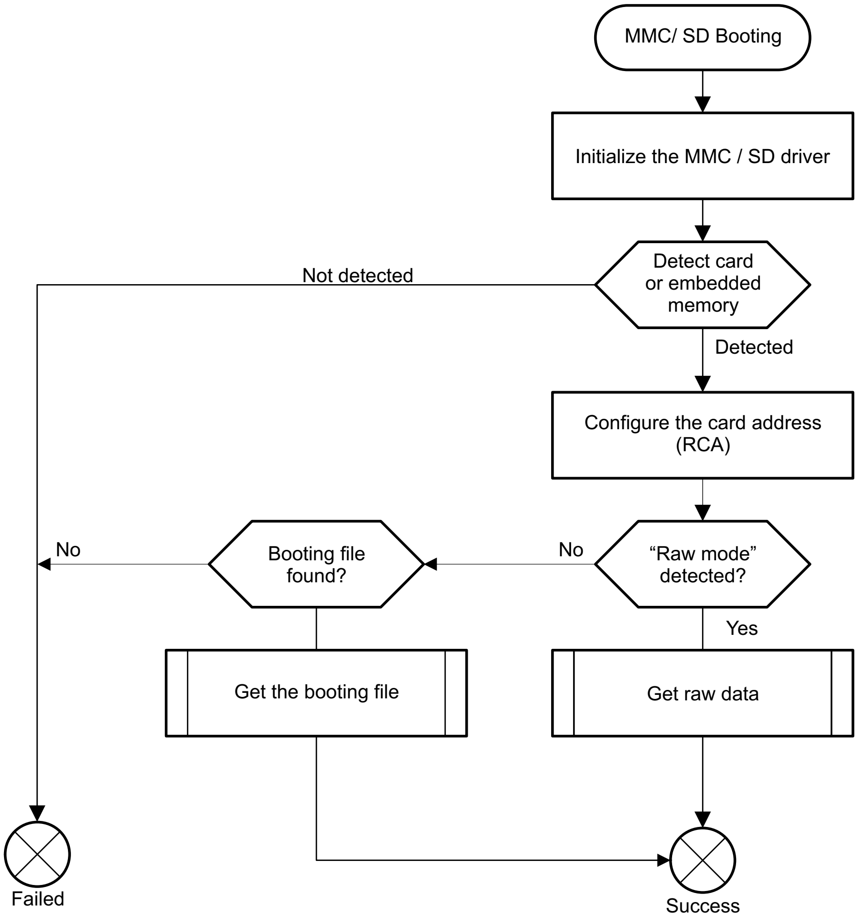
Note:

- The ROM Code does not handle the card detection feature on card cage.
- MMC0 and MMC1 support sector mode without querying the card.

5.2.6.5.3 Booting Procedure

Figure 5-18 shows the high level flowchart of the eMMC, eSD, and MMC/SD booting procedure.

Figure 5-18. MMC/SD Booting



5.2.6.5.4 *Initialization and Detection*

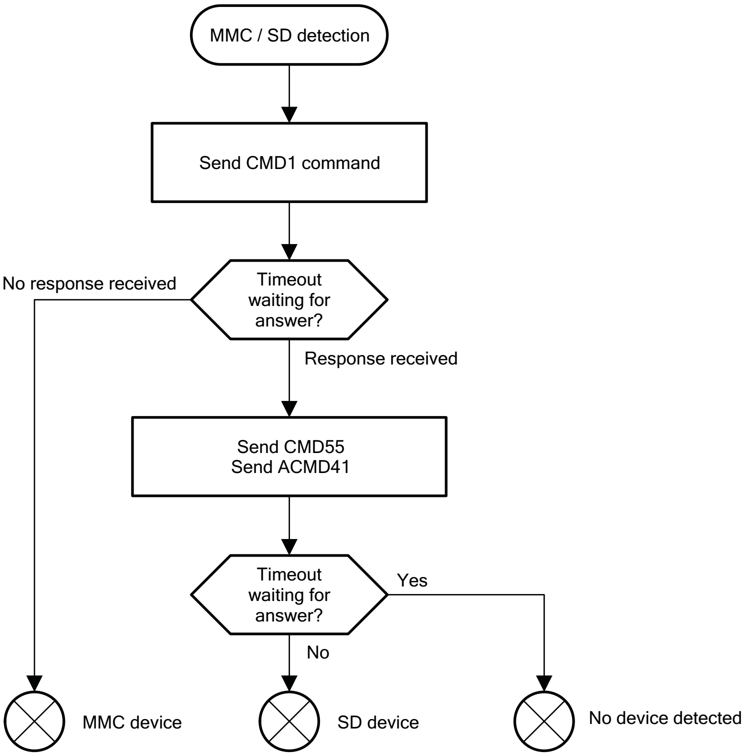
The ROM Code attempts to initialize the memory device or card connected on MMC interface. If neither memory device nor card is detected then the ROM Code carries on to the next booting device. The standard identification process and Relative Card Address (RCA) assignment are used. However, the ROM Code assumes that only one memory or card is present on the bus. This first sequence uses the CMD signal which is common to SD and MMC devices.

MMC and SD standards detail this phase as initialization phase. Both standards differ in the first commands involved: CMD1 and ACMD41. The ROM Code uses this difference in command set to discriminate between MMC and SD devices: CMD1 is supported only by the MMC standard, whereas ACMD41 is only supported by SD standard.

The ROM Code first sends a CMD1 to the device and gets a response only if an MMC device is connected. If no response is received, then ACMD41 (ACMD41 is made out of CMD55 and ACMD41) is sent and a response is expected from an SD device. If no response is received, then it is assumed that no device is connected and the ROM Code exits the MMC/SD Booting procedure with FAIL. Figure 5-19 shows the detection procedure.

At first the ROM queries the card with CMD1, ARG = 0, to get the OCR from the card. Bit 30 of the response received from the card is set to 1 by the ROM, and this modified value is used as the argument for subsequent CMD1. This is done to indicate to the card that the ROM supports sector addressing. This mode might not be compatible with older (older than v4.4) versions of cards.

Figure 5-19. MMC/SD Detection Procedure



The contents of an MMC/SD card or an eMMC/eSD device can be formatted as raw binary or within a FAT file system. The ROM Code reads out raw sectors from image or the booting file within the file system and boots from it.

5.2.6.5.5 *MMC/SD Read Sector Procedure in Raw Mode*

In raw mode the booting image can be located at one of the four consecutive locations in the main area:

- offset 0x0
- 0x40000 (256KB)
- 0x80000 (512KB)
- 0xC0000 (768KB)

A booting image must not exceed 256KB in size. However, it is possible to flash a device with an image greater than 256KB starting at one of the four locations and the ROM Code will not check the image size. The only drawback is that the image will cross the subsequent image boundary.

The raw mode is detected by reading sectors #0 and #1024. The content of these sectors is then verified for presence of a TOC structure as described in 10. In the case of a GP Device, a Configuration Header (CH) must be located in the first sector followed by a GP header (1). The CH might be void (only containing a CHSETTINGS item for which the valid field is zero).

(1) See Section 5.2.9, *Image Format*.

5.2.6.5.6 MMC/SD Read Sector Procedure in FAT Mode

MMC/SD cards or eMMC/eSD devices hold a FAT file system which ROM Code reads and processes. The image used by the booting procedure is taken from a specific booting file named “MLO”. This file must be in the root directory on an active primary partition of type FAT16 or FAT32. Please refer to [8] and [10] for a more detailed description of MMC/SD file system support.

An MMC/SD card or eMMC/eSD device can be configured either as floppy-like or hard-drive-like.

- When acting as floppy-like, the content is a single file system without any Master Boot Record (MBR) holding a partition table.
- When acting as hard-drive-like, an MBR is present in the first sector. This MBR holds a table of partitions, one of which must be FAT16/32, primary and active.

According to [8], the card or eMMC/eSD device should always hold an MBR except for ones using floppy-like file system (please refer to the CSD internal Register fields FILE_FORMAT_GRP and FILE_FORMAT in [6]). However, depending on the used operating system the MMC/SD card or device will be formatted either with partitions (using an MBR) or without. The ROM Code supports both types, described in the following section.

The ROM Code retrieves a map of the booting file from the FAT table. The booting file map is a collection of all FAT table entries related to the booting file (a FAT entry points to a cluster holding part of the file). The booting procedure uses this map to access any 512 byte sector within the booting file without involving ROM Code FAT module.

The sector read procedure utilizes standard MMC/SD raw data read function. The sector address is generated based on the booting memory file map collected during the initialization. Hence the ROM Code can address sectors freely within the booting file space.

5.2.6.5.7 FAT File System

The following sections describe functions used by the ROM Code but do not fully describe the Master Boot Record and the FAT file system:

- Recognize if a sector is the 1st sector of an MBR
- Recognize if a sector is the 1st sector of a FAT16/32
- Find the 1st cluster of the booting file
- Buffer the booting file FAT entries.

If true, an active FAT 16/32 partition is searched in all 4 MBR partition entries, based on the type field. If the MBR entries are not valid, or if no useable partition is found, then the ROM Code returns to the booting procedure with FAIL. The extended partitions are not checked. The booting file must reside in a primary partition.

If a partition is found, then its first sector is read and used further on. If no MBR is present (if a floppy-like system), the first sector of the device is read and used further on.

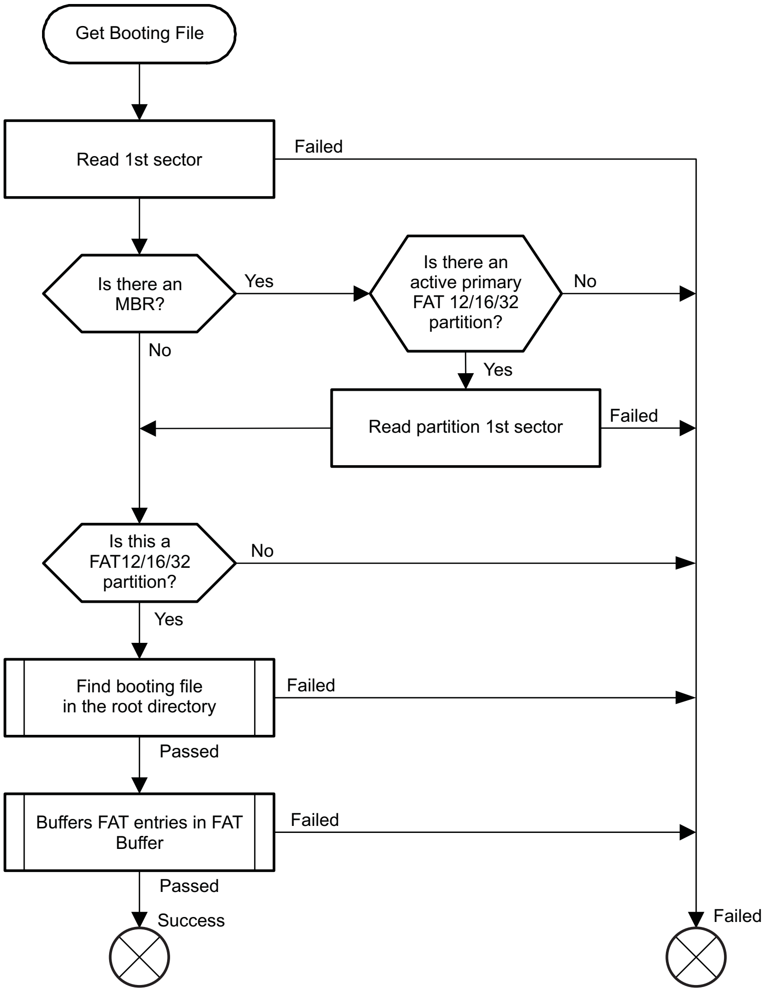
The read sector is checked to be a valid FAT16 or FAT32 partition. If this fails, and another partition type is used (that is, Linux FS or any other), or if the partition is not valid, the ROM Code returns with FAIL.

Otherwise, the root directory entries are searched for a file named depending on the booting device. The Long File Names (LFN) format is not used and only file names in 8.3 format are searched for. If no valid file is found, the ROM Code returns with FAIL.

Once the file is found, the ROM Code reads the File Allocation Table (FAT) and buffers the single-linked chain of clusters in a FAT Buffer which the booting procedure uses to access the file directly sector by sector. For FAT16 and FAT32 (valid if a specific flag has been set in the FAT32 Boot Sector), multiples copies of the FAT exist (ROM Code supports only two copies). When buffering FAT entries, the two FATs are compared. If they do not match, only entries from the last FAT are used. The FAT Buffer holds sector numbers and not cluster numbers. The ROM Code converts each cluster entry to one or several sector entries, if applicable.

Figure 5-20 shows the whole process.

Figure 5-20. SD/MMC Booting, Get Booting File



5.2.6.5.7.1 *Master Boot Record (MBR)*

The Master Boot Record is the 1st sector of a memory device. It is made out of some executable code and 4 partition entries. The aim of such a structure is to divide the hard disk in partitions mostly used to boot different systems (Microsoft Windows®, Linux, or others). Table 5-26 and Table 5-27 describe the structure. Table 5-28 describes the valid partition types searched by the ROM Code.

Table 5-26. Master Boot Record Structure

Offset	Length (bytes)	Entry Description	Value
0000h	446	Optional Code	
01BEh	16	Partition Table Entry	(see Table 5-27)
01CEh	16	Partition Table Entry	(see Table 5-27)
01DEh	16	Partition Table Entry	(see Table 5-27)
01EEh	16	Partition Table Entry	(see Table 5-27)
01FEh	2	Signature	AA55h

Table 5-27. Partition Entry

Offset	Length (bytes)	Entry Description	Value
0000h	1	Partition State	00h: Inactive 80h: Active
0001h	1	Partition Start Head	Hs
0002h	2	Partition Start Cylinder and Sector	Cs[7:0]-Cs[9:8]-Ss[5:0]
0004h	1	Partition Type	See Table 5-28 for partial partition types
0005h	1	Partition End Head	He
0006h	2	Partition End Cylinder and Sector	Ce[7:0]-Ce[9:8]-Se[5:0]
0008h	4	First sector position relative to the beginning of media	LBAs=Cs.H.S+ Hs.S+ Ss-1
000Ch	4	Number of sectors in partition	LBAe=Ce.H.S+ He.S+ Se-1 Nbs= LBAe-LBAs+1

Table 5-28. Partition Types

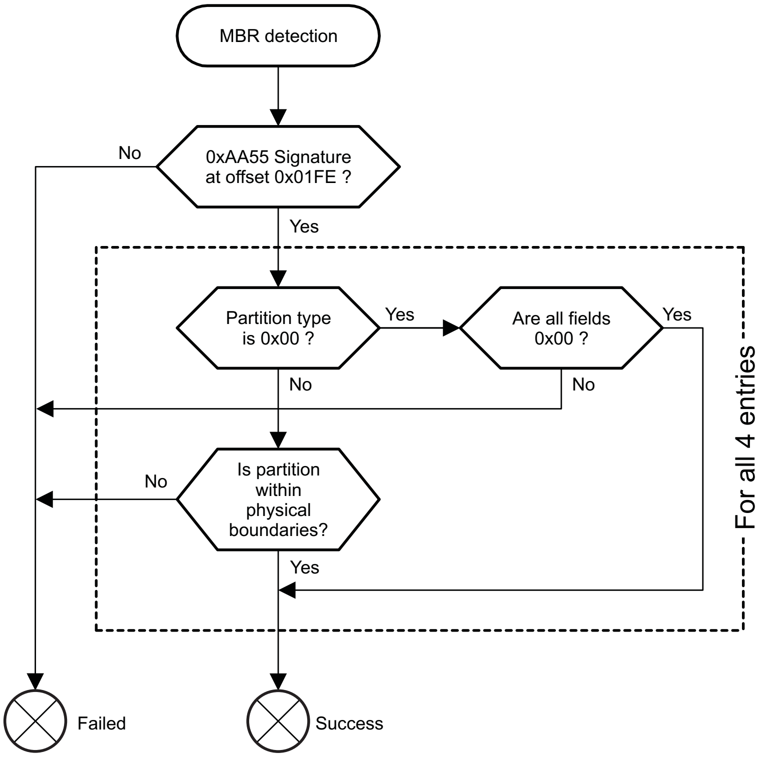
Partition Type	Description
04h, 06h, 0Eh	FAT16
0Bh, 0Ch, 0Fh	FAT32

Figure 5-21 shows whether the ROM Code detects a sector is the 1st sector of an MBR.

The ROM Code first checks if the signature is present. Each partition entry is checked:

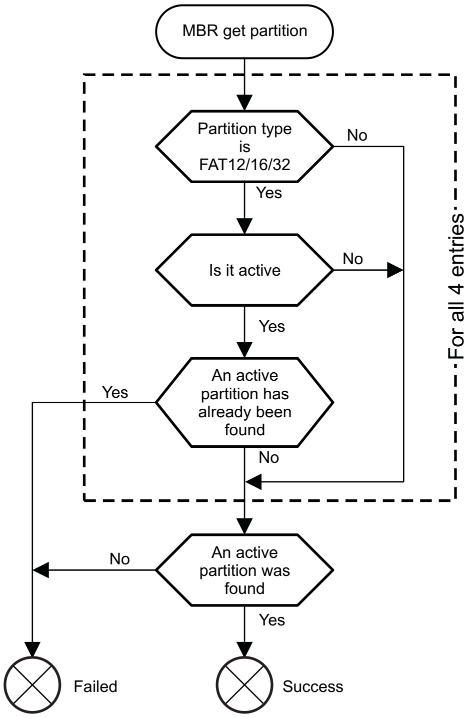
- If the type is 00h then all fields in the entry must be 00h.
- The partition is checked to be within physical boundaries, that is, the partition is located inside and its size fits the total physical sectors.

Figure 5-21. MBR Detection Procedure



Once identified, the ROM Code gets the partition using the procedure shown in Figure 5-22. The partition type is checked to be FAT16 or FAT32. Its state must be 00h or 80h. If than one active partition exists, the test fails. The ROM Code returns FAIL if no active primary FAT16/32 is found.

Figure 5-22. MBR, Get Partition



5.2.6.5.7.2 FAT16/32 Boot Sector

The FAT file system includes:

- Boot sector which holds the BIOS Parameter Block (BPB)
- File Allocation Table (FAT) which describes the use of each cluster of the partition
- Data area which holds the files, directories and root directory (for FAT16, the root directory has a specific fixed location).

Table 5-29 describes the boot sector. For more details, refer to [12]. **Note:** In the following description, all the fields whose names start with BPB_ are part of the BPB. All the fields whose names start with BS_ are part of the Boot Sector. They are not part of the BPB (not mandatory) and they are not used by the ROM Code.

Table 5-29. FAT Boot Sector

Offset		Length (bytes)	Name	Description
0000h		3	BS_jmpBoot	Jump Instruction to Boot Code (not used)
0003h		8	BS_OEMName	Name of the System which created the partition
000Bh		2	BPB_BytsPerSec	Bytes per sector (usually 512)
000Dh		1	BPB_SecPerClus	Number of sectors per allocation unit
000Eh		2	BPB_RsvdSecCnt	Number of reserved sectors for the Boot Sector. For FAT16 is 1, for FAT32, usually 32
0010h		1	BPB_NumFATs	Number of copies of FAT, usually 2
0011h		2	BPB_RootEntCnt	For FAT16, number of 32-byte entries in the Root Directory (multiple of BPB_BytsPerSec/32). For FAT32 this value is 0.
0013h		2	BPB_TotSec16	Total Count of sectors on the volume. If the size is bigger than 10000h or for FAT32, this field is 0 and BPB_TotSec32 holds the value
0015h		1	BPB_Media	Media Type, usually F8h: fixed, non-removable
0016h		2	BPB_FATSz16	For FAT16, size in sectors of one FAT. For FAT32, holds 0
0018h		2	BPB_SecPerTrk	Number of sectors per track, 63 for SD/MMC
001Ah		2	BPB_NumHeads	Number of heads, 255 for SD/MMC
001Ch		4	BPB_HiddSec	Number of sectors preceding the partition
0020h		4	BPB_TotSec32	Total Count of sectors on the volume. If the size is smaller than 10000h (for FAT16), this field is 0 and BPB_TotSec16 is valid
FAT16	0024h	1	BS_DrvNum	Drive Number
	0025h	1	BS_Reserved1	00h
	0026h	1	BS_BootSig	Extended Boot Signature 29h. Indicates that the following three fields are present
	0027h	4	BS_VolID	Volume Serial Number
	002Bh	11	BS_VolLab	Volume Label
	0036h	8	BS_FilSysType	File system Type: “FAT16”, “FAT32”. Note: This field is not mandatory (BS_) and cannot identify the partition type.

Table 5-29. FAT Boot Sector (continued)

Offset		Length (bytes)	Name	Description
FAT32	0024h	4	BPB_FATSz32	Size in sectors of one FAT. Field BPB_FATSz16 must be 0
	0028h	2	BPB_ExtFlags	FAT Flags: [7]: 0=FAT is mirrored; 1=Only one FAT is used [3:0]: Number of used FAT if no mirroring used
	002Ah	2	BPB_FSVer	File system Version Number
	002Ch	4	BPB_RootClus	First Cluster number of the Root Directory
	0030h	2	BPB_FSInfo	Sector number of FSINFO Structure in the reserved-area, usually 1
	0032h	2	BPB_BkBootSec	If non-zero, indicates the sector number in the reserved-area of a copy of the Boot Sector
	0034h	12	BPB_Reserved	Reserved, set to 00h
	0040h	1	BS_DrvNum	Drive Number
	0041h	1	BS_Reserved1	00h
	0042h	1	BS_BootSig	Extended Boot Signature 29h. Indicates that the following 3 fields are present
	0043h	4	BS_VolID	Volume Serial Number
	0047h	11	BS_VolLab	Volume Label
	0052h	8	BS_FilSysType	File system Type: “FAT16”, “FAT32”. Note: This field is not mandatory (BS_) and cannot identify the partition type.
01FEh		2	BPB_Signature	AA55h

To check whether a sector holds a valid FAT16/32 partition, only fields starting with BPB can be checked because they are mandatory. The fields starting from offset 0024h to 01FDh cannot be used for the check because they differ when using FAT16 or FAT32. Figure 5-23 describes the procedure.

1. The ROM Code checks if the BPB_Signature is equal to AA55h.
2. The ROM Code checks some fields which must have some values: BPB_BytsPerSec, BPB_SecPerClus, BPB_RsvdSecCnt, BPB_NumFATs, BPB_RootEntCnt
3. If the geometry of the device is known (valid CHS for device size < 4GB) then it is compared against BPB_SecPerTrk and BPB_NumHeads fields
4. If an MBR was found before, the partition size is also checked:

   BPB_TotSec16 = MBR_Partition_Size

   or

   BPB_TotSec32 = MBR_Partition_Size
5. The field BPB_ToSec16 is used if the total number of sectors is below 65518 (in this case BPB_TotSec32 = 0). Otherwise, BPB_TotSec32 is used (BPB_TotSec16=0).
6. The partition sector offset is also checked: BPB_HiddSec = MBR_Partition_Offset (if this value is not 0 as some operating systems do not update this field correctly).
7. The last step is to decide the type of FAT file system. The ROM Code computes the number of clusters in the Data Area part of the partition:

$$\begin{aligned} Nb_{clusters} < 4085 &\Rightarrow FAT12 \\ 4085 \le Nb_{clusters} < 65525 &\Rightarrow FAT16 \\ 65525 \le Nb_{clusters} &\Rightarrow FAT32 \end{aligned} \tag{1}$$

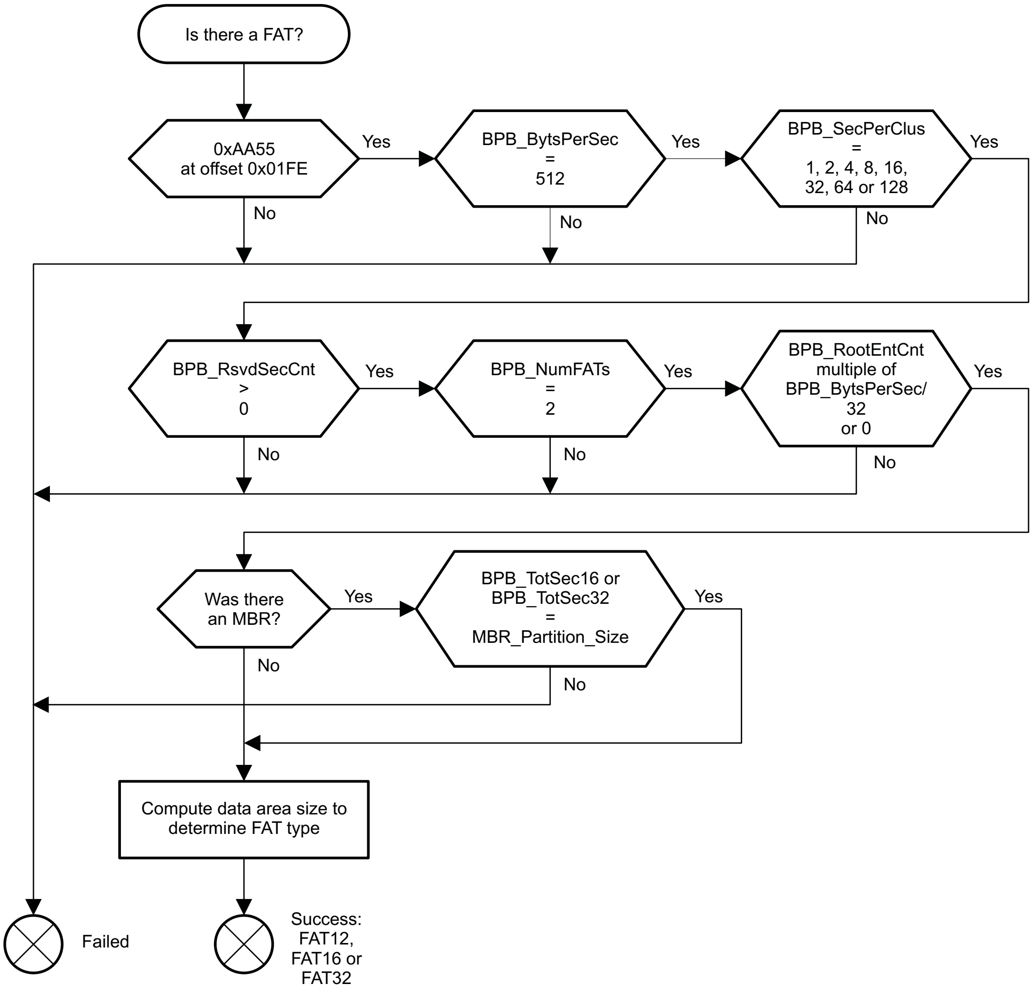
where $Nb_{clusters}$ is given by the size of the data area:

$$RootDirSectors = \frac{BPB\_RootEntCnt*32}{BPB\_BytesPerSec}$$

$$DataSec = BPB\_TotSec - (BPB\_ResvdSecCnt + (BPB\_NumFATs.BPB\_FATSz) + RootDirSectors)$$

$$Nb_{clusters} = \frac{DataSec}{BPB\_SecPerClus} \tag{2}$$

Figure 5-23. FAT Detection Procedure



5.2.6.5.7.3 *FAT16/32 Root Directory*

Next, the ROM Code finds the booting file named “MLO” inside the root directory of the FAT16/32 file system. The file is not searched in any other location.

For a FAT16 file system, the root directory has a fixed location which is cluster 0. For a FAT32 file system, its cluster location is given by BPB_RootClus. The general formulae to find the sector number (relative to device sector 0, not partition sector 0) of a cluster is given by:

$$\text{Cluster}_{\text{sector}} = \text{BPB\_HiddSec} + \text{BPB\_RsvdSecCnt} + \text{BPB\_NumFATs} \times \text{BPB\_FATSz} + \text{Cluster} \times \text{BPB\_SecPerClus} \quad (3)$$

Note: the BPB_HiddSec field can contain 0 even if the FAT file system is located somewhere other than on sector 0 (floppy-like). The ROM Code actually uses the partition offset taken from the MBR instead of this field, which can be wrong. If no MBR is found (floppy-like), the value 0 is used.

Each entry in the root directory is 32 bytes long and holds information about the file such as filename, date of creation, rights, and cluster location. Table 5-30 describes more.

The ROM Code checks each entry in the root directory until either the booting file is found or the entry is empty (first byte is 00h), or when the end of the root directory is reached. Entries with an ATTR_LONG_NAME attribute (LFN) and with first byte at E5h (erased file) are ignored. When found, the first cluster offset of the file is read from the DIR_FstClusHi and DIR_FstClusLo fields.

There is a slight difference between FAT16 and FAT32 when handling the Root Directory. On FAT16, this directory has a fixed location and length fixed by BPB_RootEntCnt which is the total number of 32-byte entries. Handling this directory is therefore straightforward. On FAT32, the root directory is like a standard file. The File Allocation Table (FAT) must retrieve each sector of the directory.

Table 5-30. FAT Directory Entry

Offset	Length (bytes)	Name	Description
0000h	11	DIR_Name	Short Name (8+3)
000Bh	1	DIR_Attr	File Attributes: ATTR_READ_ONLY 01h ATTR_HIDDEN 02h ATTR_SYSTEM 04h ATTR_VOLUME_ID 08h ATTR_DIRECTORY 10h ATTR_ARCHIVE 20h ATTR_LONG_NAME ATTR_READ_ONLY | ATTR_HIDDEN | ATTR_SYSTEM | ATTR_VOLUME_ID
000Ch	1	DIR_NTRes	Reserved, set to 00h
000Dh	1	DIR_CrtTimeTenth	Millisecond stamp at file creation
000Eh	2	DIR_CrtTime	Time file was created
0010h	2	DIR_CrtDate	Date file was created
0012h	2	DIR_LstAccDate	Last Access date
0014h	2	DIR_FstClusHi	High word of this entry’s first cluster number
0016h	2	DIR_WrtTime	Time of last write
0018h	2	DIR_WrtDate	Date of last write
001Ah	2	DIR_FstClusLo	Low word of this entry’s first cluster number
001Ch	4	DIR_FileSize	File size in bytes

5.2.6.5.7.4 FAT16/32 File Allocation Table

The ROM Code must read the FAT to retrieve sectors either for the booting file or for the root directory (if the file system is FAT32). There can be multiple copies of the FAT inside the file system (ROM Code supports only 2) located after the boot sector:

$$\text{FATn}_{\text{sector}} = \text{BPB\_HiddSec} + \text{BPB\_RsvdSecCnt} + \text{BPB\_FATSz} \times n \qquad (4)$$

Its size is given by BPB_FATSz16 or BPB_FATSz32. The ROM Code checks each copy of the FAT if identical. If the values are different, the ROM Code uses the value from the last FAT copy. With the FAT32 file system, the copy system can be disabled according to a flag located in BPB_ExtFlags[7]. If this flag is set, then FAT BPB_ExtFlags[3:0] is used and the ROM Code verifies no other copies of FAT.

The FAT is a simple array of values each referring to a cluster located in the data area. One entry of the array is 16- or 32-bit depending on the file system.

The value inside an entry defines whether the cluster is being used and if another cluster must be taken into account. This creates a single-linked chain of clusters defining the file. The meaning of an entry is described in Table 5-31.

Note: For compatibility reasons, clusters 0 and 1 are not used for files and those entries must contain FFF8h and FFFFh (for FAT16) and ?FFFFFF8h and ?FFFFFFFh (for FAT32).

Table 5-31. FAT Entry Description

FAT16	FAT32	Description
0000h	?0000000h	Free Cluster
0001h	?0000001h	Reserved Cluster
0002h-FFEFh	00000002h-?FFFFFEFh	Used Cluster; value points to next cluster
FFF0h-FFF6h	?FFFFFF0h-?FFFFFF6h	Reserved values
FFF7h	?FFFFFF7h	Bad Cluster
FFF8h-FFFFh	?FFFFFF8h-?FFFFFFFh	Last Cluster in File

Note: FAT32 uses only bits [27:0]. The upper 4 bits are usually 0 and should be left untouched. When accessing the root directory for FAT32, the ROM Code starts from the root directory cluster entry and follows the linked chain to retrieve the clusters.

When the booting file is found, the ROM Code buffers each FAT entry corresponding to the file in a sector way. This means each cluster is translated to one or several sectors depending on the number of sectors in a cluster (BPB_SecPerClus). This buffer is used later by the booting procedure to access the file.

5.2.6.5.8 Pins Used

Table 5-32 and Table 5-33 list the device pins configured by the ROM for MMC boot mode. Not all pins are driven at boot time.

Table 5-32. Pins Used for MMC0 Boot

Signal name	Pin used	CTRL_CONF Register	Register Setting
clk	mmc0_clk	CTRL_CONF_MMC0_CLK	0x08050000
cmd	mmc0_cmd	CTRL_CONF_MMC0_CMD	0x08060000
dat0	mmc0_dat0	CTRL_CONF_MMC0_DAT0	0x08060000
dat1	mmc0_dat1	CTRL_CONF_MMC0_DAT1	0x08060000
dat2	mmc0_dat2	CTRL_CONF_MMC0_DAT2	0x08060000
dat3	mmc0_dat3	CTRL_CONF_MMC0_DAT3	0x08060000

Table 5-33. Pins Used for MMC1 Boot

Signal name	Pin used	CTRL_CONF Register	Register Setting
clk	gpmc_cs1	CTRL_CONF_GPMC_CS1	0x08050002
cmd	gpmc_cs2	CTRL_CONF_GPMC_CS2	0x08060002
dat0	gpmc_ad8	CTRL_CONF_GPMC_AD8	0x08060002
dat1	gpmc_ad9	CTRL_CONF_GPMC_AD9	0x08060002
dat2	gpmc_ad10	CTRL_CONF_GPMC_AD10	0x08060002
dat3	gpmc_ad11	CTRL_CONF_GPMC_AD11	0x08060002

5.2.6.6 SPI

SPI EEPROMs or SPI flashes have an EEPROM or NOR flash backend and they connect to the device using the serial SPI protocol.

These devices operate in three stages: the command stage, the address stage, and the data transfer stage. The command is usually an 8-bit value followed by the address (depending on the size of the device) followed by the data to be read or written.

Because fewer pins are required, these devices are comparatively inexpensive, easy for board layout, and are the devices of choice when cost, complexity and form factor are critical considerations.

5.2.6.6.1 Features

- Supports 12 MHz clock (50% duty cycle)
- Supports only SPI Mode 3 (clock polarity = 1, clock phase = 1, chip select is active low)
- Supports only 24-bit addressable EEPROMs
- Supports only 4-pin SPI mode (CS, CLK, Serial Input, Serial Output)
- The boot devices must be connected to chip select 0 and must support the read command (03h)
- The boot image is copied into internal memory and then executed

5.2.6.6.2 Initialization and Detection

The ROM Code initializes the SPI controller, pin muxing, and clocks to communicate with the SPI device. The controller is initialized in Mode 3 and the clock is set to operate at 12 MHz. There is no specific device identification routine that is executed by the ROM code to identify whether a boot device is preset. If no SPI device is present, the sector read will return only 0xFFFFFFFF and the SPI boot will be treated as failed.

5.2.6.6.3 SPI Read Sector Procedure

The ROM Code reads SPI data from the boot device in 512-byte sectors. For each call to the SPI Read Sector routine, the SPI Read Command (0x03) is sent with the 24-bit start address of the data to be read.

Form the next iteration, a dummy value transmits on the master out line and the data is received on the master in line. This is a required process because SPI protocol always operates in full duplex mode. The dummy data transmitted by the ROM is the read command appended to the start address. The data from the boot device is received MSB first.

5.2.6.6.4 SPI Boot Image Requirement

Because this Cortex is a little-endian processor and SPI operates in a big-endian format, the boot image must be in a big-endian format while writing to the flash to avoid the endian conversion at boot time and improve boot performance.

5.2.6.6.5 Pins Used

Table 5-34 lists the device pins configured by the ROM for SPI boot mode. Not all pins are driven at boot time.

Table 5-34. Pins Used for SPI Boot

Signal Name	Pin Used	CTRL_CONF Register	Register Setting
Cs	spi0_cs0	CTRL_CONF_SPI0_CS0	0x08060000
Miso	spi0_d0	CTRL_CONF_SPI0_D0	0x08040000
mosi	spi0_d1	CTRL_CONF_SPI0_D1	0x08040000
Clk	spi0_sclk	CTRL_CONF_SPI0_SCLK	0x08060000

5.2.6.7 QSPI

QSPI EEPROMs or QSPI flashes have an EEPROM or NOR flash backend and they connect to the device using the serial QSPI protocol.

The device will operate in memory mapped mode.

ROM Code will execute ISW directly from QSPI Flash because it is configured in memory mapped mode

5.2.6.7.1 Features

- Supports only QSPI Mode 3 (clock polarity = 1, clock phase = 1, data delay = 0)
- Supports 12 MHz clock (50% duty cycle)
- Supports only 24-bit addressable EEPROMs
- Supports memory mapped mode
- Supports only 6-pin SPI mode (CS, CLK, D0, D1,D2, D3)
- Supports quad read mode
- The boot devices must be connected to chip select 0 and must support both the READ (03h) and QOR (6Bh) commands
- Image must be flashed in little endian format
- The boot device must default to quad mode, if quad-read support is desired

5.2.6.7.2 Initialization and Detection

The ROM Code initializes the QSPI controller, pin muxing, and clocks to communicate with the QSPI device. The controller is initialized in Mode 3 and the clock is set to operate at 12 MHz. There is no specific device identification routine executed by the ROM code to identify whether a boot device is preset. If no QSPI device is present, the QSPI read will return only 0xFFFFFFFF and the QSPI boot will be treated as failed.

The ROM code uses the Quad Read command (0x6B) to read from the flash, so the ROM expects the QSPI flash to be in quad mode right after reset. This configuration is typically be done at flashing time by writing to the non-volatile Quad Enable (QE) bit of the flash.

If the ROM detects SYSBOOT[7] as 1, it will read from the QSPI flash using the Single Line Read command (0x03).

5.2.6.7.3 Pins Used

Table 5-35 lists the device pins configured by the ROM for the QSPI boot mode. Not all pins are driven at boot time.

Table 5-35. Pins Used for QSPI Boot

Signal Name	Pin Used Pinmux Option 0	CTRL_CONF Register	Register Setting	Pin Used Pinmux Option 1	CTRL_CONF Register	Register Setting
Clk	gpmc_csn3	CTRL_CONF_CSN3	0x08460002	cam0_data2	CTRL_CONF_CAM0_DATA2	0x08040003
Csn0	gpmc_csn0	CTRL_CONF_CSN0	0x08060003	cam0_data3	CTRL_CONF_CAM0_DATA3	0x08060003
D0	gpmc_advn_ale	CTRL_CONF_ADVN_ALE	0x08060003	cam0_data4	CTRL_CONF_CAM0_DATA4	0x08060003
D1	gpmc_oen_ren	CTRL_CONF_OEN_REN	0x08060003	cam0_data5	CTRL_CONF_CAM0_DATA5	0x08060003
D2	gpmc_wen	CTRL_CONF_GPMC_WEN	0x08060003	cam0_data6	CTRL_CONF_CAM0_DATA6	0x08060003
D3	gpmc_be0n_cle	CTRL_CONF_GPMC_BE0N_CLE	0x08060003	cam0_data7	CTRL_CONF_CAM0_DATA7	0x08060003

5.2.6.7.4 SYSBOOT Signals

Table 5-36 lists the SYSBOOT signals for QSPI boot.

Table 5-36. SYSBOOT Signals for QSPI Boot

SYSBOOT[7]	Used to select QSPI bus width. 0b – 4 bits (D3-D0) used. ROM uses Quad Read command (0x6B). 1b – 1 bit (D0) used. ROM uses Single Read command (0x03).
SYSBOOT[6]	Used for Pinmux option selection in QSPI. 0b – pinmux option 0 is selected for QSPI. 1b – pinmux option 1 is selected for QSPI.

5.2.6.8 USB Mass Storage (USB_MS) Device

5.2.6.8.1 Device Initialization

The ROM Code supports booting from the USB interface in host mode under the following conditions:

- Using the USB1 interface.
- USB operates in High-Speed, Host mode.
- Supports Mass Storage Class device with SCSI command set.
- ROM does not support hubs.
- Data transfer is performed using the USB core's DMA, so the boot image can be loaded only from 0x40300000 because the internal memory of the ARM is not accessible to the DMA. This restricts the maximum image size to 220KB.
- ROM code used the value of DATA POLARITY inversion feature for USB based on SYSBOOT[16] (silicon revision PG1.2 only)

5.2.6.8.2 Overview

If USB boot is chosen by the SYSBOOT pin configuration:

- The USB hardware and PRCM clocks are configured.
- The ROM code implements the SCSI command set.

5.2.6.8.3 Enumeration and Detection

1. Enables the USB module by programming the PRCM to turn on the functional clocks.
2. Powers on the USB integrated Transceiver.
3. The ROM code detects that the USB is a USB A-device through the USB-ID pin pulled low.
4. The ROM code sets host mode.
5. When the ROM detects a device is connected, it performs a USB Reset to the device.
6. The ROM performs USB enumeration.

5.2.6.8.4 Device Detection Parameters

1. If the interface descriptor of the device has bInterfaceClass specified as 08h for Mass Storage Class, the ROM code checks the bInterfaceProtocol field of the USB interface descriptor for Bulk Only Transport mode, having two Bulk Endpoints(BULK –IN and BULK OUT)
2. If the bInterfaceSubclass is set to 06, SCSI transparent command set is supported by the device

   Other types of subclasses are not supported.
3. The BootROM communicates with the device per the Mass storage Protocol using the SCSI command set to read the sectors from the USB device.
4. The BootROM sends a standard SCSI enquiry command to the device to know the capabilities of the

device.

5. The BootROM parses the boot table and partition information to retrieve the boot image.

5.2.6.8.5 Boot Procedure

The USB Mass Storage device connected for booting should hold FAT file system which ROM code reads and processes. The image the booting procedure uses is taken from a specific booting file named “MLO”. This file must be located in the root directory on an active primary partition of type FAT16 or FAT32.

An MBR must be present in the first sector of the card. This MBR holds a table of partitions, one of which must be FAT16/32, primary and active.

The ROM Code retrieves a map of the booting file from the FAT table. The booting file map is a collection of all FAT entries related to the booting file (a FAT entry points to a cluster holding part of the file). The booting procedure uses this map to access any 512-byte sector within the booting file without involving the ROM Code FAT module.

The sector read procedure uses a standard SCSI READ10 function.

The sector address generates based on the booting memory file map collected during the initialization. Hence the ROM Code can address sectors freely within the booting file space.

5.2.6.8.6 Pins Used

Table 5-37 lists the device pins configured by the ROM for the USB boot mode. Not all pins are driven at boot time.

Table 5-37. Pins Used for USB_MS Boot

Signal Name	Pin Used	Register Setting
dm	usb1_dm	Not written by ROM
dp	usb1_dp	Not written by ROM
id	usb1_id	Not written by ROM
vbus	usb1_vbus	Not written by ROM

5.2.6.8.7 SYSBOOT Signals

Table 5-38 lists the SYSBOOT signals for USB mass storage boot.

Table 5-38. SYSBOOT Signals for USB_MS Boot

SYSBOOT[16][1]	Used to select swapping of USB signals. 0b – USB DP/DM is not swapped. 1b – USB DP/DM is swapped.

[1] The functionality provided by SYSBOOT[16] is only available on silicon revision PG1.2

5.2.6.9 Blocks and Sectors Search Summary for the Redundant Images

Table 5-39 summarizes numbers of blocks and sectors which are searched during the memory booting from devices requiring image shadowing. NAND is organized with blocks, which are erasable units.

Table 5-39. Blocks and Sectors Searched on non-XIP Memories

Memory	Maximum Number of Blocks Checked	Number of Sectors Searched
NAND	first 4	1[1]
SPI, eMMC/eSD and MMC/SD cards (raw mode)	first 4	1

[1] Depends on NAND geometry

For MMC/SD card booting in FAT mode, the file system area is searched for one file.

5.2.7 Peripheral Booting

5.2.7.1 Overview

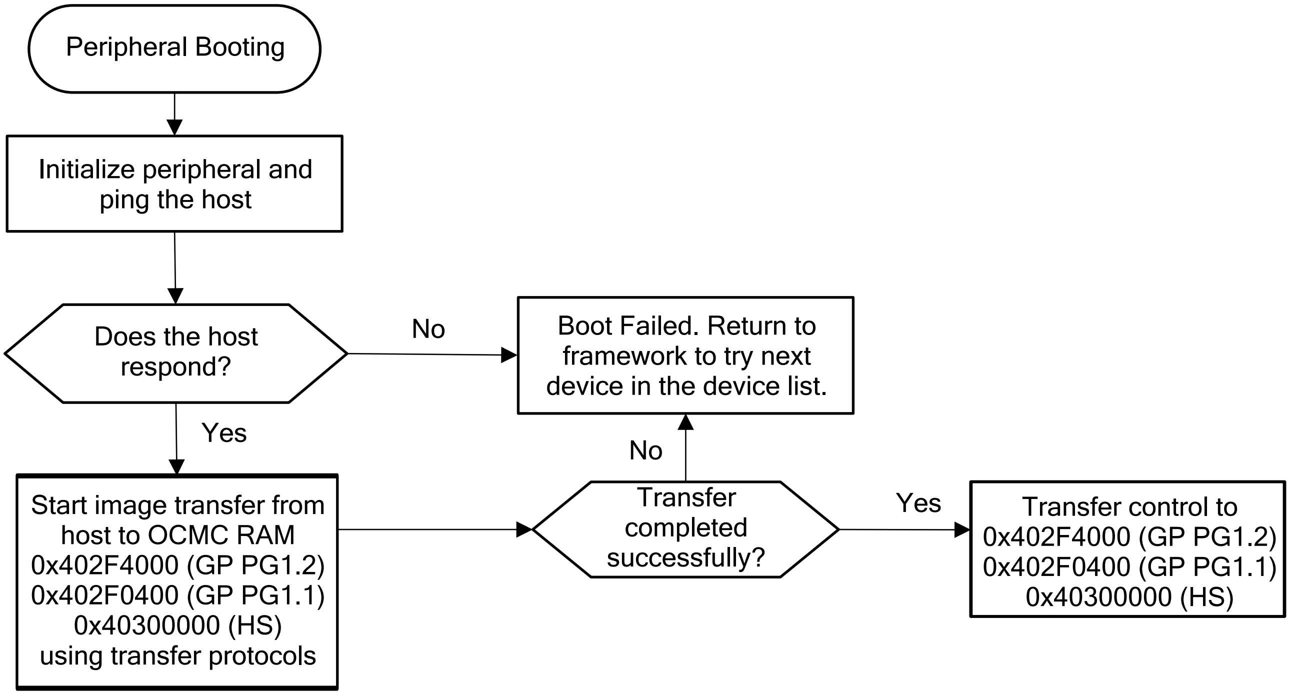
The ROM Code boots from three different peripheral interfaces:

- EMAC: 100/10 Mbps Ethernet, using standard TCP/IP network boot protocols BOOTP and TFTP
- USB: Full-speed, client mode
- UART: 115.2Kbps, 8-bit, no parity, 1 stop bit, no flow control

The purpose of booting from a peripheral interface is to download a boot image from an external host (typically a PC). This booting method is mostly used for programming flash memories connected to the device (for example, in the case of initial flashing, or for firmware updates or servicing).

5.2.7.2 Peripheral Boot Procedure Overview

Figure 5-24. Peripheral Booting Procedure



5.2.7.3 EMAC Boot Procedure

5.2.7.3.1 Device Initialization

- EMAC boot uses the CPGMAC port 1 of the device.
- Supports connection to external Ethernet PHY using the MII, RMII, RGMII and MDIO pins.
- In dual-port configurations, the ROM code assumes the Ethernet PHY with the lowest MDIO address (0–31) is connected to CPGMAC port 1.
- Device uses EFUSE register MAC_ID0 for Ethernet MAC address of the device.
- EMAC boot assumes that PHY will do auto-negotiation at power on.
- On a warm reset, if Ethernet reset isolation is enabled, the ROM will skip configuring the EMAC and directly issue the BOOTP request.
- ROM code expects an external 50-MHz reference clock requirement when using RMII PHY Interface.
- When booting using RGMII PHY interface, ROM only supports internal delay mode (delay on the Tx path with respect to clock) enabled in the device.
- Device detects if the PHY is alive on the MDIO interface and
  - Reads the STATUS register to check if Ethernet link is active
  - Waits 5 seconds for auto-negotiation to complete before timing out.
  - Reads the STATUS register to check if the PHY supports Gigabit Mode.
  - Reads the PHY registers No. 4,5,9,10 to detect the auto-negotiated mode of operation.
    - Is the mode full-duplex or half duplex
    - Speed of operation: 100/10 Mbps

5.2.7.3.2 BOOTP (RFC 951)

The device obtains the IP and boot information using the BOOTP protocol. The device prepares and broadcasts the BOOTP message with the following information:

- Device MAC address in "chaddr" field to uniquely identify the device to the server.
- "vendor-class-identifier" option number 60 (RFC 1497, RFC 1533). Servers use this information to identify the device type. The value present is "AM43xx ROM v1.0"
- "Client-identifier" option number 61 (RFC 1497, RFC 1533). This has the ASIC-ID structure (see Section 5.2.7.6), which contains additional information for the device.

The device then expects a BOOTP response that provides the following information for the booting to proceed:

- Device IP address from "yiaddr" field
- Subnetmask from extended option 1 (RFC 1497, RFC 1533)
- Gateway IP from extended option number 3 (RFC 1497, RFC 1533) or from “giaddr” field of BOOTP response.
- Boot image filename from “file” field
- TFTP server IP address from the “siaddr” field

  Timeouts and retries
- Exponentially increasing timeouts starting from 4 seconds, doubling for each retry.
- 3 retries

5.2.7.3.3 TFTP (RFC 1350)

After a successful BOOTP completion, the device initiates the TFTP download of the boot image into SRAM. The device can reach the TFTP server within the local subnet or outside through the gateway.

Timeouts and retries:

- Timeout of 1 second to receive a response for the READ request
- 5 retries for the READ request
- Retries are managed by a server once the data transfer starts (server re-sends a data packet if the ACK was not received within a timeout value)
- Device has a 60 second timeout to complete the data transfer and handle the scenario if the server dies in the middle of a data transfer

5.2.7.3.4 Pins Used

Table 5-40, Table 5-41, and Table 5-42 list the device pins configured by the ROM for the EMAC boot mode. Not all pins are driven at boot time.

Table 5-40. Pins Used for EMAC Boot in MII Mode

Signal Name	Pin Used in Device	Pin Mux Mode	CTRL_CONF Register	Register Setting
gmii1_col	MII1_COL	0	CTRL_CONF_MII1_COL	0x000c0000
gmii1_crs	MII1_CRS	0	CTRL_CONF_MII1_CRS	0x000c0000
gmii1_rxer	MII1_RX_ER	0	CTRL_CONF_MII1_RXERR	0x000c0000
gmii1_txen	MII1_TX_EN	0	CTRL_CONF_MII1_TXEN	0x00040000
gmii1_rxdv	MII1_RX_DV	0	CTRL_CONF_MII1_RXDV	0x00040000
gmii1_txd[3:0]	MII1_TXD[3:0]	0	CTRL_CONF_MII1_TXD3 to CTRL_CONF_MII1_TXD0	0x000c0000
gmii1_txclk	MII1_TX_CLK	0	CTRL_CONF_MII1_TXCLK	0x000c0000
gmii1_rxclk	MII1_RX_CLK	0	CTRL_CONF_MII1_RXCLK	0x00000000
gmii1_rxd[3:0]	MII1_RXD[3:0]	0	CTRL_CONF_MII1_RXD3 to CTRL_CONF_MII1_RXD0	0x00000000
mdio_data	MDIO	0	CTRL_CONF_MDIO_DATA	0x000a0000
mdio_clk	MDC	0	CTRL_CONF_MDC_CLK	0x000e0000

Table 5-41. Pins Used for EMAC Boot in RGMII Mode

Signal Name	Pin Used in Device	Pin Mux Mode	CTRL_CONF Register	Register Setting
rgmii1_tctl	MII1_TX_EN	2	CTRL_CONF_MII1_TXEN	0x000c0002
rgmii1_rctl	MII1_RX_DV	2	CTRL_CONF_MII1_RXDV	0x000c0002
rgmii1_td[3:0]	MII1_TXD[3:0]	2	CTRL_CONF_MII1_TXD3 to CTRL_CONF_MII1_TXD0	0x00000002
rgmii1_tclk	MII1_TX_CLK	2	CTRL_CONF_MII1_TXCLK	0x00000002
rgmii1_rclk	MII1_RX_CLK	2	CTRL_CONF_MII1_RXCLK	0x000c0002
rgmii1_rd[3:0]	MII1_RXD[3:0]	2	CTRL_CONF_MII1_RXD3 to CTRL_CONF_MII1_RXD0	0x00000002
mdio_data	MDIO	0	CTRL_CONF_MDIO_DATA	0x000a0000
mdio_clk	MDC	0	CTRL_CONF_MDIO_CLK	0x000e0000

Table 5-42. Pins Used for EMAC Boot in RMII Mode

Signal Name	Pin Used in Device	Pin Mux Mode	CTRL_CONF Register	Register Setting
rmii1_crs_dv	MII1_CRS	1	CTRL_CONF_MII1_CRS	0x00040000
rmii1_rxer	MII1_RX_ER	1	CTRL_CONF_MII1_RXERR	0x000c0001
rmii1_txen	MII1_TX_EN	1	CTRL_CONF_MII1_TXEN	0x000c0001
rmii1_txd[1:0]	MII1_TXD[1:0]	1	CTRL_CONF_MII1_TXD1 to CTRL_CONF_MII1_TXD0	0x000c0001
rmii1_rxd[1:0]	MII1_RXD[1:0]	1	CTRL_CONF_MII1_RXD1 to CTRL_CONF_MII1_RXD0	0x000c0001
rmii1_refclk	RMII1_REF_CLK (Driven by External 50-MHz Source)	0	CTRL_CONF_RMII1_REFCLK	0x00000001
mdio_data	MDIO	0	CTRL_CONF_MDIO_DATA	0x000a0000
mdio_clk	MDC	0	CTRL_CONF_MDIO_CLK	0x000e0000

5.2.7.3.5 SYSBOOT Signals

Some of the SYSBOOT pins have special meanings when EMAC boot is selected. SYSBOOT[5] is returns the EMAC PHY interface being used **(Note:** RGMII is automatically detected). Table 5-43 details more.

Table 5-43. Ethernet PHY Mode Selection

SYSBOOT[5]	PHY Mode
0b	MII
1b	RMII

SYSBOOT[18] provides a way to select a clock that can be generated by the device and output on CLKOUT2. This can be used to connect to an external ethernet PHY to eliminate the need for an external crystal or oscillator for the PHY. The frequency will differ depending on the PHY mode chosen with SYSBOOT[5]. More details in Table 5-44.

Table 5-44. Ethernet Clock Selection

SYSBOOT[18](1)(2)	CLKOUT2
0	CLKOUT2 not used
1	CLKOUT2 = 25MHz if SYSBOOT[5] = 0 CLKOUT2 = 50MHz if SYSBOOT[5] = 1

(1) The functionality provided by SYSBOOT[18] is only available on silicon revision PG1.2.

(2) If EMAC1 is not in the boot order, CLKOUT2 will not be configured, regardless of the value of SYSBOOT[18].

5.2.7.4 UART Boot Procedure

5.2.7.4.1 Device Initialization

- UART boot uses UART0.
- UART0 is configured to run at 115200 baud, 8-bit, no parity, 1 stop bit and no flow control.

5.2.7.4.2 Boot Image Download

- UART boot uses x-modem client protocol to receive the boot image.
- Utilities such as hyperterm, teraterm, minicom can be used on the PC side to download the boot image to the board
- With x-modem packet size of 1K throughout is roughly about 4KB/s.
- The ROM code will ping the host 10 times in 3s to start x-modem transfer. If host does not respond, UART boot will time out.

- Once the transfer starts, if the host does not send any packet for 3 seconds, UART boot will time out
- If the delay between two consecutive bytes of the same packet is more than 2 ms, the host is requested to re-transmit the entire packet again
- Error checking using the CRC-16 support in x-modem. If an error is detected, the host is requested to re-transmit the packet again.

5.2.7.4.3 Pins Used

Table 5-45 lists the device pins configured by the ROM for the UART boot. Not all pins are driven at boot time.

Table 5-45. Pins Used for UART Boot

Signal Name	Pin Used	CTRL_CONF Register	Register Setting
Rx	uart0_rxd	CTRL_CONF_UART0_RXD	0x080e0000
Tx	uart0_txd	CTRL_CONF_UART0_TXD	0x080b0000

5.2.7.5 USB Client (USB_CL) Boot Procedure

5.2.7.5.1 Device Initialization

The ROM Code supports booting from the USB interface in client (peripheral) mode under the following conditions:

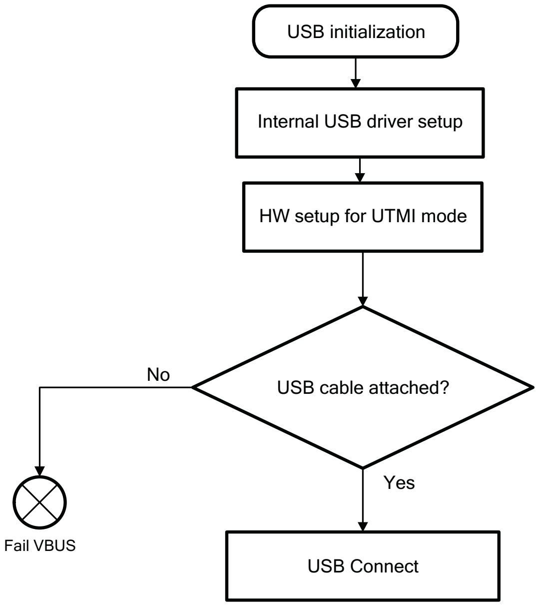
- Using the USB0 interface.
- USB operates in Full Speed, Client mode.
- USB will operate only in device powered mode with maximum power rating of 100mA.
- USB cable should be plugged in within 20 sec of powering up the device.
- ROM code uses the value of DATA POLARITY inversion feature for USB based on SYSBOOT[16].

5.2.7.5.2 Overview

If USB_CL boot is chosen by the SYSBOOT pin configuration:

- The USB hardware and PRCM clocks are configured for UTMI mode.
- The ROM Code continues with the USB procedure only if the USB cable is detected present (that is, VBUS is detected at transceiver level and communicated as such through the UTMI traffic). If not, the initialization procedure is aborted.
- The ROM code implements the RNDIS class driver.
- From user’s perspective, USB boot is indistinguishable from Ethernet boot. Refer to Section 5.2.7.3.2 for information on the BOOTP protocol.
- The USB initialization procedure is shown in Figure 5-25.

Figure 5-25. USB Initialization Procedure



5.2.7.5.3 Enumeration Descriptors

Table 5-46 lists the device descriptor parameters used during enumeration. The default Vendor ID and Product ID can be automatically overridden by the customer by programming the EFUSE that stores these values.

Table 5-46. Customized Descriptor Parameters

Parameter	Size (bytes)	TI Default Values
Device Class	1	02h (Communication Class)
Device Id Vendor (VID)	String	“Texas Instruments” (0x0451)
Device IProduct	String	“AM43xx 1.0”
Device Id Product (PID)	2	0x6142

Four endpoints are configured for USB:

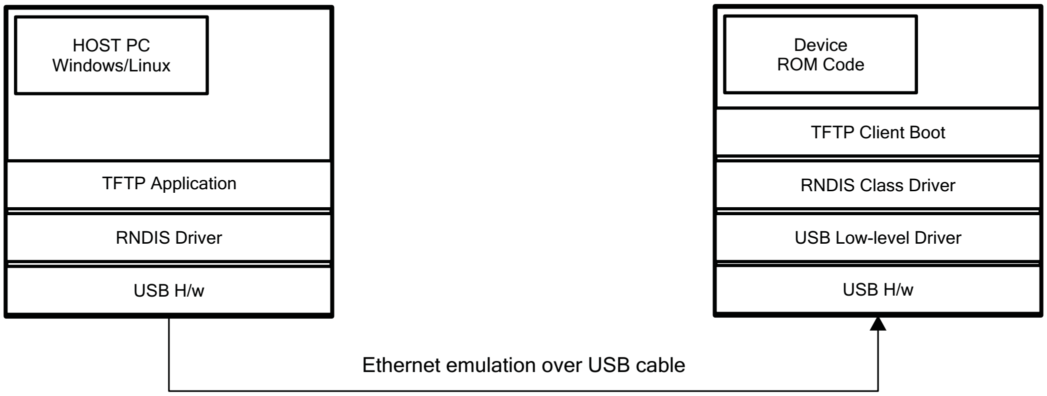
- One control endpoint (default)
- Two bulk endpoints: Data transmit and receive having interface class as “Communication Data”
- One interrupt endpoint used for RNDIS notifications

The interface class for USB is “Communication Control” and interface sub-class is “Abstract Line Control Model”.

5.2.7.5.4 Image Download Procedure

- The ROM implements as RNDIS class driver, so it enumerates as an Ethernet port.
- Standard RNDIS drivers present on Linux and Windows are picked up during the enumeration, no special drivers need to be installed.
- Once the enumeration is complete, the customer can download the boot image using any standard TFTP server application.

Figure 5-26. Image Transfer for USB Boot



5.2.7.5.5 Pins Used

Table 5-47 lists the device pins configured by the ROM for the USB boot mode. Not all pins are driven at boot time.

Table 5-47. Pins Used for USB_CL Boot

Signal Name	Pin Used	CTRL_CONF Register Setting
dm	usb0_dm	Not written by ROM
dp	usb0_dp	Not written by ROM
id	usb0_id	Not written by ROM
vbus	usb0_vbus	Not written by ROM

5.2.7.5.6 SYSBOOT Signals

Table 5-48 lists the SYSBOOT signals for USB client boot.

Table 5-48. SYSBOOT Signals for USB_CL Boot

SYSBOOT[16](1)	Used to select swapping of USB signals. 0b – USB DP/DM is not swapped. 1b – USB DP/DM is swapped.

(1) The functionality provided by SYSBOOT[16] is only available on silicon revision PG1.2.

5.2.7.6 ASIC ID Structure

The ASIC ID size is 81 bytes for Ethernet and USB. All fields of this structure are reserved.

5.2.8 Low Latency NOR Booting

5.2.8.1 Overview

The low latency NOR boot features:

- Possible only on GP Device
- ROM only supports connection to 16-bit memory
- Consists of a blind jump in ARM mode to a code located in an external NOR device connected to CS0
- The jump is performed with minimum on-chip ROM Code execution, without configuring any PLL
- Allows the customer to create its own booting code
- Set up by means of the configuration pins, see Table 5-10

5.2.8.2 Settings

The NOR device requires the following settings:

- Addr/Data muxed device or a non-muxed (selected using SYSBOOT[11]) device is connected in pinmux option 0 configuration (except for be1n), as mentioned in the Table 5-14 and Table 5-15, respectively. The be1n pin on the flash device needs to be connected to gpmc_csn2.
- Only 16-bit mode is supported (2)
- CS0 chip select
- Device wait signal connected to the WAIT0 mux config 0 GPMC signal (if used)
- The wait monitoring enable/disable is based on the value of SYSBOOT[9]

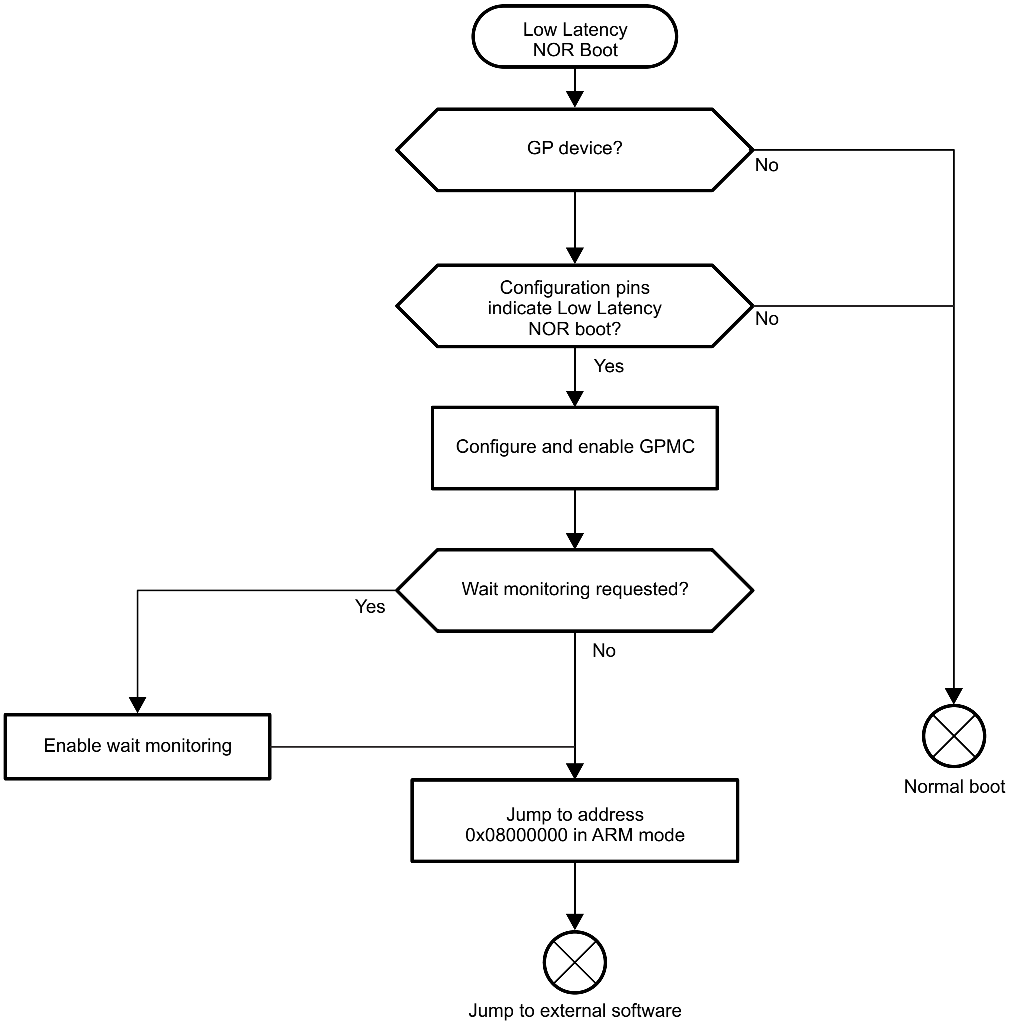
---

(2) See Table 5-10, *SYSBOOT Configuration Pins*.

5.2.8.3 External Booting

Figure 5-27 shows the Low Latency NOR boot procedure. The code does not use RAM and is designed for fast execution.

Figure 5-27. Low Latency NOR Boot



5.2.8.4 SYSBOOT Signals

Table 5-49 describes the SYSBOOT signals relevant to Low Latency NOR Boot (fast NOR)

Table 5-49. SYSBOOT Signals for Low Latency NOR Boot

SYSBOOT[8]	Must be set to 1 (only 16-bit device is supported)
SYSBOOT[9]	Used to determine if Wait is enabled. 0b – Ignore WAIT input 1b – Use WAIT input
SYSBOOT[11]	0b – Non-muxed device 1b – A/D-muxed device

5.2.9 Image Format

5.2.9.1 Overview

All preceding sections describe how the ROM Code searches and detects a boot image from a memory or a peripheral device type. This section describes the format of the boot image itself.

A boot image includes two major parts:

- The software to execute.
- A header containing the destination address and size of the image for non-XIP memory devices.

The mandatory section of a boot image contains the software loaded into the memory and executed. An overview of the image formats is shown in Figure 5-28:

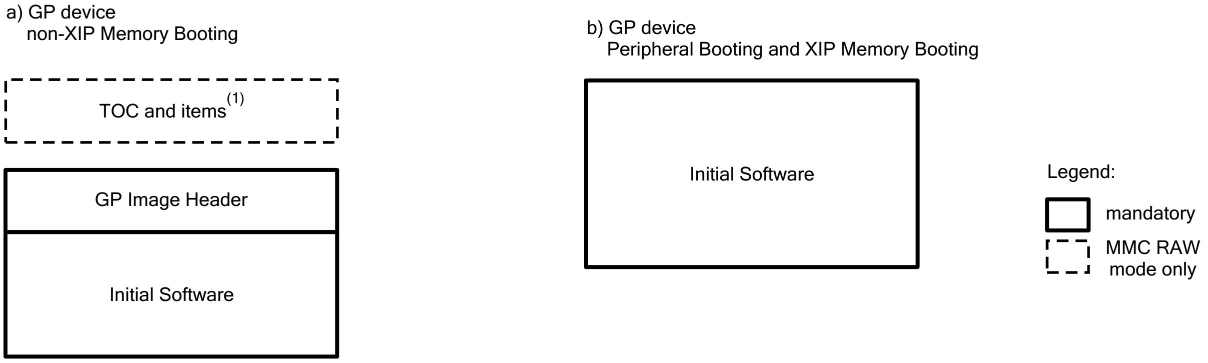
a. GP non-XIP Memory Booting:

   It is used for memories which require shadowing (for example, MMC). Image must begin with a GP header which contains information on image size and destination address. Only when MMC RAW mode is used must the image contain a TOC, in addition to the GP header.

b. GP Peripheral Booting and XIP Memory Booting:

   When memory device is of XIP type (for example, NOR) the GP header is not needed and the image can contain code for direct execution. The same image format is used for peripheral booting (where the code is transferred to internal RAM).

Figure 5-28. Image Formats on GP Devices



(1) TOC and items must fit into one 512-byte sector.

5.2.9.2 Image format for GP Device

A GP device involves no security, so keys and certificates are not required in the boot image.

When the booting memory device is non-XIP (for example, MMC) the image must contain a small header (referred as GP header) with the size of the image and the destination address where to store it.

The GP header is not required when booting from an XIP memory device (for example, NOR) or if peripheral booting. In this case, the peripheral or memory booting image starts directly with executable code.

Table 5-50. GP Device Image Format

Field	non-XIP Device (offset)	XIP Device (offset)	Size (bytes)	Description
Size	0000h	-	4	Size of the executable code plus 8-byte header
Destination	0004h	-	4	Address where to store the image / code entry point
Image	0008h	0000h	X	Executable code

Note: the “Destination” address field stands for both:

- Target address for the image copy from the non-XIP storage to the target XIP location (for example, internal RAM or SDRAM)
- Entry point for image code

The user must locate the code entry point to the target address for image copy.

5.2.10 Table of Contents

The Table of Contents (TOC) is a header needed only in GP devices while using MMC RAW mode. This must not be confused with the TOC used in HS devices. The TOC is 512 bytes long and consists of a maximum of 2 TOC items (32 bytes long each), located one after the other. The second TOC item must be filled by FFh. Each TOC item contains information required by the ROM Code to find a valid image in RAW mode, as illustrated in Table 5-51. To detect RAW mode, the ROM also needs the magic values mentioned in Table 5-52. Other than the TOC item fields and magic values, all the other bytes in the 512-byte TOC must be zero.

Table 5-51. The TOC Item Fields

Offset	Field	Size (bytes)	Description
0000h	Start	4	0x00000040
0004h	Size	4	0x0000000C
0008h	Flags	4	Not used, should be zero.
000Ch	Align	4	Not used, should be zero.
0010h	Load Address	4	Not used, should be zero.
0014h	Filename	12	12 character long name of sub image, including the zero (‘\0’) terminator. The ASCII representation is "CHSETTINGS".

Table 5-52. Magic Values for MMC RAW Mode

Offset	Value
40h	0xC0C0C0C1
44h	0x00000100

The ROM Code recognizes the TOC based on the filename described in Table 5-53.

Table 5-53. Filenames in TOC for GP Device

Filename	Description
CHSETTINGS	Magic string used by ROM

5.2.11 Services for HLOS Support – API

This Cortex core restricts accesses to few ARM coprocessor registers to the secure mode only. The GP Device forbids entering the secure mode and hence do not provide any secure services. However HLOS need to access secure registers for L2 cache maintenance.

For these purposes the ROM Code provides different primitives that can be called on GP or HS Device type. These services are implemented in monitor mode (service must be called by writing function ID into R12 register and using the SMC instruction) and do not use any resources like RAM/stack or hardware outside the MPU. The caller must save R0-R5 before using these APIs because the ROM code overwrites R0-R5.

The services include:

- L2 cache set debug register
- L2 cache clean and invalidate range of physical address
- L2 cache set control register
- L2 cache set auxiliary control
- L2 cache get control
- L2 cache set latency

Table 5-54. L2 Cache Set Debug Register

Function ID	**Description**	
R12 = 0x100	This function writes the PL310 Debug Control Register with the input given value in r0.	
Parameters		
Type	**Field**	**Description**
Input	32-bit unsigned integer – R0	Debug register value to set

Table 5-55. L2 Cache Clean and Invalidate Range of Physical Address

Function ID	**Description**	
R12 = 0x101	This function cleans and invalidates the range of the physical address given.	
Parameters		
Type	**Field**	**Description**
Input	32-bit unsigned integer – R0	Physical Start Address of range.
Input	32-bit unsigned integer – R1	Size of the range to invalidate.

Table 5-56. L2 Cache Set Control Register

Function ID	**Description**	
R12 = 0x102	This function writes the PL310 Control Register which enables and disables the L2 Cache.	
Parameters		
Type	**Field**	**Description**
Input	32-bit unsigned integer – R0	Control register value to set

Table 5-57. L2 Cache Set Auxiliary Control Register

Function ID	**Description**	
R12 = 0x109	This function writes a given input value into the PL310 Auxiliary Control Register.	
Parameters		
Type	**Field**	**Description**
Input	32-bit unsigned integer – R0	Auxiliary control register value to set

Table 5-58. L2 Cache Set Latency Control Register

Function ID	Description	
R12 = 0x112	This function writes the input values in the PL310 Tag and Data RAM Latency Control Register.	
Parameters		
Type	**Field**	**Description**
Input	32-bit unsigned integer – R0	TAG RAM Latency control register value to be set.
Input	32-bit unsigned integer – R1	Data RAM Latency to be set.

Table 5-59. L2 Cache Set Pre-fetch Control Register

Function ID	Description	
R12 = 0x113	This function sets the given input value in the L2 Cache Pre-fetch Control Register.	
Parameters		
Type	**Field**	**Description**
Input	32-bit unsigned integer – R0	Value to set in the Pre-fetch Control Register.

Table 5-60. L2 Cache Set Address Filtering Register

Function ID	Description	
R12 = 0x114	Writes the given input values to the PL130 Address Filtering Start and Address Filtering End registers.	
Parameters		
Type	**Field**	**Description**
Input	32-bit unsigned integer – R0	Value to set in address filtering start register.
Input	32-bit unsigned integer – R1	Value to be set in address filtering end register.

Table 5-61. L2 Cache Clean Set Way

Function ID	Description	
R12 = 0x115	Cleans the L2 cache by Set and Way.	
Parameters		
Type	**Field**	**Description**
Input	None	

Table 5-62. L1 Cache Set Pre-fetch Enable

Function ID	Description	
R12 = 0x116	Enable and disable L1 pre-fetch.	
Parameters		
Type	**Field**	**Description**
Input	32-bit unsigned integer – R0	0x0 = Disable pre-fetch. 0x1 = Enable pre-fetch.

Table 5-63. SCTLR Round-Robin Enable

Function ID	Description	
R12 = 0x117	Enable and disable round-robin replacement strategy for caches, BTAC, and micro TLBs.	
Parameters		
Type	**Field**	**Description**
Input	32-bit unsigned integer – R0	0x0 = Disable round-robin. 0x1 = Enable round-robin.

Table 5-64. CP15 Set ACTLR Register

Function ID	Description	
R12 = 0x118	This function sets the given input value in the CP15 ACTLR register.	
Parameters		
Type	**Field**	**Description**
Input	32-bit unsigned integer – R0	Value to set in the CP15 ACTLR register.

5.2.12 Tracing

Tracing in the Public ROM Code includes five 32-bit vectors for which each bit corresponds to a particular “way point” in the ROM Code execution sequence (see Table 5-6). Tracing vectors are initialized at the very beginning of the startup phase and updated all along the boot process.

Table 5-65. Tracing Vectors

Trace Vector	Bit	Group	Meaning
1	0	General	Passed the public reset vector
1	1	General	Entered main function
1	2	General	Running after the cold reset
1	3	Boot	Main booting routine entered
1	4	Memory Boot	Memory booting started
1	5	Peripheral Boot	Peripheral booting started
1	6	Reserved	Reserved
1	7	Boot	Header found
1	8	Boot	Reserved
1	9	Boot	Reserved
1	10	Peripheral Boot	Reserved
1	11	Peripheral Boot	Reserved
1	12	Peripheral Boot	Device initialized
1	13	Peripheral Boot	Searching for HOST (Bootp message in USB_CL/EMAC , ‘C’ character in UART)
1	14	Peripheral Boot	Image received
1	15	Peripheral Boot	Peripheral booting failed
1	16	Peripheral Boot	HOST not found (timeout)
1	17	Reserved	Reserved
1	18	Peripheral Boot	Image not received (timeout)
1	19	Peripheral Boot	Image received is bigger than expected
1	20	MMC Configuration Header	CHSETTINGS found
1	21	Reserved	Reserved
1	22	Reserved	Reserved
1	23	Reserved	Reserved
1	24	Reserved	Reserved
1	25	Reserved	Reserved
1	26	Reserved	Reserved
1	27	Reserved	Reserved
1	28	Reserved	Reserved
1	29	Reserved	Reserved
1	30	Reserved	Reserved
1	31	Reserved	Reserved
2	0	Reserved	Reserved
2	1	Reserved	Reserved
2	2	Reserved	Reserved
2	3	Reserved	Reserved
2	4	USB	USB connect
2	5	USB	USB_CL configured state
2	6	USB	USB_CL VBUS valid
2	7	USB	USB_CL session valid
2	8	Reserved	Reserved
2	9	Reserved	Reserved

Table 5-65. Tracing Vectors (continued)

Trace Vector	Bit	Group	Meaning
2	10	Reserved	Reserved
2	11	Reserved	Reserved
2	12	Memory Boot	Memory booting trial 0
2	13	Memory Boot	Memory booting trial 1
2	14	Memory Boot	Memory booting trial 2
2	15	Memory Boot	Memory booting trial 3
2	16	Memory Boot	Execute GP image
2	17	Reserved	Reserved
2	18	Memory and Peripheral Boot	Jumping to Initial SW
2	19	Reserved	Reserved
2	20	Reserved	Reserved
2	21	Reserved	Reserved
2	22	Reserved	Reserved
2	23	Reserved	Reserved
2	24	Reserved	Reserved
2	25	Reserved	Reserved
2	26	Reserved	Reserved
2	27	Reserved	Reserved
2	28	Reserved	Reserved
2	29	Reserved	Reserved
2	30	Reserved	Reserved
2	31	Reserved	Reserved
3	0	Memory Boot	Memory booting device NULL
3	1	Memory Boot	Memory booting device XIP
3	2	Memory Boot	Memory booting device NAND
3	3	Memory Boot	Memory booting device NAND_I2C
3	4	Memory Boot	Memory booting device MMCSD0
3	5	Memory Boot	Memory booting device MMCSD1
3	6	Memory Boot	Memory booting device SPI
3	7	Memory Boot	Memory booting device QSPI
3	8	Reserved	Reserved
3	9	Reserved	Reserved
3	10	Reserved	Reserved
3	11	Reserved	Reserved
3	12	Reserved	Reserved
3	13	Reserved	Reserved
3	14	Reserved	Reserved
3	15	Reserved	Reserved
3	16	Peripheral Boot	Peripheral booting device UART0
3	17	Reserved	Reserved
3	18	Reserved	Reserved
3	19	Reserved	Reserved
3	20	Peripheral Boot	Peripheral booting device USB_CL
3	21	Peripheral Boot	Peripheral booting device USB_MS
3	22	Peripheral Boot	Peripheral booting device CPGMAC1
3	23	Reserved	Reserved
3	24	Peripheral Boot	Peripheral booting device NULL

Table 5-65. Tracing Vectors (continued)

Trace Vector	Bit	Group	Meaning
3	25	Reserved	Reserved
3	26	Reserved	Reserved
3	27	Reserved	Reserved
3	28	Reserved	Reserved
3	29	Reserved	Reserved
3	30	Reserved	Reserved
3	31	Reserved	Reserved
4	0	Memory Boot - NOR	Non-Muxed NOR detected
4	1	Memory Boot - NOR/NAND	NOR/NAND Wait 1 Selected
4	2	Memory Boot - NOR/QSPI	NOR/QSPI pin mux mode option 0 selected
4	3	Memory Boot - NOR/QSPI	NOR/QSPI pinmux option 1 selected
4	4	Memory Boot - NOR	NOR pinmux mode option 2 selected
4	5	Memory Boot - NAND	NAND 16bit BUS Width detected
4	6	Memory Boot - NAND	NAND ECC Failure found
4	7	Memory Boot - NAND	NAND Device Identified
4	8	Memory Boot - NAND	BCH 16 ECC Scheme used
4	9	Memory Boot	MMC Card in Ready State(CMD1 complete)
4	10	Memory Boot	Data Read from the MMC Card
4	11	Memory Boot - USB_MS/MMC	Master Boot record found
4	12	Memory Boot - USB_MS/MMC	Active Partition Found
4	13	Memory Boot-MMC	Raw Image found
4	14	Memory Boot - (MMC/USB_MS)	MLO found
4	15	Memory Boot	Reserved
4	16	Memory Boot - USB_MS	Device protocol supported
4	17	Memory Boot - USB_MS	Mass Storage Class enumeration completed
4	18	Memory Boot – SPI	SPI configuration completed
4	19	Memory Boot – SPI	SPI Read Initialized
4	20	Memory Boot	Reserved
4	21	Memory Boot	Reserved
4	22	Memory Boot	Reserved
4	23	Memory Boot	Reserved
4	24	Memory Boot	Reserved
4	25	Memory Boot	Reserved
4	26	Memory Boot	Reserved
4	27	Memory Boot	Reserved
4	28	Memory Boot	Reserved
4	29	Memory Boot	Reserved
4	30	Memory Boot	Reserved
4	31	Memory Boot	Reserved
5	0	Peripheral Boot - EMAC	RMII PHY detected
5	1	Peripheral Boot - EMAC	RGMII PHY detected
5	2	Peripheral Boot - EMAC	MII PHY detected
5	3	Peripheral Boot - EMAC	GMII PHY detected - Reserved
5	4	Peripheral Boot - EMAC	10 Mbps Network detected
5	5	Peripheral Boot - EMAC	100 Mbps Network detected
5	6	Peripheral Boot - EMAC	1 Gbps Network detected
5	7	Peripheral Boot - EMAC	RGMII internal delay enabled

Table 5-65. Tracing Vectors (continued)

Trace Vector	Bit	Group	Meaning
5	8	Peripheral Boot	Reserved
5	9	Peripheral Boot	Reserved
5	10	Peripheral Boot	Reserved
5	11	Peripheral Boot	Reserved
5	12	Peripheral Boot	Reserved
5	13	Peripheral Boot	Reserved
5	14	Peripheral Boot	Reserved
5	15	Peripheral Boot	Reserved
5	16	Peripheral Boot - USB_CL/UART	TFTP transfer started
5	17	Peripheral Boot- USB_CL/UART	TFTP transfer completed
5	18	Peripheral Boot - USB_CL/UART	TFTP timeout occurred
5	19	Peripheral Boot - UART	Xmodem 1K protocol selected
5	20	Peripheral Boot	Reserved
5	21	Peripheral Boot	Reserved
5	22	Peripheral Boot	Reserved
5	23	Peripheral Boot	Reserved
5	24	Peripheral Boot	Reserved
5	25	Peripheral Boot	Reserved
5	26	Peripheral Boot	Reserved
5	27	Peripheral Boot	Reserved
5	28	Peripheral Boot	Reserved
5	29	Peripheral Boot	Reserved
5	30	Peripheral Boot	Reserved
5	31	Peripheral Boot	Reserved

Power, Reset, and Clock Management (PRCM)

This chapter describes the PRCM of the device.

Topic **Page**

6.1 Introduction

The device power-management architecture ensures maximum performance and operation time for user satisfaction (audio/video support) while offering versatile power-management techniques for maximum design flexibility, depending on application requirements. This introduction contains the following information:

- Power-management architecture building blocks for the device
- State-of-the-art power-management techniques supported by the power-management architecture of the device

6.2 Device Power-Management Architecture Building Blocks

To provide a versatile architecture supporting multiple power-management techniques, the power-management framework is built with three levels of resource management: clock, power, and voltage management.

These management levels are enforced by defining the managed entities or building blocks of the power-management architecture, called the clock, power, and voltage domains. A domain is a group of modules or subsections of the device that share a common entity (for example, common clock source, common voltage source, or common power switch). The group forming the domain is managed by a policy manager. For example, a clock for a clock domain is managed by a dedicated clock manager within the power, reset, and clock management (PRCM) module. The clock manager considers the joint clocking constraints of all the modules belonging to that clock domain (and, hence, receiving that clock).

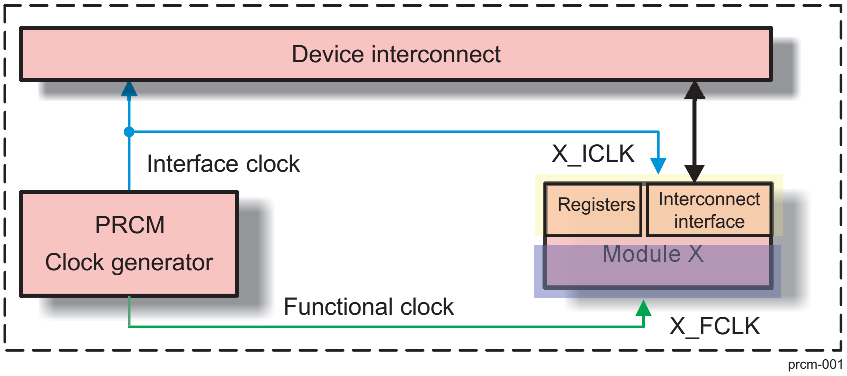
6.3 Clock Management

The PRCM module along with the control module manages the gating (that is, switching off) and enabling of the clocks to the device modules. The clocks are managed based on the requirement constraints of the associated modules. The following sections identify the module clock characteristics, management policy, clock domains, and clock domain management

6.3.1 Module Interface and Functional Clocks

Each module within the device has specific clock input characteristic requirements. Based on the characteristics of the clocks delivered to the modules, the clocks are divided into two categories: interface clocks and functional clocks

Figure 6-1. Functional and Interface Clocks



The interface clocks have the following characteristics:

- They ensure proper communication between any module/subsystem and interconnect.
- In most cases, they supply the system interconnect interface and registers of the module.
- A typical module has one interface clock, but modules with multiple interface clocks may also exist (that is, when connected to multiple interconnect buses).
- Interface clock management is done at the device level.
- From the standpoint of the PRCM module, an interface clock is identified by an _ICLK suffix.

Functional clocks have the following characteristics:

- They supply the functional part of a module or subsystem.
- A module can have one or more functional clocks. Some functional clocks are mandatory, while others are optional. A module needs its mandatory clock(s) to be operational. The optional clocks are used for specific features and can be shut down without stopping the module activity
- From the standpoint of the PRCM module, a functional clock is distributed directly to the related modules through a dedicated clock tree. It is identified with an _FCLK suffix

6.3.2 Module-Level Clock Management

Each module in the device may also have specific clock requirements. Certain module clocks must be active when operating in specific modes, or may be gated otherwise. Globally, the activation and gating of the module clocks are managed by the PRCM module. Hence, the PRCM module must be aware of when to activate and when to gate the module clocks. The PRCM module differentiates the clock-management behavior for device modules based on whether the module can initiate transactions on the device interconnect (called master module or initiators) or cannot initiate transactions and only responds to the transactions initiated by the master (called slave module or targets). Thus, two hardware-based power-management protocols are used:

- Master standby protocol: Clock-management protocol between the PRCM and master modules.
- Slave idle protocol: Clock-management protocol between the PRCM and slave modules.

6.3.2.1 Master Standby Protocol

Master standby protocol is used to indicate that a master module must initiate a transaction on the device interconnect and requests specific (functional and interface) clocks for the purpose. The PRCM module ensures that the required clocks are active when the master module requests the PRCM module to enable them. This is called a module wake-up transition and the module is said to be functional after this transition completes. Similarly, when the master module no longer requires the clocks, it informs the PRCM module, which can then gate the clocks to the module. The master module is then said to be in standby mode. Although the protocol is completely hardware-controlled, software must configure the clock-management behavior for the module. This is done by setting the module register bit field <Module>_SYSCONFIG.MIDLEMODE or <Module>_SYSCONFIG.STANDBYMODE. The behavior, identified by standby mode values, must be configured.

Table 6-1. Master Module Standby-Mode Settings

Standby Mode Value	Selected Mode	Description
0x0	Force-standby	The module unconditionally asserts the standby request to the PRCM module, regardless of its internal operations. The PRCM module may gate the functional and interface clocks to the module. This mode must be used carefully because it does not prevent the loss of data at the time the clocks are gated.
0x1	No-standby	The module never asserts the standby request to the PRCM module. This mode is safe from a module point of view because it ensures that the clocks remain active. However, it is not efficient from a power-saving perspective because it never allows the output clocks of the PRCM module to be gated
0x2	Smart-standby	The module asserts the standby request based on its internal activity status. The standby signal is asserted only when all ongoing transactions are complete and the module is idled. The PRCM module can then gate the clocks to the module.

Table 6-1. Master Module Standby-Mode Settings (continued)

Standby Mode Value	Selected Mode	Description
0x3	Smart-standby wakeup-capable mode	The module asserts the standby request based on its internal activity status. The standby signal is asserted only when all ongoing transactions are complete and the module is idle. The PRCM module can then gate the clocks to the module. The module may generate (master-related) wake-up events when in STANDBY state. The mode is relevant only if the appropriate module mwakeup output is implemented.

The standby status of a master module is indicated by the CM_<Power_domain>_<Module>_CLKCTRL[x]. STBYST bit in the PRCM module.

Table 6-2. Master Module Standby Status

STBYST Bit Value	Description
0x0	The module is functional.
0x1	The module is in standby mode

6.3.2.2 Slave Idle Protocol

This hardware protocol allows the PRCM module to control the state of a slave module. The PRCM module informs the slave module, through assertion of an idle request, when its clocks (interface and functional) can be gated. The slave can then acknowledge the request from the PRCM module and the PRCM module is then allowed to gate the clocks to the module. A slave module is said to be in IDLE state when its clocks are gated by the PRCM module. Similarly, an idled slave module may need to be wakened because of a service request from a master module or as a result of an event (called a wake-up event; for example, interrupt or DMA request) received by the slave module. In this situation the PRCM module enables the clocks to the module and then deasserts the idle request to signal the module to wake up. Although the protocol is completely hardware-controlled, software must configure the clock-management behavior for the slave module. This is done by setting the module register bit field <Module>_SYSCONFIG. SIDLEMODE or <Module>_SYSCONFIG. IDLEMODE. The behavior, listed in the Idle Mode Value column, must be configured by software.

Table 6-3. Module Idle Mode Settings

Idle Mode Value	Selected Mode	Description
0x0	Force-idle	The module unconditionally acknowledges the idle request from the PRCM module, regardless of its internal operations. This mode must be used carefully because it does not prevent the loss of data at the time the clock is switched off.
0x1	No-idle	The module never acknowledges any idle request from the PRCM module. This mode is safe from a module point of view because it ensures that the clocks remain active. However, it is not efficient from a power-saving perspective because it does not allow the PRCM module output clock to be shut off, and thus the power domain to be set to a lower power state.

Table 6-3. Module Idle Mode Settings (continued)

Idle Mode Value	Selected Mode	Description
0x2	Smart-idle	The module acknowledges the idle request basing its decision on its internal activity. Namely, the acknowledge signal is asserted only when all pending transactions, interrupts, or direct memory access (DMA) requests are processed. This is the best approach to efficient system power management.
0x3	Smart-idle wakeup-capable mode	The module acknowledges the idle request basing its decision on its internal wakeup-capable mode activity. Namely, the acknowledge signal is asserted only when all pending transactions, interrupts, or DMA requests are processed. This is the best approach to efficient system power management. The module may generate (IRQ- or DMA-request-related) wake-up events when in IDLE state. The mode is relevant only if the appropriate module swakeup output(s) is implemented.

The idle status of a slave module is indicated by the CM_<Powerdomain>_<Module>_CLKCTRL[x] IDLEST bit field in the PRCM module.

Table 6-4. Idle States for a Slave Module

IDLEST Bit VALUE	Idle Status	Description
0x0	Functional	The module is fully functional.The interface and functional clocks are active.
0x1	In transition	The module is performing a wake-up or a sleep transition.
0x2	Interface idle	The module interface clock is idled. The module may remain functional if using a separate functional clock.
0x3	Full idle	The module is fully idle. The interface and functional clocks are gated in the module.

For the idle protocol management on the PRCM module side, the behavior of the PRCM module is configured in the CM_<Power domain>_<module>_CLKCTRL[x] MODULEMODE bit field. Based on the configured behavior, the PRCM module asserts the idle request to the module unconditionally (that is, immediately when the software requests).

Table 6-5. Slave Module Mode Settings in PRCM

MODULEMODE Bit VALUE	Selected Mode	Description
0x0	Disabled	The PRCM module unconditionally asserts the module idle request. This request applies to the gating of the functional and interface clocks to the module. If acknowledged by the module, the PRCM module can gate all clocks to the module (that is, the module is completely disabled)..
0x1	Reserved	NA

Table 6-5. Slave Module Mode Settings in PRCM (continued)

MODULEMODE Bit VALUE	Selected Mode	Description
0x2	Enabled	This mode applies to a module when the PRCM module manages its interface and functional clocks. The functional clock to the module remains active unconditionally, while the PRCM module automatically asserts/deasserts the module idle request based on the clock-domain transitions. If acknowledged by the module, the PRCM module can gate only the interface clock to the module.
0x3	Reserved	NA

In addition to the IDLE and STANDBY protocol, PRCM offers also the possibility to manage optional clocks, through a direct SW control: “OptFclken” bit from programming register.

Table 6-6. Module Clock Enabling Condition

<table>
<tr><th colspan="4">Clock Enabling</th></tr>
<tr><td rowspan="3">Clock associated with STANDBY protocol</td><td rowspan="3">AND</td><td colspan="2">Clock Domain is 'ACTIVE'</td></tr>
<tr><td rowspan="2">OR</td><td>MStandby is de-asserted</td></tr>
<tr><td>Mwakeup is asserted</td></tr>
<tr><td rowspan="4">Clock associated with IDLE protocol, as interface clock</td><td rowspan="4">AND</td><td colspan="2">Clock Domain is 'ACTIVE'</td></tr>
<tr><td rowspan="3">OR</td><td>Idle status = FUNCT</td></tr>
<tr><td>Idle status = TRANS</td></tr>
<tr><td>SWakeup is asserted</td></tr>
<tr><td rowspan="5">Clock associated with IDLE protocol, as functional clock</td><td rowspan="5">AND</td><td colspan="2">Clock Domain is 'ACTIVE'</td></tr>
<tr><td rowspan="4">OR</td><td>Idle status = FUNCT</td></tr>
<tr><td>Idle status = IDLE</td></tr>
<tr><td>Idle status = TRANS</td></tr>
<tr><td>SWakeup is asserted</td></tr>
<tr><td rowspan="2">Optional clock</td><td rowspan="2">AND</td><td colspan="2">Clock domain is ready</td></tr>
<tr><td colspan="2">OptFclken=Enabled ('1')</td></tr>
</table>

6.3.3 Clock Domain

A clock domain is a group of modules fed by clock signals controlled by the same clock manager in the PRCM module By gating the clocks in a clock domain, the clocks to all the modules belonging to that clock domain can be cut to lower their active power consumption (that is, the device is on and the clocks to the modules are dynamically switched to ACTIVE or INACTIVE (GATED) states). Thus, a clock domain allows control of the dynamic power consumption of the device. The device is partitioned into multiple clock domains, and each clock domain is controlled by an associated clock manager within the PRCM module. This allows the PRCM module to individually activate and gate each clock domain of the device

Figure 6-2. Generic Clock Domain

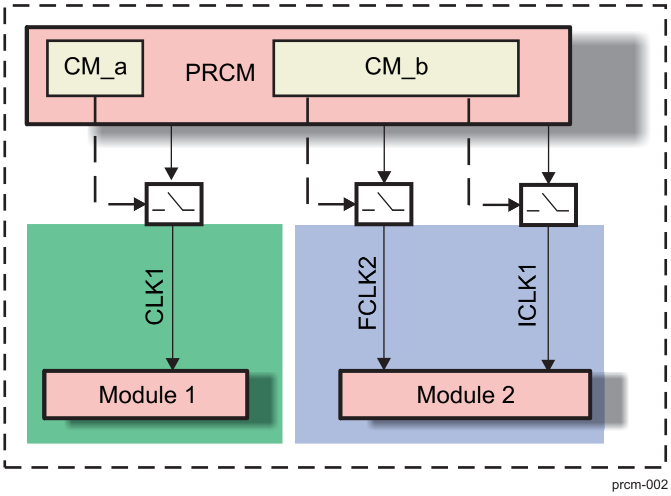


Figure above is an example of two clock managers: CM_a and CM_b. Each clock manager manages a clock domain. The clock domain of CM_b is composed of two clocks: a functional clock (FCLK2) and an interface clock (ICLK1), while the clock domain of CM_a consists of a clock (CLK1) that is used by the module as a functional and interface clock. The clocks to Module 2 can be gated independently of the clock to Module 1, thus ensuring power savings when Module 2 is not in use. The PRCM module lets software check the status of the clock domain functional clocks. The CM_<Clock domain>_CLKSTCTRL[x] CLKACTIVITY_<FCLK/Clock name_FCLK> bit in the PRCM module identifies the state of the functional clock(s) within the clock domain. Table shows the possible states of the functional clock.

Table 6-7. Clock Domain Functional Clock States

CLKACTIVITY BIT Value	Status	Description
0x0	Gated	The functional clock of the clock domain is inactive
0x1	Active	The functional clock of the clock domain is running

6.3.3.1 Clock Domain-Level Clock Management

The domain clock manager can automatically (that is, based on hardware conditions) and jointly manage the interface clocks within the clock domain. The functional clocks within the clock domain are managed through software settings. A clock domain can switch between three possible states: ACTIVE, IDLE_TRANSITION, and INACTIVE. Figure 6-3 shows the sleep and wake-up transitions of the clock domain between ACTIVE and INACTIVE states.

Figure 6-3. Clock Domain State Transitions

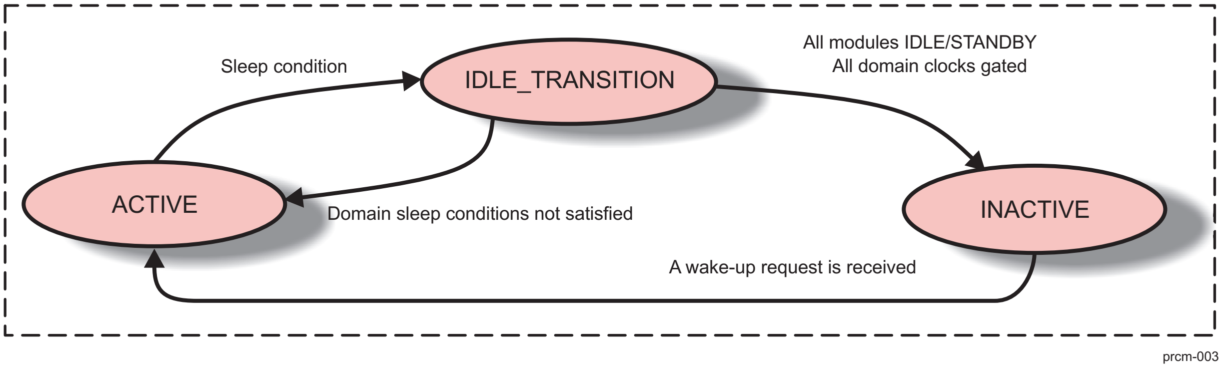


Table 6-8. Clock Domain States

State	Description
ACTIVE	Every nondisabled slave module (that is, those whose MODULEMODE value is not set to disabled) is put out of IDLE state. All interface clocks to the nondisabled slave modules in the clock domain are provided. All functional and interface clocks to the active master modules (that is, not in STANDBY) in the clock domain are provided. All enabled optional clocks to the modules in the clock domain are provided.
IDLE_TRANSITION	This is a transitory state. Every master module in the clock domain is in STANDBY state. Every idle request to all the slave modules in the clock domain is asserted. The functional clocks to the slave module in enabled state (that is, those whose MODULEMODE values are set to enabled) remain active. All enabled optional clocks to the modules in the clock domain are provided.
INACTIVE	All clocks within the clock domain are gated. Every slave module in the clock is in IDLE state and set to disabled. Every slave module in the clock domain (that is, those whose MODULEMODE is set to disabled) is in IDLE state and set to disabled. Every optional functional clock in the clock domain is gated

Each clock domain transition behavior is managed by an associated register bit field in the CM_<Clock domain>_CLKSTCTRL[x] CLKTRCTRL PRCM module

Table 6-9. Clock Transition Mode Settings

CLKTRCTRL Bit Value	Selected Mode	Description
0x0	NO_SLEEP	Sleep transition cannot be initiated. Wakeup transition may however occur.
0x1	SW_SLEEP	A software-forced sleep transition. The transition is initiated when the associated hardware conditions are satisfied
0x2	SW_WKUP	A software-forced clock domain wake-up transition is initiated
0x3	Reserved	NA

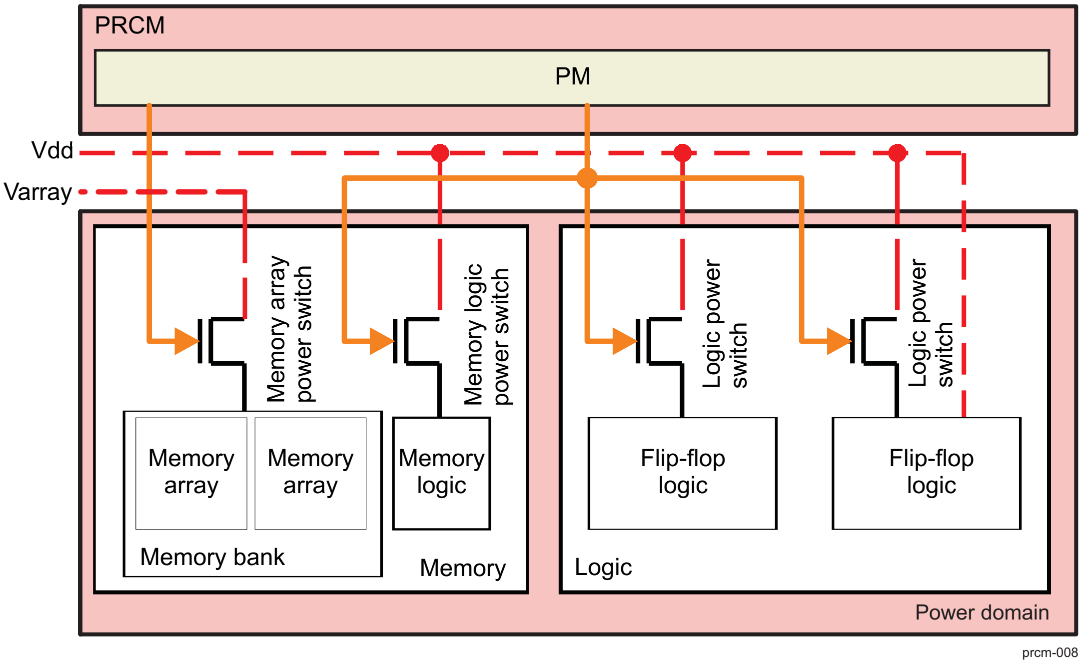
6.4 Power Management

The PRCM module manages the switching on and off of the power supply to the device modules. To minimize device power consumption, the power to the modules can be switched off when they are not in use. Independent power control of sections of the device allows the PRCM module to turn on and off specific sections of the device without affecting the others.

6.4.1 Power Domain

A power domain is a section (that is, a group of modules) of the device with an independent and dedicated power manager (see Figure 6-4). A power domain can be turned on and off without affecting the other parts of the device.

Figure 6-4. Generic Power Domain Architecture



To minimize device power consumption, the modules are grouped into power domains. A power domain can be split into a logic area and a memory area.

Table 6-10. States of a Memory Area in a Power Domain

State	Description
ON	The memory array is powered and fully functional
OFF	The memory array is powered down

Table 6-11. States of a Logic Area in a Power Domain

State	Description
ON	Logic is fully powered
OFF	Logic power switches are off. All the logic (DFF) is lost

6.4.2 Power Domain Management

The power manager associated with each power domain is assigned the task of managing the domain power transitions. It ensures that all hardware conditions are satisfied before it can initiate a power domain transition from a source to a target power state

Table 6-12. Power Domain Control and Status Registers

Register/Bit Field	Type	Description
PM_<Power domain>_PWRSTCTRL[1:0] POWERSTATE	Control	Selects the target power state of the power domain among OFF, ON, or RETENTION.
PM_<Power domain>_PWRSTST[1:0] POWERSTATEST	Status	Identifies the current state of the power domain. It can be OFF, ON, or RETENTION.
PM_<Power domain>_PWRSTST[2] LOGICSTATEST	Status	Identifies the current state of the logic area in the power domain. It can be OFF or ON.

Table 6-12. Power Domain Control and Status Registers (continued)

Register/Bit Field	Type	Description
PM_<Power domain>_PWRSTST[5:4] MEMSTATEST	Status	Identifies the current state of the memory area in the power domain. It can be OFF, ON, or RETENTION

6.4.2.1 Power Management Techniques

The following section describes the state-of-the-art power management techniques supported by the device.

6.4.2.1.1 Dynamic Voltage Frequency Scaling (DVFS)

DVFS is a power management technique where the operating voltage and frequency are dynamically scaled across device Operating Performance Points (OPP). An OPP is a voltage/frequency pair that defines a specific power state. For each OPP, software sends control signals to external regulators in order to set the minimum voltage.

6.4.2.1.1.1 Switching and Sequencing

DVFS switching primarily involves the following aspects:

- DVFS switching must always be ensured that the voltage of the external regulator is always changed to the right OPP voltage.
  - When moving from higher to lower OPP: Frequency changes first and then voltage
  - When moving from lower to higher OPP: Voltage changes first and then frequency

6.4.2.1.2 MPU DPS and DCG

The MPU supports dynamic clock gating (DCG / HW_AUTO) feature as part of the PRCM’s Clock Domain Management. For active usecase sensitive to power, this feature can be used to save power. MPU will automatically be clock gated when it enter’s WFI state. Wakeup will be via interrupts. Having DCG feature enables dynamic power switching (DPS) using the power domain management of PRCM.

6.4.3 Power Modes

The following is a high-level description of the different power modes of the device. They are listed in order from highest power consumption, lowest wakeup latency (Standby), to lowest power consumption, highest wakeup latency (RTC-only). If your application requires some sort of power management, you must determine which power mode level described below satisfies your requirements. Each level must be evaluated based on power consumed and latency (the time it takes to wakeup to Active mode). Specific values are detailed in the device-specific data sheet. Note that not all modes are supported by software packages supplied by Texas Instruments.

Table 6-13. Typical Power Modes

Power Modes	Application State	Power Domains, Clocks, and Voltage Supply States
Active	All Features	**Power supplies:** All power supplies are ON. VDD_MPU = 1.1 V (nom) VDD_CORE = 1.1 V (nom) **Clocks:** Main Oscillator (OSC0) = ON All DPLLs are locked. **Power domains:** PD_PER = ON PD_MPU = ON PD_GFX = ON or OFF (depending on use case) PD_WKUP = ON DDR is active.
Standby	DDR memory is in self-refresh and contents are preserved. Wakeup from any GPIO. Cortex-A9 context/register contents are lost and must be saved before entering standby. On exit, context must be restored from DDR. For wakeup, boot ROM executes and branches to system resume.	**Power supplies:** All power supplies are ON. VDD_MPU = 0.95 V (nom) VDD_CORE = 0.95 V (nom) **Clocks:** Main Oscillator (OSC0) = ON All DPLLs are in bypass. **Power domains:** PD_PER = ON PD_MPU = OFF PD_GFX = OFF PD_WKUP = ON DDR is in self-refresh.
Deepsleep	PD_PER peripheral and Cortex-A9/MPU register information will be lost. On-chip peripheral register (context) information of PD-PER domain needs to be saved by application to SDRAM before entering this mode. DDR is in self-refresh. For wakeup, boot ROM executes and branches to peripheral context restore followed by system resume.	**Power supplies:** All power supplies are ON. VDD_MPU = 0.95 V (nom) VDD_CORE = 0.95 V (nom) **Clocks:** Main Oscillator (OSC0) = OFF All DPLLs are in bypass. **Power domains:** PD_PER = OFF (except 64KB L3 OCMC retained) PD_MPU = OFF (except MPU OCM RAM retained) PD_GFX = OFF PD_WKUP = ON DDR is in self-refresh.

Table 6-13. Typical Power Modes (continued)

Power Modes	Application State	Power Domains, Clocks, and Voltage Supply States
RTC-Only	RTC timer remains active and all other device functionality is disabled.	**Power supplies:** All power supplies are OFF except VDDS_RTC. VDD_MPU = 0 V VDD_CORE = 0 V **Clocks:** Main Oscillator (OSC0) = OFF **Power domains:** All power domains are OFF.

6.4.3.1 Active

In Active mode, the supply to all voltage rails must be maintained. All power domains come up in ON state and the device is fully functional.

6.4.3.2 Standby

The device can be placed in Standby mode to reduce power consumption during low activity levels. This first level of power management allows you to maintain the device context for fast resume times. The main characteristics of this mode which distinguish it from Active mode are:

- All modules are clock gated except GPIOs
- PLLs may be placed in bypass mode if downstream clocking does not require full performance
- Voltage domains VDD_MPU and VDD_CORE voltage levels can be reduced to OPP50 levels because the required performance of the entire device is reduced
- MPU power domain (PD_MPU) is in OFF state
- DDR memory is in low power self-refresh mode.

Further power reduction can be achieved in this mode if the RTC function is not required. See Section 6.4.3.5, *Internal RTC LDO*.

The above conditions result in lower power consumption than Active mode but require the user to save the MPU context to OCMC RAM or DDR to resume properly upon wakeup. Contents of the internal SRAM are lost because PD_MPU is turned OFF. Wakeup in Standby mode is achieved using any GPIO. GPIO wakeup is possible by switching the pad to GPIO mode and configuring the corresponding GPIO bank for generating an interrupt to the MPUSS. Note that pads that do not have a GPIO muxmode (for example, ADC or USB), cannot cause these wakeups. If additional or other wakeup sources are required, the associated peripheral module clock and interconnect clock domain should remain enabled (this may require the associated PLL to remain locked) and the module must be configured appropriately for wakeup by configuring it to generate an interrupt to the MPUSS.

6.4.3.3 DeepSleep

DeepSleep mode enables lower power consumption than Standby. The main characteristics of the mode which distinguish it from other higher power modes are:

- All on-chip power domains are shut off (except PD_WKUP and PD_RTC remain ON) to reduce power leakage
- VDD_CORE power (except VDDA analog) to DPLLs is turned OFF using CTRL_DPLL_PWR_SW register
- VDDS_SRAM_CORE_BG is in retention using CTRL_VSLDO.vsldo_core_auto_ramp_en

DeepSleep mode is typically used during periods of inactivity when the user requires very low power while waiting for an event that requires processing or higher performance. This is the lowest power mode which still includes DDR in self-refresh, so wakeup events do not require a full cold boot, which greatly reduces wakeup latencies over RTC-only mode.

Further power reduction can be achieved in this mode if the RTC function is not required. See Section 6.4.3.5, *Internal RTC LDO*.

The contents of the internal SRAM are lost because PD_MPU is turned OFF.

Before entering DeepSleep mode, peripheral and MPU context must be saved in the DDR. Upon wakeup, the boot ROM executes and checks to see if it has resumed from a DeepSleep state. If so, it redirects to the DDR to continue the resume process. Because power to PD_WKUP is ON throughout DeepSleep, power to key modules such as GPIO0, I2C, and others is maintained to allow wakeup events to exit out of this mode. In addition, power to OCMC RAM is maintained to preserve information internally during DeepSleep.

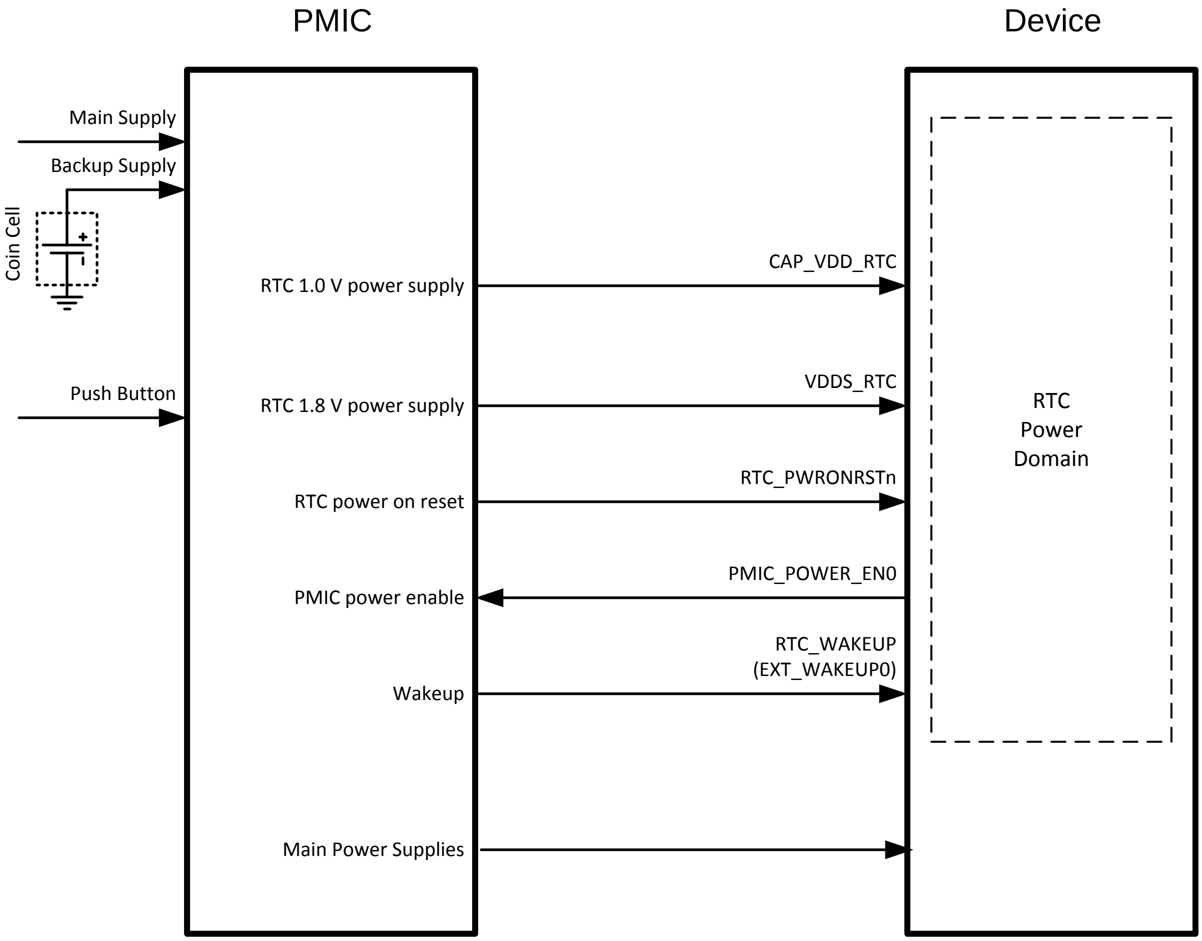
Activity on wakeup peripherals via wakeup events enables the master crystal oscillator using the oscillator control circuit. The wakeup events also interrupt the wakeup processor, which controls proper enabling of power domains and clocks in the PRCM. See Section 6.4.5, *Wakeup Sources/Events*, for details on wakeup sources during DeepSleep and other low power modes mentioned.

6.4.3.4 RTC-Only

RTC-only mode is an ultra-low power mode which allows the user to maintain power and clocks to the real-time clock (RTC) domain while the rest of the device is powered down. All context and memories will be lost, and the only portion of the chip that will be maintained is the RTC. Only the RTC power supply must be ON. All the remaining supplies must be OFF. The RTC battery backup domain consists of the RTC subsystem (RTCSS), a dedicated on-chip 32.768 Hz crystal oscillator, and I/Os associated with the RTCSS: pmic_power_enable0 and RTC_WAKEUP (EXT_WAKEUP0).

Figure 6-5 gives a high level view of system which implements the RTC-only mode.

Figure 6-5. High Level System View for RTC-only Mode



Wakeup from RTC-only mode can only be achieved using the RTC_WAKEUP (EXT_WAKEUP0) signal. Once a wakeup is triggered using this signal, the device drives pmic_pwr_enable0 to initiate a power-up sequence by the PMIC. The device must go through a full cold boot upon wakeup from RTC-only mode.

6.4.3.5 Internal RTC LDO

The device contains an internal LDO (low dropout) regulator which powers the RTC digital core. Depending on your application, you may be able to disable this regulator to save power in low power use cases.

If your application never uses the RTC functionality, connect RTC_KALDO_ENn to VDDS_RTC, CAP_VDD_RTC to VDD_CORE, and RTC_PWRONRSTn to ground. These connections disable the internal RTC LDO because RTC_KALDO_ENn is high, and they use the external VDD_CORE supply to power the RTC digital core. The RTC LDO must be disabled for internal power sequencing even though the RTC is not used. Grounding the reset signal will ensure the RTC stays in reset. Disabling the internal LDO will allow the application to achieve lower power consumption in all the low power modes.

If your application uses the RTC functionality and never needs RTC-only mode, the hardware scenario is similar to the previous description, but the RTC reset signal can be connected to the device PWRONRSTn. Note that PWRONRSTn and RTC_PWRONRSTn may be at different voltage levels, so PWRONRSTn may require level shifting before connecting to RTC_PWRONRSTn. This connection allows full functionality of the RTC subsystem without the internal RTC LDO consuming power.

If your application uses the RTC functionality and requires RTC-only mode, the internal LDO is required to enable proper wakeup signaling from the RTC domain. The proper wakeup signaling requires the following connections:

- RTC_KALDO_ENn is grounded
- CAP_VDD_RTC is connected to 1uF decoupling capacitor to ground
- RTC_PWRONRSTn is connected to 1.8V RTC power on reset
- PMIC_POWER_EN is connected to power input of PMIC
- RTC_WAKEUP (EXT_WAKEUP0) is connected to a wakeup source

See the device datasheet for more information on these signals.

6.4.3.6 Supported Low Power USB Wakeup Scenarios

Table 6-14 summarizes different USB wakeup use cases which are supported in each system sleep state (DeepSleep or Standby). Three use case scenarios exist:

- USB Connect: Wakeup is cause by physically inserting the USB cable.
- USB Disconnect: Wakeup is caused by physically removing the USB cable.
- USB Suspend/Resume: Wakeup is caused by a USB suspend or resume command. For example, a USB mouse click can cause a USB resume command.

Within each wakeup use case, each row describes whether or not that type of wakeup is supported in each system sleep mode. USB mode (host or device) is also considered.

There are two possible Wakeup events that are generated:

- PHY WKUP: this is an internal wakeup signal to the wakeup processor that is generated by the USB PHY based off of USB signaling.
- VBUS2GPIO: this is an external wakeup signal coming from a level change on VBUS voltage. This event requires an external board solution which routes VBUS to a GPIO on the device. Ensure you level shift the voltage to conform to the I/O requirements. When VBUS transitions from 0V to 5V (or vice versa), the transition on a GPIO will trigger a wakeup.

Table 6-14. USB Wakeup Use Cases Supported in System Sleep States

No.	USB Wakeup Use Case	System Sleep State	USB Controller State	USB Mode	Supported	USB Wakeup Event
1	USB Connect	DS	POWER OFF	Host	No	N/A
2		DS	POWER OFF	Device	Yes	VBUS2GPIO
3		Standby	Clock Gated	Host	Yes	PHY WKUP
4		Standby	Clock Gated	Device	Yes	VBUS2GPIO
5	USB Suspend / Resume	DS	POWER OFF	Host	No	N/A
6		DS	POWER OFF	Device	No	N/A
7		Standby	Clock Gated	Host	Yes	PHY WKUP
8		Standby	Clock Gated	Device	Yes	PHY WKUP
9	USB Disconnect	DS	POWER OFF	Host	No	N/A
10		DS	POWER OFF	Device	No	N/A
11		Standby	Clock Gated	Host	Yes	PHY WKUP
12		Standby	Clock Gated	Device	Yes	VBUS2GPIO

6.4.4 Main Oscillator Control During DeepSleep

The DeepSleep oscillator circuit is used to control the main oscillator by disabling it during deep sleep and enabling during active/wakeup. By default during reset, the oscillator is enabled and the oscillator control circuit comes up disabled (in-active).In order to activate the oscillator control circuit for deepsleep, DSEN bit of CTRL_DEEPSLEEP register must set. Once this is set and whenever the wakeup processor enters standby, the oscillator control will disable the oscillator causing the clock to be shut OFF. Any async event from the wakeup sources will cause the oscillator control to re-enable the oscillator after a period of DSCOUNT configured in CTRL_DEEPSLEEP register.

For use cases where an external oscillator is used, the CLKREQ pin provides a similar function to shut off the oscillator. The CLKREQ will be active high. Typically, polarity control is expected at the destination device. In the absence of polarity control and polarity mismatch, an inverter has to be put on the board.

The wakeup processor will always be powered up. But when the main oscillator is in power down / OFF state, the wakeup processor will not receive any clock.

6.4.5 *Wakeup Sources/Events*

Following are the wake sources when the main oscillator is in OFF state. These are part of the Wakeup Power domain and remain always ON.

- DMTimer0
- DMTimer_1ms (timer-based wakeup)
- ADC0: Touchscreen Controller (TSC, ADC monitor functions)
- UART0 (Infra-red support)
- I2C0
- RTCSS
- IOs via daisy chaining
- Debug
- Warm reset pin

6.4.6 *Functional Sequencing for Power Management with Wakeup Processor*

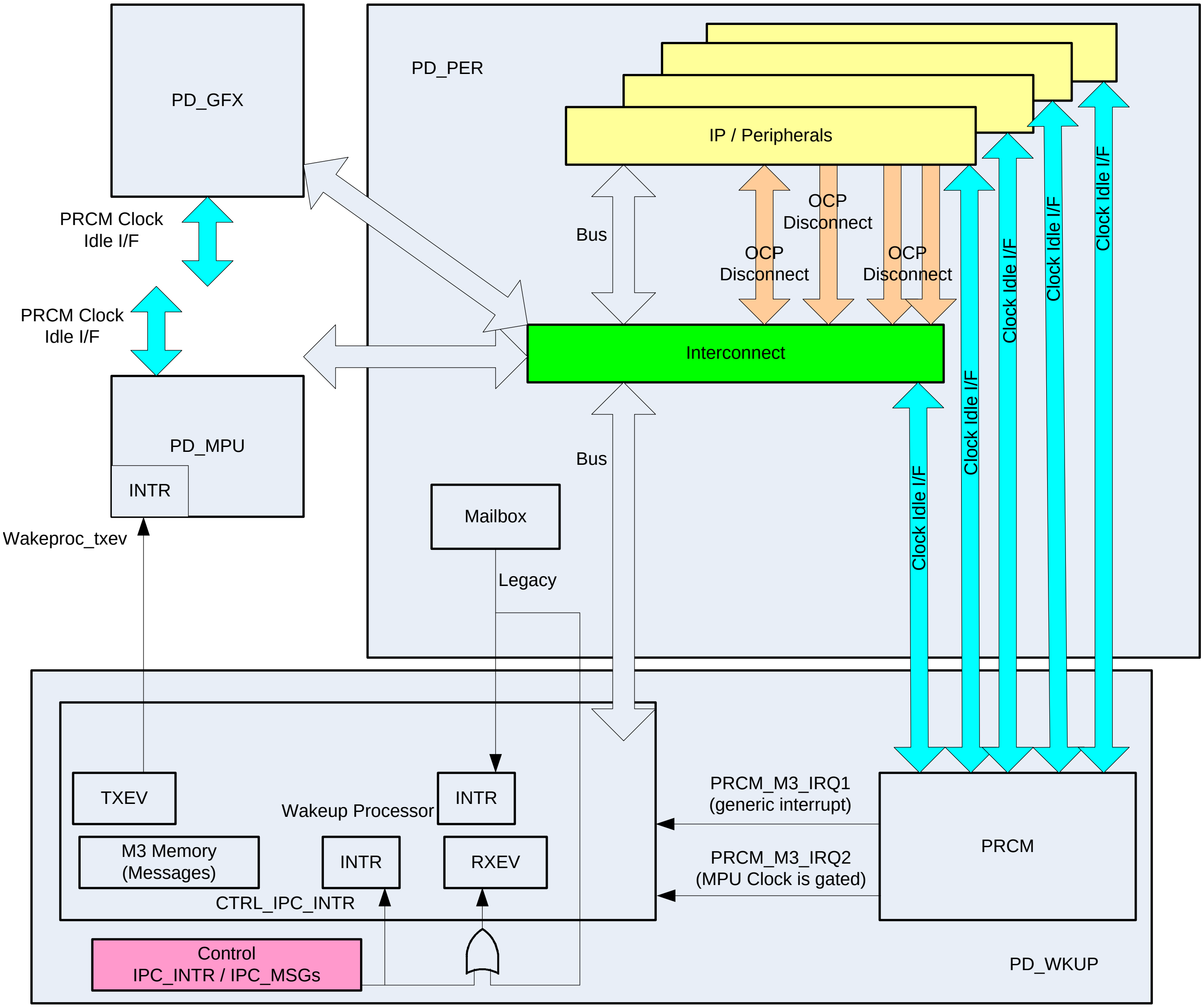
The device contains a dedicated wakeup processor to handle the power management transitions. It is part of the Wake up Power domain (PD_WKUP).

The power management sequence kicks off with the MPU executing a WFI instruction with the following steps:

1. During Active power mode, the MPU executes a WFI instruction to enter IDLE mode.
2. The wakeup processor receives an interrupt and becomes active.
3. The processor then powers down the MPU power domain (if required).
4. Interrupt registers are configured for the wakeup source.
5. The wakeup processor executes WFI and goes into idle state.
6. The wakeup event triggers an interrupt to the wakeup processor and it wakes up the MPU.

Generally, the MPU and wakeup processor are not expected to be active at the same time. The wakeup processor along with the PRCM is the power manager primarily for PD_MPU and PD_PER. Other power domains (such as PD_GFX) may be handled directly using the MPU software. Figure 6-6 gives a system level view of the Power management system between the MPU and wakeup processor.

Figure 6-6. DeepSleep System View

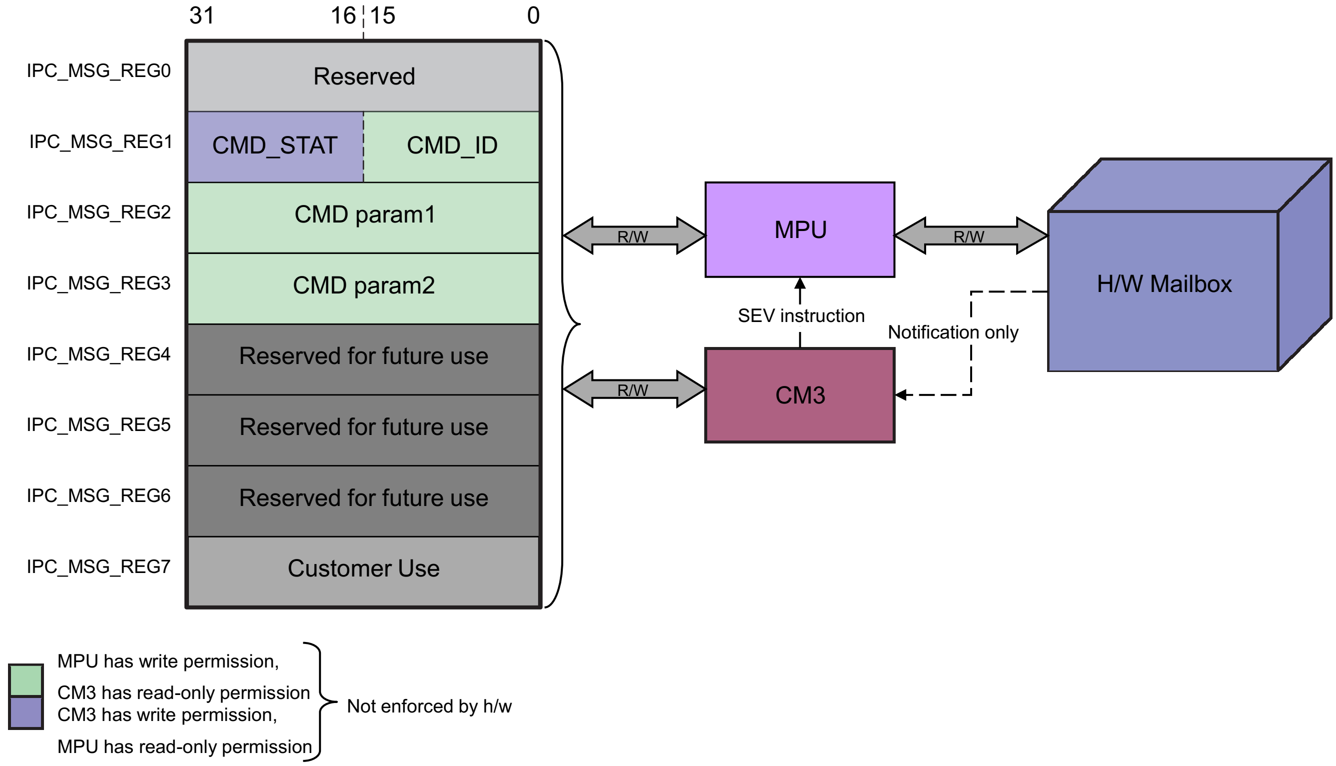


As Figure 6-6 shows, a graceful shutdown of power domains PD_PER and PD_MPU involves:

1. Software to shutdown the peripheral
2. PRCM power idle interface with each peripheral
3. OCP disconnect interface between peripherals
4. Wakeup processor to shut down the Interconnect infrastructure
5. Interprocessor communication (IPC) between MPU and wakeup processor

The wakeup processor handles all of the low-level power management control of the device. A firmware binary is provided by Texas Instruments that includes all of the necessary functions to achieve low power modes. Inter-Processor Communication (IPC) registers (ipc_msg_reg*x*, located in the control module registers) are available to communicate with the wakeup processor so the user can provide certain configuration parameters based on the level of low power that is required. Figure 6-7 provides a mapping of these registers.

Figure 6-7. IPC Mechanism



IPC_MSG_REG1 contains the CMD_STAT and CMD_ID parameters as described in Table 6-15 and Table 6-16.

Table 6-15. CMD_STAT Field

CMD_STAT	Value	Description
PASS	0x1	In the initialization phase, PASS (0x1) denotes that the CM3 was successfully initialized.
IN_PROGRESS	0x2	Early indication of command being carried out.
FAIL	0x3	In the initialization phase, 0x2 denotes CM3 could not properly initialize. When other tasks are to be done, FAIL (0x3) indicates some error in carrying out the task. Check trace vector for details.
WAIT4OK	0x4	CM3 INTC will catch the next WFI of A9 and continue with the pre-defined sequence.

Table 6-16. CMD_ID Field

CMD_ID	Value	Description
CMD_RTC	0x1	1. Initiates force_sleep on interconnect clocks. 2. Turns off MPU and PER power domains. 3. Programs the RTC alarm register for deasserting pmic_pwr_enable.
CMD_RTC_FAST	0x2	Programs the RTC alarm register for deasserting pmic_pwr_enable.
CMD_DS0	0x3	1. Initiates force_sleep on interconnect clocks. 2. Turns off the MPU and PER power domains. 3. Configures the system for disabling MOSC when CM3 executes WFI.
CMD_DS1	0x5	1. Initiates force_sleep on interconnect clocks. 2. Turns off the MPU power domains. 3. Configures the system for disabling MOSC when CM3 executes WFI.

6.4.6.1 Sleep Sequencing

This section gives the system level guidelines for sleep sequencing. The guidelines can serve as an example for implementing the sleep mode sequencing. The user can opt to implement a sequence with certain steps interchanged between the MPU and the wakeup processor.

1. Application saves context of peripherals to memories supporting retention and DDR – this step is only required for DeepSleep.
2. MPU OCMC_RAM goes into retention.
3. Unused power domains are turned OFF: program clock/power domains PWRSTCTRL, save contexts, and so forth.
4. Software populates L3_OCMC_RAM for wakeup restoration viz Save EMIF settings, public restoration pointers, and so forth.
5. Execute WFI from SRAM.
6. Any peripheral interrupt will trigger a wake interrupt to the wakeup processor through the MPU's WKUP signal, INTR2.
7. After the MPU power domain is clock-gated, the PRCM provides an interrupt to the wakeup processor using INTR1.
8. The wakeup processor performs low-level power sequencing to turn off certain power domain and eventually executes WFI.
9. When the wakeup processor goes into WFI, the hardware oscillator control circuit disables the master oscillator.

6.4.6.2 Wakeup Sequencing

This section gives the guidelines for Wakeup sequencing.

1. One of the wakeup event triggers (which was configured during the sleep sequencing) initiates a wakeup sequence
2. The wakeup event switches ON the oscillator, which was configured to go OFF during DeepSleep.
3. The wakeup event also triggers an interrupt to the wakeup processor.
4. On the wakeup event interrupt, the wakeup processor executes the following:
   - Restore the voltages to normal Operating voltage.
   - Enable PLL locking.
   - Switch ON the power domains and/or enable clocks for PD_PER.
   - Switch ON the power domains and/or enable clocks for PD_MPU.
   - Execute WFI.
5. The MPU executes from ROM reset vector.
6. Restore the application context (required only for Deep sleep 0).

6.4.7 I/O Power Management and Daisy Chaining

I/O power management is useful when the main oscillator is clock gated and the PER power domain is OFF (power-gated state). Since the I/Os are controlled by modules in power-gated state, I/O power management is required to have flexibility when interfacing with external devices. During DeepSleep, the wakeup feature is active, described in **Wakeup** below. Isolation is required to be activated and de-activated during sleep and wakeup sequencing, respectively (see following description of **Isolation**).

Three aspects comprise I/O cell power management.

Isolation— Isolation from power state transitions.

When ISOLATED, the I/Os will hold their previous state (0,1, tristate) until the ISOLATION is released. The controls for the ISOLATION travel through the chain. Isolation is controlled during sleep and wakeup sequencing.

Note: The PRCM must have the optional feature to remove isolation from DDR I/Os while allowing rest of the I/O isolation to be removed later. This allows for software to restore any critical peripherals from DDR and ensure the peripheral is in a required or valid state (like GPIO) before the remaining I/Os are removed from Isolation.

Wakeup— I/O PAD can be individually enabled for wakeup using PADCONFx.WUEN register bit.

The wakeup controls and events (as well as enable and disable controls) travels through asynchronously through the chain during DeepSleep. The global wakeup chain is enabled/controlled by the PRCM. This is global control is achieved by qualifying each of the I/O PAD wakeup enables with global wakeup daisy chain control coming from PRCM.

PADCONFIG— Individual control on what will be the IO state during Isolation using DS-PADCONFIG control register bits.

The DS-PADCONFIG registers must be used only if the default state of the I/O during the low power state does not meet system requirements.

The following terminal signal names are special, and are excluded from the isolation and wakeup daisy chains. Control of these signals can remain with GPIO0 even when I/O isolation is enabled. Refer to the device data sheet to determine which pins map to each signal name. Each pin has a corresponding GPIO that can be mapped to one of the GPIOs in the GPIO0 module.

- System signals: PWRONRSTn, WARMRSTn, xdma_event_intr0, xdma_event_intr1, clkreq
- JTAG interface: TMS, TDI, TCK, nTRST, EMU0, EMU1
- I2C0: I2C0_SDA, I2C0_SCL
- spi0_cs1 and eCAP0_in_PWM0_out

6.5 PRCM Module Overview

The PRCM is structured using the architectural concepts presented in the 5000x Power Management Framework. This framework provides:

A set of modular, re-usable FSM blocks to be assembled into the full clock and power management mechanism. A register set and associated programming model. Functional sub-block definitions for clock management, power management, system clock source generation, and master clock generation.

The device supports an enhanced power management scheme based on four functional power domains:

Generic Domains

- WAKEUP
- MPU
- PER
- RTC

The PRCM provides the following functional features:

- Software configurable for direct, automatic, or a combination thereof, functional power domain state transition control
- Device power-up sequence control
- Device sleep / wake-up sequence control
- Centralized reset generation and management
- Centralized clock generation and management

The PRCM modules implement these general functional interfaces:

- OCP configuration ports
- Direct interface to device boundary
- Power switch control signals
- Device control signals
- Clocks control signals

- Resets signals
- A set of power management protocol signals for each module to control and monitor standby, idle and wake-up modes (CM and PRM)
- Emulation signals

6.5.1 Interface Descriptions

This section lists and shortly describes the different interfaces that allow PRCM to communicate with other modules or external devices.

6.5.1.1 OCP Interfaces

The PRCM has 1 target OCP interfaces, compliant with respect to the OCP/IP2 standard. The OCP port, for the PRCM module is used to control power, reset and wake-up Management.

6.5.1.2 OCP Slave Interfaces

PRCM implements a 32-bit OCP target interface compliant to the OCP/IP2.0 standard.

6.5.1.3 Power Control Interface

The Device does has power domain switches over the device, this interface provides PRCM control over power domain switches and receives responses from the power domains which indicate the switch status. It also controls the isolation signals. The control for power domain switches will be latched in PRCM Status Registers

6.5.1.4 Device Control Interface

This interface provides PRM management of several device-level features which are not specific to any single power domain. This PRM interface controls signals to/from the device for global control:

- Device Type coding
- IOs isolation control

6.5.1.5 Clocks Interface

This interface gathers all clock inputs and outputs managed by PRCM modules.

6.5.1.6 Resets Interface

This interface gathers all resets inputs and outputs managed by PRCM module.

6.5.1.7 Modules Power Management Control Interface

Modules or subsystems in the device are split over 2 categories:

- Initiator: an initiator is a module able to generate traffic on the device interconnects (typically: processors, MMU, EDMA).
- Target: a target is a module that cannot generate traffic on the device interconnects, but that can generate interrupts or DMA request to the system (typically: peripherals). PRCM handles a power management handshake protocol with each module or sub-system. This protocol allows performing proper clock and power transition taking into account each module activity or state.

6.5.1.8 Initiator Modules Interface

PRCM module handle all initiator modules power management interfaces: MStandby signal MWait signal

6.5.1.9 Targets Modules Interface

PRCM module handle all target modules power management interfaces: SIdleReq signal SIdleAck signal FCLKEN signal

Note: USB Support for SWakeUp

6.6 Clock Generation and Management

PRCM provides a centralized control for the generation, distribution and gating of most clocks in the device. PRCM gathers external clocks and internally generated clocks for distribution to the other modules in the device. PRCM manages the system clock generation

6.6.1 Terminology

The PRCM produces 2 types of clock:

Interface clocks: these clocks primarily provide clocking for the system interconnect modules and the portions of device's functional modules which interface to the system interconnect modules. In most cases, the interface clock supplies the functional module's system interconnect interface and registers. For some modules, interface clock is also used as functional clock. In this document, interface clocks are represented by blue lines.

Functional clock: this clock supplies the functional part of a module or a sub-system. In some cases, a module or a subsystem may require several functional clocks: 1 or several main functional clock(s), 1 or several optional clock(s). A module needs its main clock(s) to be operational. Optional clocks are used for specific features and can be shutdown without stopping the module

6.6.2 Clock Structure

To generate high-frequency clocks, the device supports multiple on-chip DPLLs controlled directly by the

PRCM module. They are of two types of PLLs, referred to ADPLLS and ADPLLLJ throughout this document.

The ADPLLS module is used for the Core, Display, ARM Subsystem and DDR PLLs

The ADPLLLJ module is used for the peripheral functional clocks

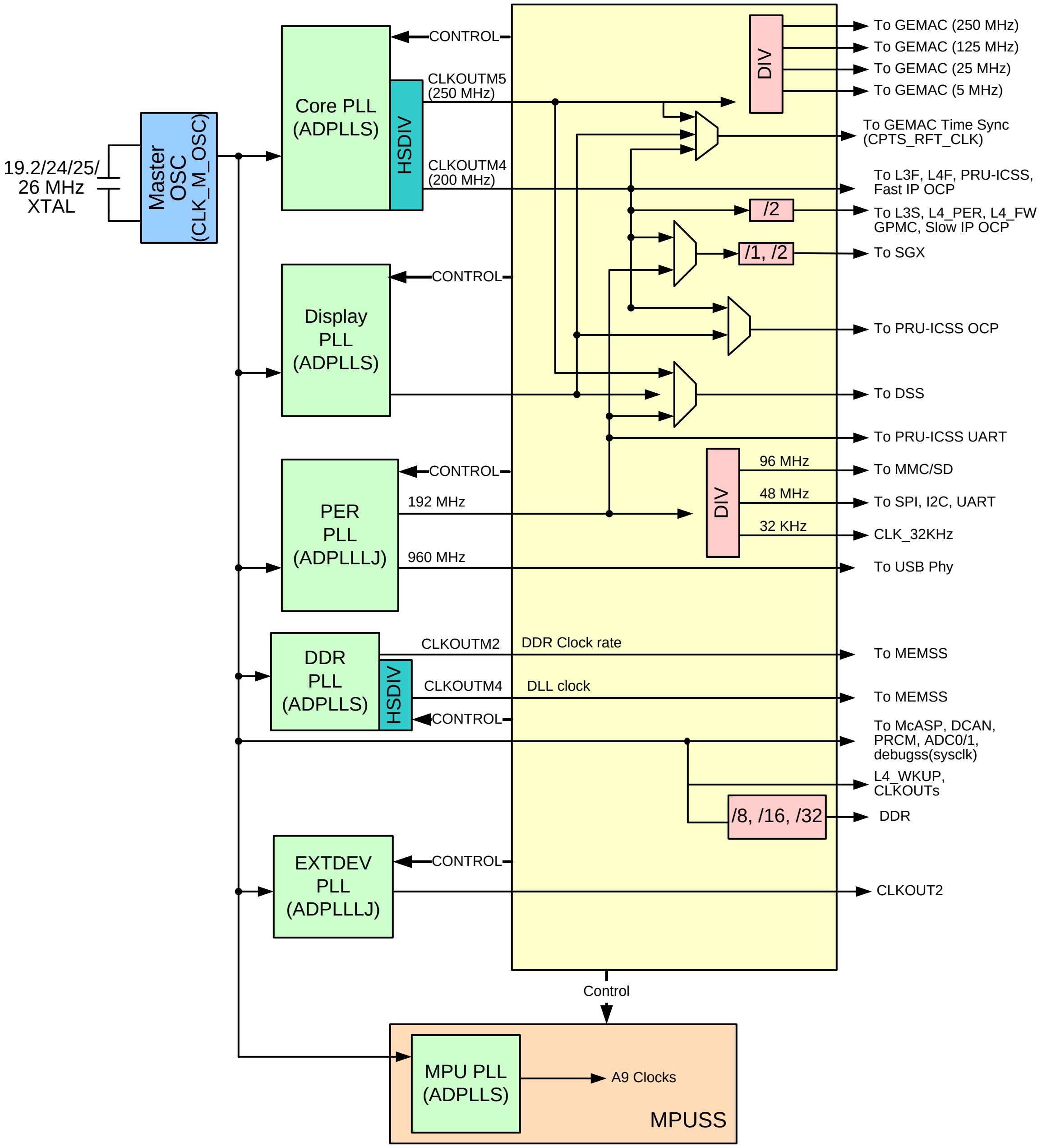
The device has two reference clocks which are generated by on-chip oscillators or externally. These are for the main clock tree and RTC block, respectively.

In the case of an external oscillator, a clock can directly be connected to XTALIN pin and the oscillator will be put in bypass mode. The 32-Khz crystal oscillator is controlled and configurable by the RTC. This device also contains an on-chip RC oscillator. This oscillator is not configurable and is always on.

The main oscillator on the device (see Chapter 5, *Initialization*, for possible frequencies) produces the master high frequency clock CLK_M_OSC.

Figure 6-8 shows a high-level overview of the device clock architecture. The diagram shows only the peripheral functional clocks, and the frequency values shown are typical (OPP100) values. See each DPLL section for a more detailed clocking structure.

Figure 6-8. Internal Clocking Architecture



6.6.3 ADPLLS

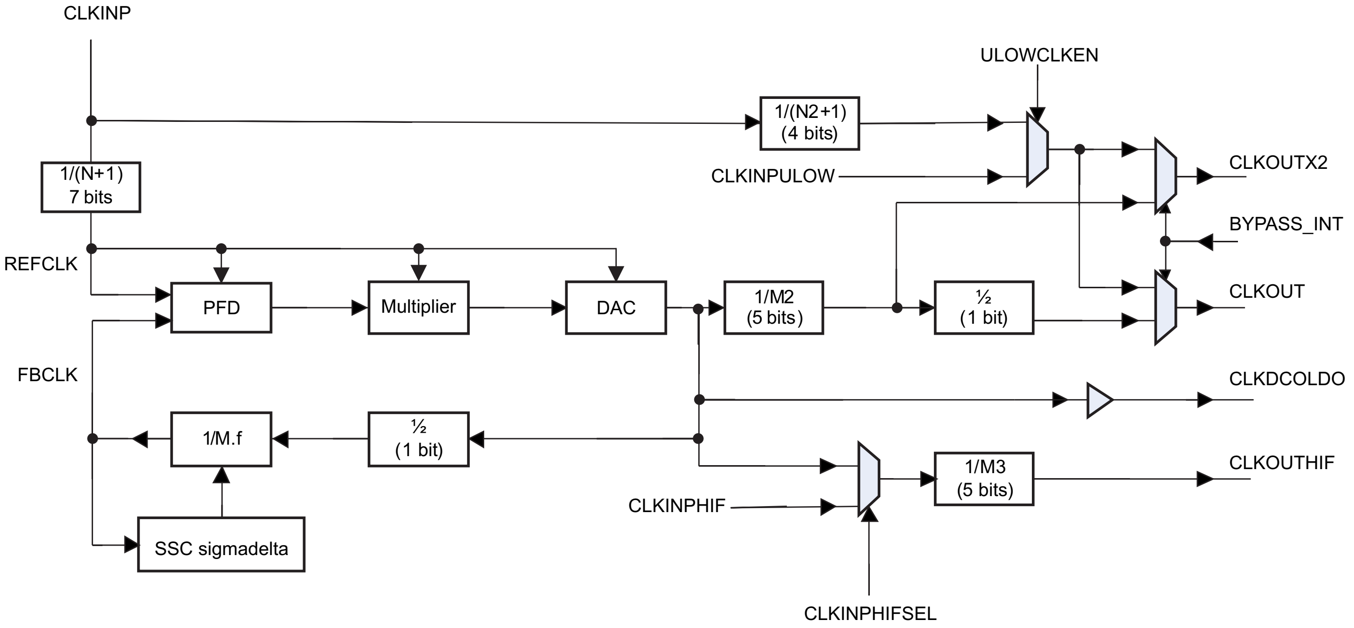
The ADPLLS is a high resolution frequency synthesizer PLL with built in level shifters which allows the generation of PLL locked frequencies up to 2 GHz. ADPLLS has a predivide feature which allows user to divide, for instance, a 24- or 26-MHz reference clock to 1-MHz and then multiply up to 2-GHz maximum.

All PLLs will come-up in bypass mode at reset. SW needs to program all the PLL settings appropriately and then wait for PLL to be locked. For more details, see the configuration procedure for each PLL.

The following PLLs are:

- MPU PLL
- Core PLL
- Display PLL
- DDR PLL

Figure 6-9. ADPLLS



The ADPLLS has three input clocks:

- CLKINP: Reference input clock
- CLKINPULOW: Low frequency input clock for bypass mode only.
- CLKINPHIF: High Frequency Input Clock for post-divider M3

The ADPLLS has four output clocks:

- CLKOUTHIF: High Frequency Output Clock from Post divider M3
- CLKOUTX2: Secondary 2x Output
- CLKOUT: Primary output clock
- CLKDCOLDO: Oscillator (DCO) output clock with no bypass

The DPLL has two internal clocks:

- REFCLK (Internal reference clock): This is generated by dividing the input clock CLKINP by the programmed value N+1. The entire loop of the PLL runs on the REFCLK.

  Here, REFCLK = CLKINP/(N+1).
- BCLK: Bus clock which is used for programming the various settings using registers

The ADPLLS lock frequency is defined as follows: f_{DPLL} = CLKDCOLDO

6.6.3.1 Clock Functions

Table 6-17. Output Clocks in Locked Condition

Pin Name	Frequency	Comments
	REGM4XEN='0'	
CLKOUT	[M / (N+1)] * CLKINP * [1/M2]	
CLKOUTX2	2 * [M / (N+1)] * CLKINP * [1/M2]	
CLKDCOLDO	2 * [M / (N+1)] * CLKINP	
CLKOUTHIF	CLKINPHIF / M3	CLKINPHIFSEL='1'
	2 * [M / (N+1)] * CLKINP * [1/M3]	CLKINPHIFSEL='0'
	REGM4XEN='1'	
CLKOUT	[4M / (N+1)] * CLKINP * [1/M2]	
CLKOUTX2	2 * [4M / (N+1)] * CLKINP * [1/M2]	
CLKDCOLDO	2 * [4M / (N+1)] * CLKINP	
CLKOUTHIF	CLKINPHIF / M3	CLKINPHIFSEL='1'
	2 * [4M / (N+1)] * CLKINP * [1/M3]	CLKINPHIFSEL='0'

Table 6-18. Output Clocks Before Lock and During Relock Modes

Pin Name	Frequency	Comments
CLKOUT	CLKINP / (N2+1)	ULOWCLKEN='0'
	CLKINPULOW	ULOWCLKEN='1'
CLKOUTX2	CLKINP / (N2+1)	ULOWCLKEN='0'
	CLKINPULOW	ULOWCLKEN='1'
CLKDCOLDO	Low	
CLKOUTHIF	CLKINPHIF/M3	ULOWCLKEN='1'
	Low	ULOWCLKEN='0'

Note: Since M3 divider is running on the internal LDO domain, in the case when CLKINPHIFSEL=’1’, CLKOUTHIF could be active only when internal LDO is ON. Hence, whenever LDOPWDN goes low to high to powerdown LDO (happens when TINITZ activated / when entering slow relock bypass mode), output CLKOUTHIF will glitch and stop. To avoid this glitch, it is recommended to gate CLKOUTHIF using control CLKOUTHIFEN before asserting TINITZ / entering any slow relock bypass mode Frequency Range (MHz)

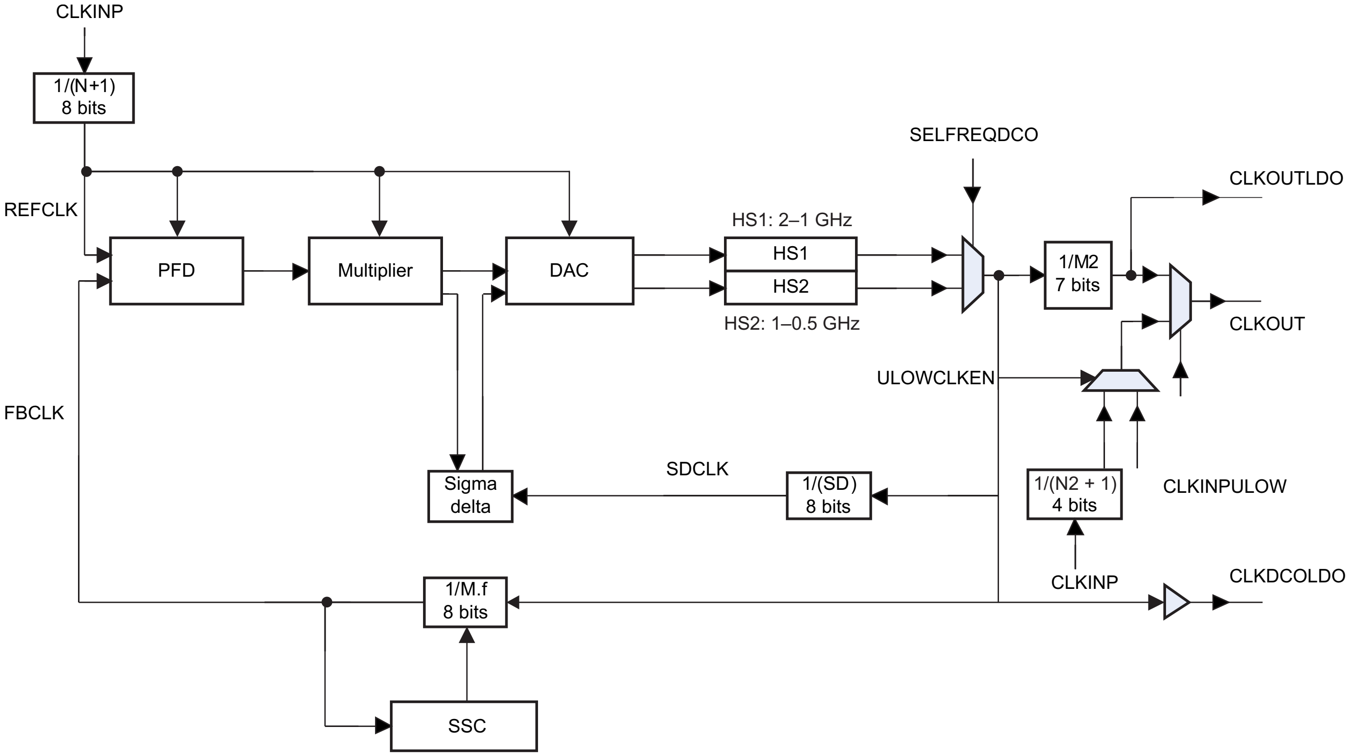
See the device-specific data manual for details on operating performance points (OPPs) supported by your device.

6.6.4 ADPLLLJ (Low Jitter DPLL)

The ADPLLLJ is a low jitter PLL with a 2-GHz maximum output. ADPLLLJ has a predivide feature which allows user to divide, for instance, a 24-MHz or 26-MHz reference clock to 1 MHz and then multiply up to 2 GHz maximum.

All PLLs will come-up in bypass mode at reset. SW needs to program all the PLL settings appropriately and then wait for PLL to be locked. For more details, see the configuration procedure for each PLL.

Figure 6-10. Basic Structure of the ADPLLLJ



The peripheral PLL and EXTDEV PLL belong to type ADPLLLJ:

The DPLL has two input clocks:

- CLKINP: Reference input clock
- CLKINPULOW: Bypass input clock.

The DPLL has two internal clocks:

- REFCLK (Internal reference clock): This is generated by dividing the input clock CLKINP by the programmed value N+1. The entire loop of the PLL runs on the REFCLK.

  Here, REFCLK = CLKINP/(N+1).
- CLKDCO (Internal Oscillator clock.):This is the raw clock directly out of the digitally controlled oscillator (DCO) before the post-divider. The PLL output clock is synthesized by an internal oscillator which is phase locked to the refclk. There are two oscillators built within ADPLLLJ. The oscillators are user selectable based on the synthesized output clock frequency requirement. In locked condition, CLKDCO = CLKINP *[M/(N+1)].

The ADPLLLJ lock frequency is defined as follows: f_{DPLL} = CLKDCOOUT

The DPLL has three external output clocks:

- CLKOUTLDO: Primary output clock in VDDLDOOUT domain. Bypass option not available on this output.

  CLKOUTLDO = (M / (N+1))*CLKINP*(1/M2)
- CLKOUT:

  Primary output clock on digital core domain

  CLKOUT = (M / (N+1))*CLKINP*(1/M2)
- CLKDCOLDO:

  Oscillator (DCO) output clock before post-division in VDDLDOOUT domain. Bypass option is not available on this output.

  CLKDCOLDO = (M / (N+1))*CLKINP.

All clock outputs of the DPLL can be gated. The Control module provides the DPLL with a clock gating control signal to enable or disable the clock, and the DPLL provides the PRCM module with a clock activity status signal to let the PRCM module hardware know when the clock is effectively running or effectively gated. Output clock gating control for various clockouts: CLKOUTEN/CLKOUTLDOEN/CLKDCOLDOEN.

6.6.4.1 Clock Functions

Table 6-19. Output Clocks in Locked Condition

Pin Name	Frequency
CLKOUT	[M /(N+1)] * CLKINP * [1/M2]
CLKOUTLDO	[M /(N+1)] * CLKINP * [1/M2]
CLKDCOOUT	[M /(N+1)] * CLKINP

Table 6-20. Output Clocks Before Lock and During Relock Modes

Pin Name	Frequency	Comments
CLKOUT	CLKINP/(N2+1)	ULOWCLKEN=’0’
	CLKINPLOW	ULOWCLKEN=’1’
CLKDCOLDO	LOW	
CLKOUTLDO	LOW	

6.6.5 M2 and N2 Change On-the-Fly

The dividers M2 and N2 are designed to change on the fly and provide a glitch-free frequency switch from the old to new frequencies. In other words, they can be changed while the PLL is in a locked condition, without having to switch to bypass mode. A status toggle bit will give an indication if the new divisor was accepted. These dividers can also be changed in bypass mode, and the new divisor value will be reflected on output after the PLL relocks. For more details, see the PLL configuration procedures for each PLL.

6.6.6 Spread Spectrum Clocking (SSC)

NOTE: Spread spectrum clock is only supported for the Display and MPU PLLs on this device. Spread spectrum clocking is not supported for DDR, PER, DISP, and CORE and EXTDEV PLLs. When enabling SSC on MPU PLL, ensure the maximum MPU frequency remains below the maximum rated frequency for the chosen OPP (see the device-specific Data Manual for more details).

The module supports spread spectrum clocking (SSC) on its output clocks. SSC is used to spread the spectral peaking of the clock to reduce any electromagnetic interference (EMI) that may be caused due to the clock’s fundamental or any of its harmonics. When SSC is enabled the clock’s spectrum is spread by the amount of frequency spread, and the attenuation is given by the ratio of the frequency spread (Δf) and the modulation frequency (f_m), i.e., [{10*log10(Df/f_m)}-10] dB.

6.6.6.1 Definition

The aim of SSC is to add a variation in the frequency of an original clock, which spreads the generated interferences over a larger band of frequency.

In theory, SSC means that the clock signal is varied around the desired frequency. For example, for a 1-GHz clock, the frequency may be 999.5 MHz at one moment and 1.0005 GHz at another. When SSC is enabled the clock spectrum is spread by the amount of frequency spread. Doing this constantly causes the power of the tone to be spread out more over a broader band of tight frequencies (centered at the desired tone). To realize this constant variation on the original signal, a modulation with an additional signal (called spreading waveform) is realized.

Creating an SSC by spreading the initial clock frequency is done by defining the following parameters:

- The spreading frequency (deviation), which is the ratio of the range of spreading frequency over the original clock frequency
- The modulation rate (f_m), which is used to determine the clock-frequency spreading-cycling rate and is the time during which the generated clock frequency varies through Δf and returns to the original frequency
- The modulation waveform, which describes the variation curve in terms of time

The spectral power reduction in the DPLL clocks is dependent on the modulation index (K), which is a ratio of spreading frequency calculated from the frequency deviation (Δf) and the modulation rate (f_m) .

6.6.6.2 Effect on the Clock Signal

Figure 6-11 is an example of the effect of a triangular spreading on a clock signal.

Figure 6-11. Effect of the SSC in Frequency

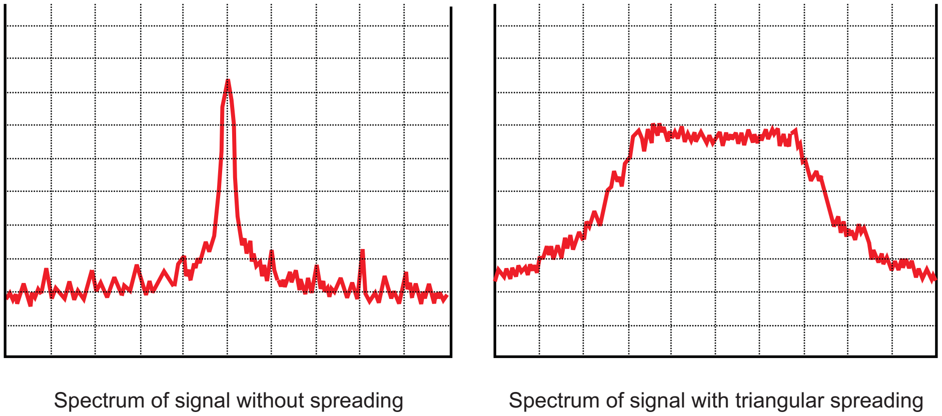
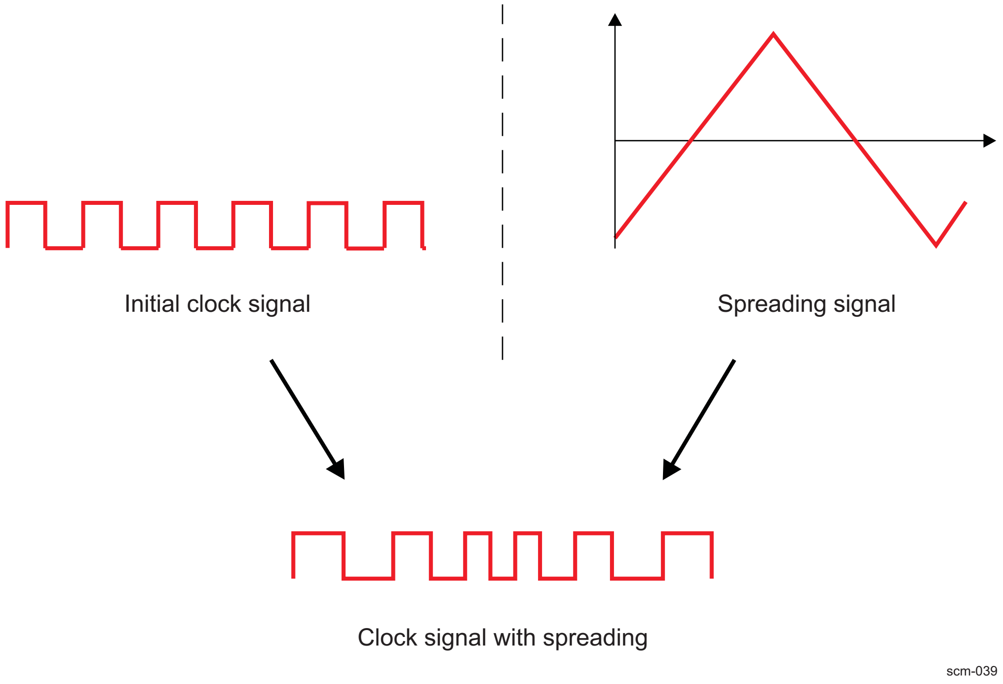


Figure 6-11 shows not only the power reduction of the main peak, but also the flatter aspect of the modulated signal. The minimum level of the second signal is higher than the minimum level of the first signal. This effect is normal and is due to the noise added for the modulation.

NOTE: The spreading technique scatters the energy of the peaks on the other frequencies, which reduces the power of the peaks but increases the global noise of the signal.

Figure 6-12 shows the effect of triangular spreading on a clock signal in the time domain.

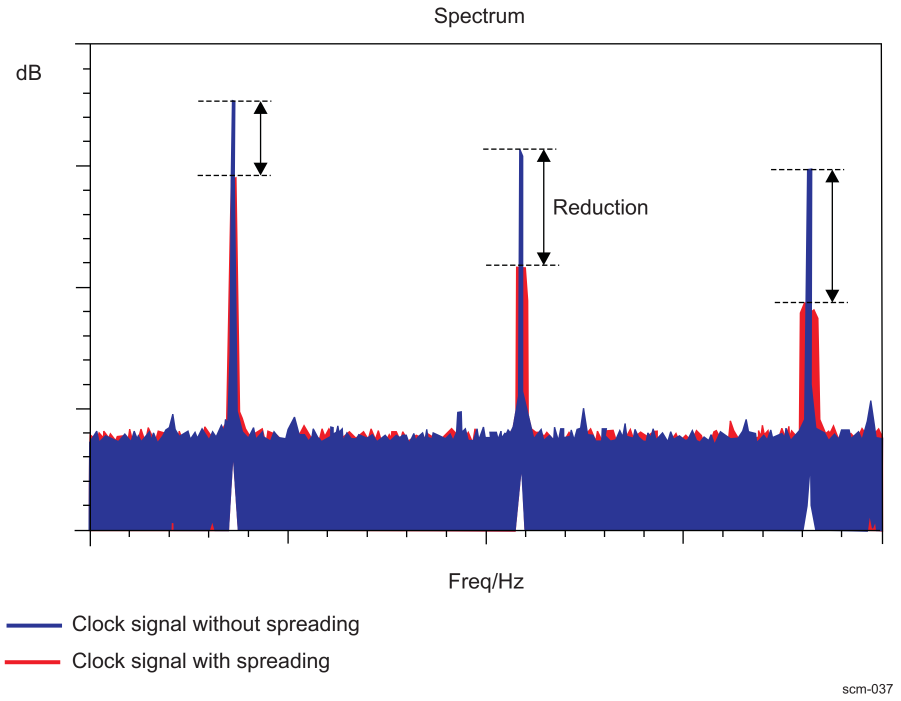
Figure 6-12. Effect of the SSC in the Time Domain



6.6.6.3 Estimation of the EMI Reduction Level

Figure 6-13 shows the effect of spreading on a clock and its harmonics.

Figure 6-13. Peak Reduction Caused by Spreading



The electromagnetic interference reduction can be estimated with the following equation:

Peak_power_reduction = 10 * log ((Deviation * f_c) / f_m

With:

- Peak_power_reduction in dB
- Deviation in % of the initial clock frequency (f_c), equals Δf / f_c
- f_c is the original clock frequency, in MHz
- f_m is the spreading frequency, in MHz

According to equation (1), it is also possible to compute the deviation, and then Δf, for a required peak power reduction:

Deviation = $(f_m / f_c) * 10^{(\text{Peak\_power\_reduction} / 10)}$

Example:

For f_c=400 MHz, deviation =1% peak from f_c(Δf = 4MHz) and f_m=400kHz; the estimated peak power reduction is 10dB.

6.6.6.4 Bandwidth Calculation (Carson Bandwidth Rule)

The Carson bandwidth rule defines the approximate bandwidth requirements of communications system components for a carrier signal that is frequency-modulated by a continuous or broad spectrum of frequencies rather than a single frequency.

The Carson bandwidth rule is expressed by the relation CBR = 2 * (Δf + f_m), where CBR is the bandwidth requirement, Δf is the peak frequency deviation, and f_m is the highest frequency in the modulating signal.

For example, an FM signal with a 5-kHz peak deviation and a maximum audio frequency of 3 kHz, would require an approximate bandwidth 2*(5 3) = 16 kHz.

Theoretically, any FM signal has an infinite number of sidebands and hence an infinite bandwidth, but in practice all significant sideband energy (98% or more) is concentrated within the bandwidth defined by the Carson bandwidth rule.

6.6.6.5 SSC Generation Control in the Device

SSC is performed by changing the feedback divider (M) in a triangular pattern. Implying, the frequency of the output clock would vary in a triangular pattern. The frequency of this pattern would be modulation frequency (f_m). The peak (ΔM) or the amplitude of the triangular pattern as a percent of M would be equal to the percent of the output frequency spread (Δf); that is, $\Delta M/M = \Delta f / f_c$. Next mark with Finp the frequency of the clock signal at the input of the DPLL. Because it is divided to N+1 before entering the phase detector, so the internal reference frequency is Fref = Finp / (N + 1).

Assume the central frequency f_c to be equal to the DPLL output frequency Fout, or f_c= Fout = (Finp / (N + 1)) * (M / M2). Since this is in band modulation for the DPLL, the modulation frequency is required to be within the DPLL's loop bandwidth (lowest BW of Fref / 70). A higher modulation frequency would result in lesser spreading in the output clock.

SSC can be enabled/disabled using bit CM_CLKMODE_DPLL_xxx.DPLL_SSC_EN (where xxx can be any one of the following DPLLs: MPU, DDR, DISP, CORE, PER). An acknowledge signal CM_CLKMODE_DPLL_xxx.DPLL SSC_ACK notifies the exact start and end of SSC. When SSC_EN is de-asserted, SSC is disabled only after completion of one full cycle of the triangular pattern given by the modulation frequency. This is done in order to maintain the average frequency.

Modulation frequency (f_m) can be programmed as a ratio of Fref / 4; that is, the value that needs to be programmed ModFreqDivider = Fref / (4*f_m). The ModFreqDivider is split into Mantissa and 2^Exponent (ModFreqDivider = ModFreqDividerMantissa * 2^ModFreqDividerExponent). The mantissa is controlled by 7-bit signal ModFreqDividerMantissa through CM_SSC_MODFREQDIV_DPLL_xxx.MODFREQDEV_MANTISSA bit field. The exponent is controlled by 3bit signal ModFreqDividerExponent through the CM_SSC_MODFREQDIV_DPLL_xxx.MODFREQDEV_EXPONENT bit field.

NOTE: Although the same value of ModFreqDivider can be obtained by different combinations of mantissa and exponent values, it is recommended to get the target ModFreqDivider by programming maximum mantissa and a minimum exponent.

To define the Frequency spread (Δf), ΔM must be controlled as explained previously. To define ΔM, the step size of M for each Fref during the triangular pattern must be programmed; that is,

ΔM = (2^ModFreqDividerExponent) * ModFreqDividerMantissa * DeltaMStep IF ModFreqDividerExponent ≤ 3ΔM = 8 * ModFreqDividerMantissa * DeltaMStep IF ModFreqDividerExponent > 3

DeltaMStep is split into integer part and fractional part. Integer part is controlled by 2-bit signal DeltaMStepInteger through the CM_SSC_DELTAMSTEP_DPLL_xxx.DELTAMSTEP_INTEGER bit field. Fractional part is controlled by 18-bit signal DeltaMStepFraction through the CM_SSC_DELTAMSTEP_DPLL_xxx.DELTAMSTEP_FRACTION bit field.

The frequency spread achieved has an overshoot of 20 percent or an inaccuracy of +20 percent. If the CM_CLKMODE_DPLL.DPLL_SSC_DOWNSPREADis set to 1, the frequency spread on lower side is twice the programmed value. The frequency spread on higher side is 0 (except for the overshoot as described previously).

There is restriction of range of M values. The restriction is M-ΔM should be ≥ 20. Also, M+ΔM should be ≤ 2045. In case the downspread feature is enabled, M-2*ΔM should be ≥ 20 and M ≤ 2045.

6.6.6.6 SSC Generation

The configuration of the spreading feature is not mandatory when programming the DPLL. This feature is usually enabled when the DPLL clocks generate harmonics that can potentially interfere with the GSM carrier frequencies.

Let's take the SSC featured Display ADPLL and try to set the output frequency to F_{out} = f_c = 11 MHz. Software most likely sets the DPLL higher to clock the DSS module at a higher functional clock, and then sets the DISPC_DIVISOR to achieve an 11 MHz pixel clock. But in this example, the PLL is set to output 11 MHz. The frequency of the input clock source for Display ADPLL is F_{inp} = 25 MHz.

1. The desired output frequency can be achieved with the following ratio of the divider coefficients: (M / M2) * 1 / (N + 1) = F_{out} / F_{inp} = 11 / 25. The dividers used in the Display ADPLL can be set within the following ranges: N = 0..127; M = 0..2047; M2 = 1;2. The desired output frequency is achieved through the following choice of possible divider values: M = 22; N = 4; and M2 = 10. In that case the reference clock F_{ref} = F_{inp} / (N + 1) = 25 / (4 + 1) = 5 MHz.

   The feedback divider value M = 22 is chosen to satisfy the restriction from Section 6.6.6. If, for example, the deviation ΔM / M = Δf f_c= 0.05 (5%) is chosen, we have M + ΔM < 2045 and at the same time M + ΔM > 20

   Once the clock generation control registers are configured, it is possible to configure the spreading on the clock signal.

2. Calculate the ratio between central(output) frequency and modulation frequency on the base of the desired peak power reduction (PPR) and chosen relative deviation Δf / F_{out}, where Δf / F_{out} = f_m / f_c* 10 ^ (PPR / 10). To achieve PPR = 10dB with SSC deviation (Δf / f_c) chosen to be equal to 5 percent, f_m = Δf / $10^{(PPR/10)}$ = 55 kHz. To check whether the modulation frequency has the appropriate value, check whether it is within the DPLL loop bandwidth or if f_m < F_{ref} / 70 = 5 / 70 = 71.4 KHz, which is true.

3. Calculate the contents of the MODFREQDEV_MANTISSA and MODFREQDEV_EXPONENT bit fields on the base of ModFreqDivider value: ModFreqDivider = F_{ref} / (4 * f_m) = 5 / (4 * 0.055) = 22.73. The resulting value needs to be put in the form MODFREQDEV_MANTISSA * $2^{MODFREQDEV\_EXPONENT}$. Thus, we can approximate 23 = 23 * 20. The approximation will just slightly affect the PPR.

   This means we should write MODFREQDEV_EXPONENT = 0x0 and MODFREQDEV_MANTISSA = 0x17.

4. The DeltaMStep parameter is calculated according to the formula: DeltaMStep = ΔM / ModFreqDivider.

   Since ΔM = M * (Δf / f_c), DeltaMStep = M * (Δf / f_c) / ModFreqDivider. Thus in this example, DeltaMStep = 22 * 0.05 / 23 = 0.047826.

   In this case, write 0x0 in DELTAMSTEP_INTEGER (bits 19:18). To express the fractional part 0.05 as a binary, calculate: 0.047826 * 2^18 = 12537.3, then round to 12537, convert the integer part to binary and write it into the field: DELTAMSTEP_FRACTIONAL (bits 17:0) = 0x30F9.

5. The spreading must be enabled using the SSC_EN bit.

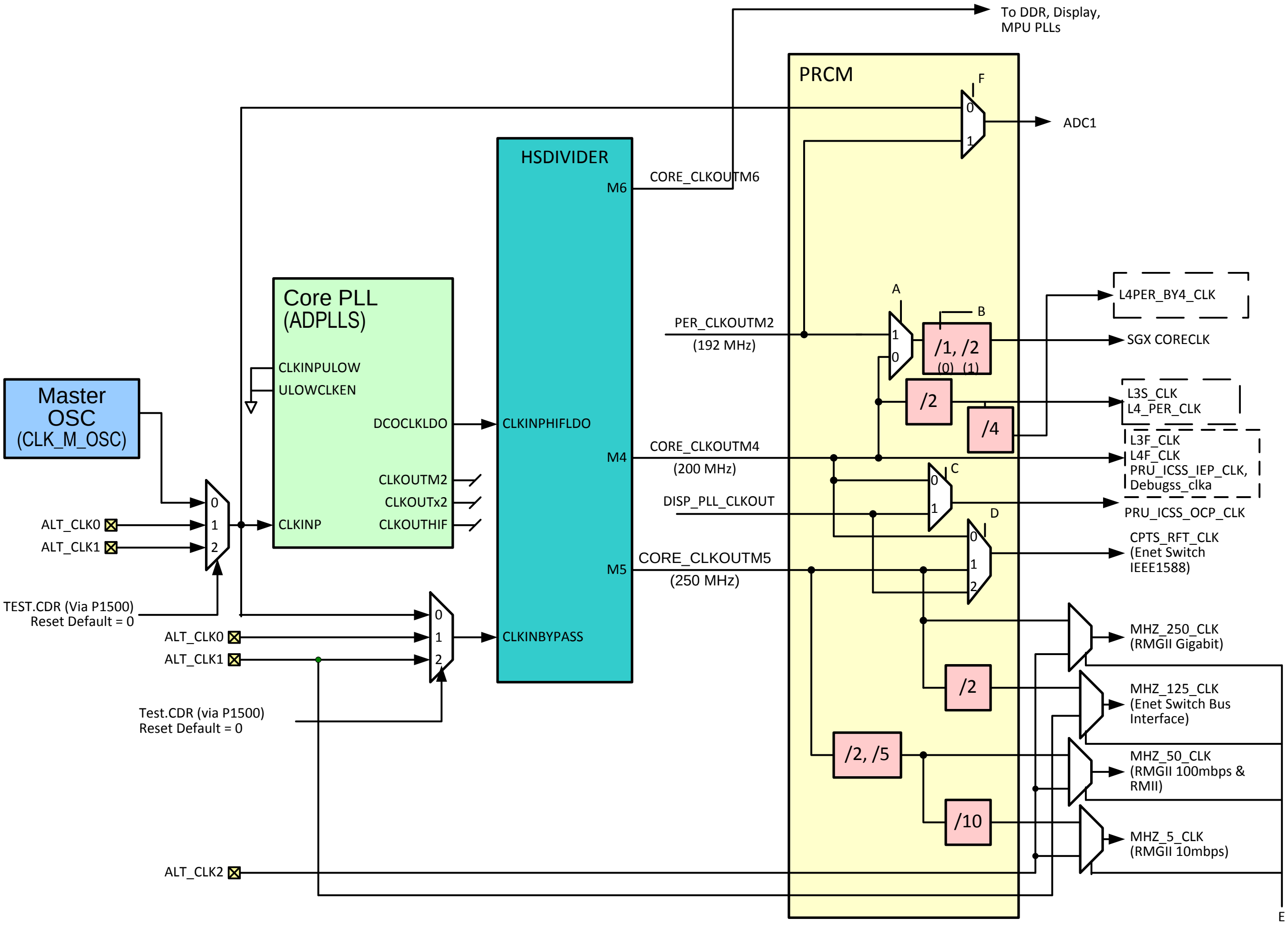
NOTE: It is necessary to configure the spreading on a clock carefully to avoid adding noise on frequencies that are used by another module. For example, adding spreading on a clock to reduce noise on GSM frequencies can "move" the generated noise to the frequency of the memory controller and degrade its performance.

The state of the modulation feature can be monitored with the DPLL_SSC_ACK bit of the corresponding register.

6.6.7 Core PLL Description

The Core PLL provides the source for a majority of the device infrastructure and peripheral clocks. The Core PLL comprises an ADPLLS with HSDIVIDER and additional dividers and muxes located in the PRCM as shown in Figure 6-14.

Figure 6-14. Core PLL Structure



ALT_CLKs are to be used for internal test purpose and should not be used in functional mode.

Table 6-21. PLL and Clock Frequences

Mux Select	Register BitSection 7.2.3.4
A	PRCM.CLKSEL_GFX_FCLK[1]
B	PRCM.CLKSEL_GFX_FCLK[0]
C	PRCM.CLKSEL_PRU_ICSS_OCP_CLK[0]
D	PRCM.CM_CPTS_RFT_CLKSEL[0]
F	PRCM.CLKSEL_ADC1_CLK[0]

Table 6-22 gives the typical PLL and clock frequencies. The HSDIVIDER is used to generate three divided clocks M4, M5, and M6. M4 and M5 are nominally 200 and 250 MHz, respectively.

Table 6-22. Core PLL Typical Frequencies (MHz)

CLOCK	Source	Power-On-Reset / HSDIVIDER Bypass		OPP100		OPP50[1]	
		DIV	Freq	DIV Value	Freq (MHz)	DIV Value	Freq (MHz)
CLKDCOLDO (PLL Lock frequency)	APLLS	-	-	-	2000	-	100
CORE_CLKOUTM4	HSDIVIDER-M4	-	Mstr Xtal	10	200	1	100
L3F_CLK, L4F_CLK, PRU-ICSS IEP CLK, DebugSS clka, SGX.MEMCLK, SGX.SYSCLK	CORE_CLKOUTM4	-	Mstr Xtal	-	200	-	100
L4_PER, L4_WKUP	CORE_CLKOUTM4	2	Mstr Xtal / 2	2	100	2	50
SGX CORECLK	CORE_CLKOUTM4	1	Mstr Xtal	1	200	1	100
				2	100	2	50
CORE_CLKOUTM5	HSDIVIDER-M5	-	Mstr Xtal	8	250	1	100
MHZ_250_CLK (Gigabit RGMII)	CORE_CLKOUTM5	-	NA	-	250	-	NA
MHZ_125_CLK (Ethernet Switch Bus Clk)	CORE_CLKOUTM5	2	Mstr Xtal / 2	2	125	2	50
MHZ_50_CLK (100 mbps RGMII or 10/100 RMII)	CORE_CLKOUTM5	5	Mstr Xtal / 5	5	50	2	50
MHZ_5_CLK (10 mbps RGMII)	MHZ_50_CLK	10	Mstr Xtal / 50	10	5	10	5
CORE_CLKOUTM6	HSDIVIDER M6	-	Mstr Xtal	4	500	1	100

[1] Not all interfaces and peripheral modules are available in OPP50. For more information, see the device-specific datasheet.

Table 6-23. Bus Interface Clocks

L3F_CLK	SGX530 (MEMCLK and SYSCLK), MPU Subsystem, GEMAC Switch (Ethernet), DAP, PRU-ICSS, EMIF, TPTC, TPCC, OCMC RAM, DEBUGSS, DSS, EDMA, VPFE
L3S_CLK	USB, ADC0, GPMC, MMCSD2, McASP0, McASP1, QSPI, ADC1
L4_PER_CLK	DCAN0, DCAN1 DMTIMER2, DMTIMER3, DMTIMER4, DMTIMER5, DMTIMER6, DMTIMER7, DMTIMER8, DMTIMER9, DMTIMER10, DMTIMER11 eCAP/eQEP/ePWM0, eCAP/eQEP/ePWM1, eCAP/eQEP/ePWM2, eFuse ELM, GPIO1, GPIO2, GPIO3, GPIO4, GPIO5 I2C1, I2C2, IEEE1500, Mailbox0 HDQ1W, McASP0, McASP1 MMCSD0, MMCSD1, OCP Watchpoint, SPI0, SPI1, Spinlock UART1, UART2, UART3, UART4, UART5
L4_WKUP_CLK	Clock Manager, Control Module DMTIMER0, DMTIMER_1MS, SyncTimer32k, RTCSS, GPIO0 I2C0, UART0, WDT1

6.6.7.1 Configuring the Core PLL

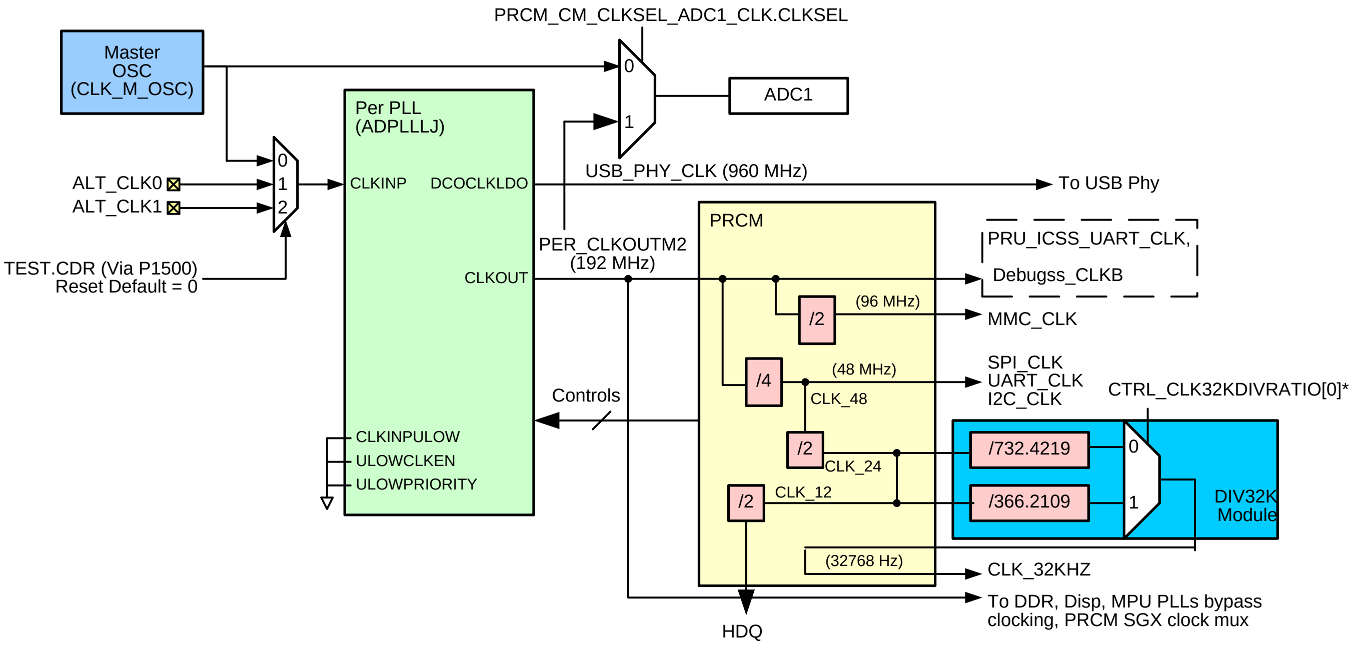
1. Switch PLL to bypass mode by setting CM_CLKMODE_DPLL_CORE.DPLL_EN to 0x4.
2. Wait for CM_IDLEST_DPLL_CORE.ST_MN_BYPASS = 1 to ensure PLL is in bypass (CM_IDLEST_DPLL_CORE.ST_DPLL_CLK should also change to 0 to denote the PLL is unlocked).
3. Configure Multiply and Divide values by setting CM_CLKSEL_DPLL_CORE.DPLL_MULT and DPLL_DIV to the desired values.
4. Configure M4, M5 and M6 dividers by setting HSDIVIDER_CLKOUT1_DIV bits in CM_DIV_M4_DPLL_CORE,CM_DIV_M5_DPLL_CORE, and CM_DIV_M6_DPLL_CORE to the desired values.
5. Switch over to lock mode by setting CM_CLKMODE_DPLL_CORE.DPLL_EN to 0x7.
6. Wait for CM_IDLEST_DPLL_CORE.ST_DPLL_CLK = 1 to ensure PLL is locked (CM_IDLEST_DPLL_CORE.ST_MN_BYPASS should also change to 0 to denote the PLL is out of bypass mode).

Note: M4, M5, and M6 dividers can also be changed on-the-fly so that there is no need to put the PLL in bypass and back to lock mode. After changing CM_DIV_Mx_DPLL_CORE.DPLL_CLKOUT_DIV, check CM_DIV_Mx_DPLL_CORE.DPLL_HSDIVIDER_CLKOUT1_DIVCHACK for a toggle (a change from 0 to 1 or 1 to 0) to see if the change was acknowledged by the PLL.

6.6.8 Peripheral PLL Description

The Per PLL provides the source for peripheral functional clocks. The Per PLL comprises an ADPLLLJ and additional dividers and muxes located in the PRCM as shown

Figure 6-15. Peripheral PLL Structure



* Reset default zero

ALT_CLKs are to be used for internal test purpose and should not be used in functional mode.

The PLL is locked at 960 MHz. The PLL output is divided by the M2 divider to generate a 192-MHz CLKOUT. This clock is gated in the PRCM to form the PRU-ICSS UART clock. There is a /2 divider to create 96 MHz for MMC_CLK. The clock is also divided within the PRCM by a fixed /4 divider to create a 48-MHz clock for the SPI, UART and I2C modules. The 48-MHz clock is further divided by a fixed /2 divider and a fixed /732.4219 divider to create an accurate 32.768-KHz clock for Timer and debounce use.

Table 6-24. Per PLL Typical Frequencies (MHz)

Clock	Source	Power-On-Reset / PLL Bypass		OPP100		OPP50[(1)][(2)]	
		DIV Value	Freq	DIV Value	Freq (MHz)	DIV Value	Freq (MHz)
PLL Lock frequency	PLL	-	-	-	960	-	960
USB_PHY_CLK	CLKDCOLDO	-	Held Low	-	960	-	960
PER_CLKOUTM2	CLKOUT of ADPLLLJ CLKOUT uses PLL's M2 Divider when PLL is locked and PLL's N2 divider when PLL Bypass	N2 is 0 on power-on-reset	Mstr Xtal/ (N2+1)	5	192	5	192
MMC_CLK	PER_CLKOUTM2	2	Mstr Xtal/ ((N2+1)*2)	2	96	2	96
SPI_CLK, UART_CLK, I2C_CLK	PER_CLKOUTM2	4	Mstr Xtal/ ((N2+1)*4)	4	48	4	48
CLK_24	CLK_48	2	CLK_48 /2	2	24	2	24
CLK_32KHZ	CLK_24 (output of CLK_48/2)	732.4219	CLK_24 / <CLK32_DIV>	732.4219	0.032768	732.4219	0.032768

(1) For limitations using OPP50, see the device-specific data manual.
(2) Not all interfaces and peripheral modules are available in OPP50. For more information, see the device-specific data manual.

6.6.8.1 Configuring the Peripheral PLL

The following steps detail how to configure the peripheral PLL.

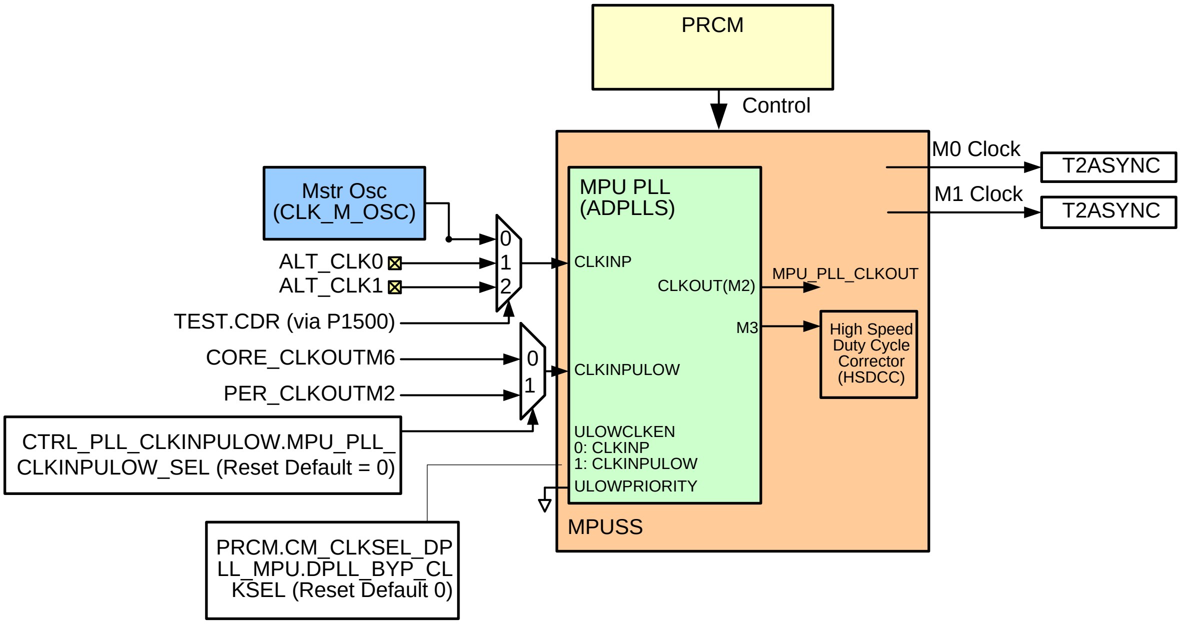
1. Switch PLL to bypass mode by setting CM_CLKMODE_DPLL_PER.DPLL_EN to 0x4.
2. Wait for CM_IDLEST_DPLL_PER.ST_MN_BYPASS = 1 to ensure PLL is in bypass (CM_IDLEST_DPLL_PER.ST_DPLL_CLK should also change to 0 to denote the PLL is unlocked).
3. Configure Multiply and Divide values by setting CM_CLKSEL_DPLL_PER.DPLL_MULT and DPLL_DIV to the desired values.
4. Configure M2 divider by setting CM_DIV_M2_DPLL_PER.DPLL_CLKOUT_DIV to the desired value.
5. Switch over to lock mode by setting CM_CLKMODE_DPLL_PER.DPLL_EN to 0x7.
6. Wait for CM_IDLEST_DPLL_PER.ST_DPLL_CLK = 1 to ensure PLL is locked (CM_IDLEST_DPLL_PER.ST_MN_BYPASS should also change to 0 to denote the PLL is out of bypass mode).

Note: M2 divider can also be changed on-the-fly (ie., there is no need to put the PLL in bypass and back to lock mode). After changing CM_DIV_M2_DPLL_PER.DPLL_CLKOUT_DIV, check CM_DIV_M2_DPLL_PER.DPLL_CLKOUT_DIVCHACK for a toggle (a change from 0 to 1 or 1 to 0) to see if the change was acknowledged by the PLL.

6.6.9 MPU PLL Description

The MPU subsystem includes an internal ADPLLS for generating the required MPU clocks. This PLL is driven by the master oscillator output with control provided by PRCM registers.

Figure 6-16. MPU Subsystem PLL Structure



For example:

For a frequency for MPU, say 600 MHz, the ADPLLS is configured (PLL locked at 1200 MHz and M2 Divider =1) so as to expect CLKOUT = 600 MHz .

The ULOWCLKEN input from a programmable PRCM register selects whether CLKINP or CLKINPULOW is the bypass clock source. This is a glitch free switch. When CLKINP is selected it is sourced through the ADPLLS 1/(N2+1) divider. The PRCM register defaults to 0 on power-up to select the CLKINP source.

The CLKINPULOW input may be sourced from the CORE_CLKOUTM6 from the Core PLL, or PER_CLKOUTM2 from the Per PLL. These PLL output clocks can be used as alternate clock sources in low power active use cases for the MPU Subsystem clock when the PLL is in bypass mode.

6.6.9.1 Configuring the MPU PLL

The following steps detail how to configure the MPU PLL.

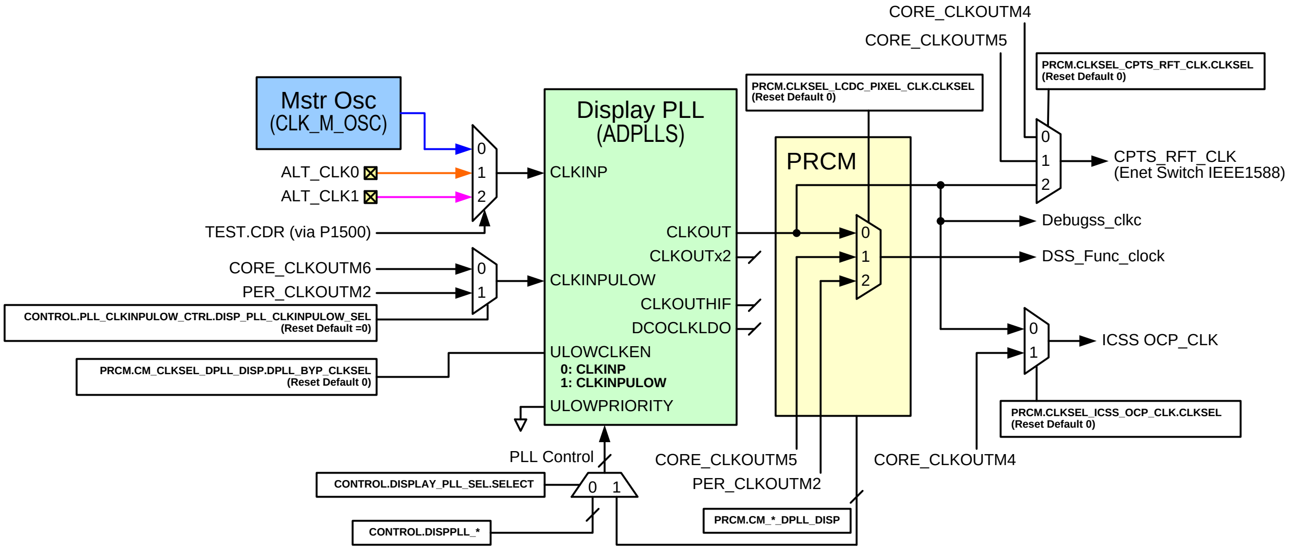
1. Switch PLL to bypass mode by setting CM_CLKMODE_DPLL_MPU.DPLL_EN to 0x4.
2. Wait for CM_IDLEST_DPLL_MPU.ST_MN_BYPASS = 1 to ensure PLL is in bypass (CM_IDLEST_DPLL_MPU.ST_DPLL_CLK should also change to 0 to denote the PLL is unlocked).
3. Configure Multiply and Divide values by setting CM_CLKSEL_DPLL_MPU.DPLL_MULT and DPLL_DIV to the desired values.
4. Configure M2 divider by setting CM_DIV_M2_DPLL_MPU.DPLL_CLKOUT_DIV to the desired value.
5. Switch over to lock mode by setting CM_CLKMODE_DPLL_MPU.DPLL_EN to 0x7.
6. Wait for CM_IDLEST_DPLL_MPU.ST_DPLL_CLK = 1 to ensure PLL is locked (CM_IDLEST_DPLL_MPU.ST_MN_BYPASS should also change to 0 to denote the PLL is out of bypass mode).

Note: M2 divider can also be changed on-the-fly (ie., there is no need to put the PLL in bypass and back to lock mode). After changing CM_DIV_M2_DPLL_MPU.DPLL_CLKOUT_DIV, check CM_DIV_M2_DPLL_MPU.DPLL_CLKOUT_DIVCHACK for a toggle (a change from 0 to 1 or 1 to 0) to see if the change was acknowledged by the PLL.

6.6.10 Display PLL Description

The Display PLL provides the pixel clock required for the LCD display and is independent from the other peripheral and infrastructure clocks. The PLL is clocked from the Master Oscillator. The ADPLLS M2 divider determines the output clock frequency which is clock gated by the PRCM as shown in Figure 6-17.

Figure 6-17. Display PLL Structure



The display PLL also provides a clock to the time sync module clock of the Ethernet switch (CPTS_RFT_CLK). This PLL can be optionally controlled using the control module registers (CTRL_DISPPLL_i) for on-the-fly fine tuning of the time sync clock frequency using the fractional M multiplier. This synchronizes the frequency with the external master clock in an Ethernet IEEE 1588-compliant system. The clock frequency would be fine tuned up to ±3% of the nominal clock frequency of 250 MHz, that is, effectively the time sync module is required to run up to a maximum frequency of 258 MHz.

For example: say frequency for pixel clock 100 MHz, the ADPLLS is configured (PLL locked at 200 MHz and M2 Divider =1) so as to expect CLKOUT = 100 MHz.

The ULOWCLKEN input from a programmable PRCM register selects whether CLKINP or CLKINPULOW is the bypass clock source. This is a glitch free switch. When CLKINP is selected it is sourced through the ADPLLS 1/(N2+1) divider. The PRCM register defaults to 0 on power-up to select the CLKINP source.

The CLKINPULOW input is sourced from the CORE_CLKOUTM6 from the Core PLL or PER_CLKOUTM2 from the Per PLL. This PLL output clock can be used as an alternate clock source in low power active use cases for the pixel clock when the Display PLL is in bypass mode.

6.6.10.1 Configuring the Display PLL

The following steps detail how to configure the display PLL.

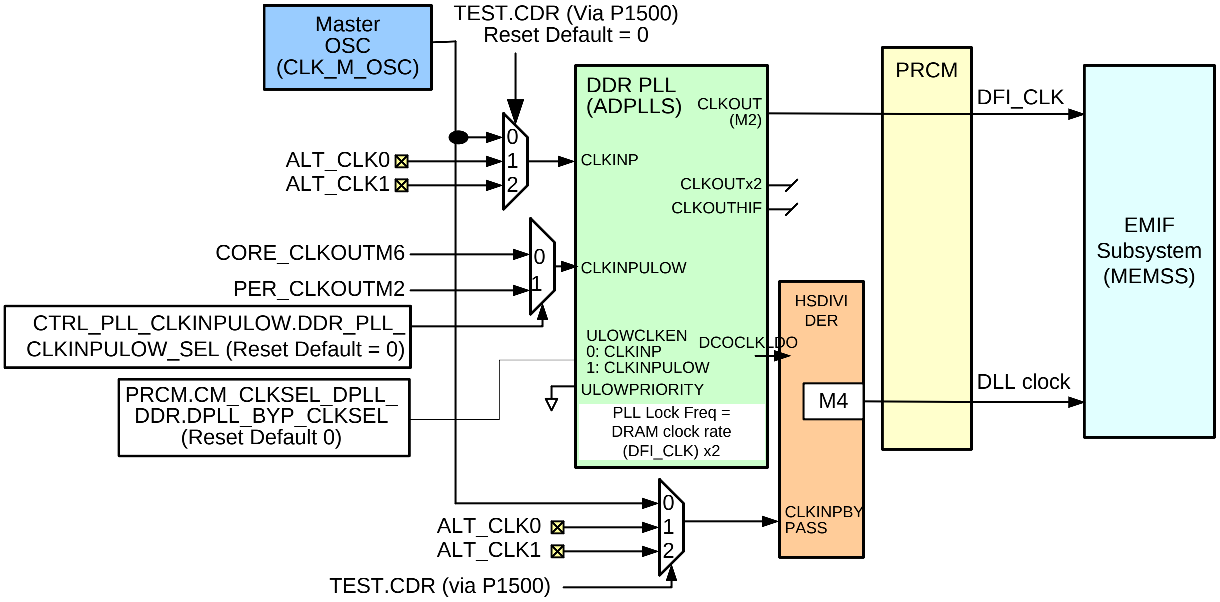
1. Switch PLL to bypass mode by setting CM_CLKMODE_DPLL_DISP.DPLL_EN to 0x4.
2. Wait for CM_IDLEST_DPLL_DISP.ST_MN_BYPASS = 1 to ensure PLL is in bypass (CM_IDLEST_DPLL_DISP.ST_DPLL_CLK should also change to 0 to denote the PLL is unlocked).
3. Configure Multiply and Divide values by setting CM_CLKSEL_DPLL_DISP.DPLL_MULT and DPLL_DIV to the desired values.
4. Configure M2 divider by setting CM_DIV_M2_DPLL_DISP.DPLL_CLKOUT_DIV to the desired value.
5. Switch over to lock mode by setting CM_CLKMODE_DPLL_DISP.DPLL_EN to 0x7.
6. Wait for CM_IDLEST_DPLL_DISP.ST_DPLL_CLK = 1 to ensure PLL is locked (CM_IDLEST_DPLL_DISP.ST_MN_BYPASS should also change to 0 to denote the PLL is out of bypass mode).

Note: M2 divider can also be changed on-the-fly (ie., there is no need to put the PLL in bypass and back to lock mode). After changing CM_DIV_M2_DPLL_DISP.DPLL_CLKOUT_DIV, check CM_DIV_M2_DPLL_DISP.DPLL_CLKOUT_DIVCHACK for a toggle (a change from 0 to 1 or 1 to 0) to see if the change was acknowledged by the PLL.

6.6.11 DDR PLL Description

The DDR PLL provides the clocks required by the DDR macros and the EMIF and is independent from the other peripheral and infrastructure clocks. The PLL is clocked from the Master Oscillator. The ADPLLS M2 divider determines the output clock frequency which is connected to the EMIF subsystem (MEMSS) for clocking the memory interfaces. An HSDIVIDER, clocked by DCOCLKLDO, provides the DLL clock, which is an integer ratio with the DFI_CLK (going to MEMSS and DRAM), as Figure 6-18 shows.

Figure 6-18. DDR PLL Structure



For OPP information, see the device-specific data manual.

For example:

- The PLL lock frequency is 666 MHz.

- M2 Divider equals 1, so as to expect CLKOUT/DFI_CLK is 333 MHz.
- The DDR pin clock will be 333 MHz.

The ULOWCLKEN input from a programmable PRCM register selects whether CLKINP or CLKINPULOW is the bypass clock source. This is a glitch free switch. When CLKINP is selected it is sourced through the ADPLLS 1/(N2+1) divider. The PRCM register defaults to 0 on power-up to select the CLKINP source.

The CLKINPULOW input may be sourced from the CORE_CLKOUTM6 from the Core PLL, or PER_CLKOUTM2 from the Per PLL. These PLL output clocks can be used as alternate clock sources in low power active use cases for the DDR clocks when PLL is in bypass mode

6.6.11.1 Configuring the DDR PLL

The following steps detail how to configure the DDR PLL.

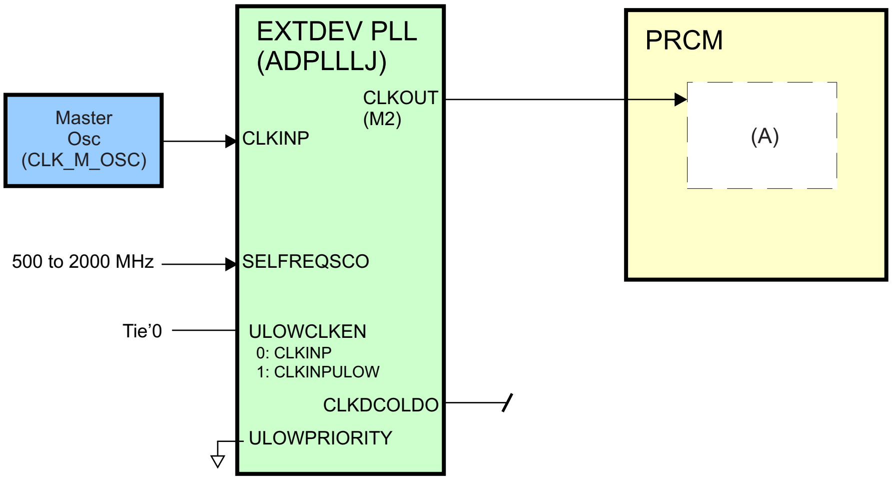
1. Switch PLL to bypass mode by setting CM_CLKMODE_DPLL_DDR.DPLL_EN to 0x4.
2. Wait for CM_IDLEST_DPLL_DDR.ST_MN_BYPASS = 1 to ensure PLL is in bypass (CM_IDLEST_DPLL_DDR.ST_DPLL_CLK should also change to 0 to denote the PLL is unlocked).
3. Configure Multiply and Divide values by setting CM_CLKSEL_DPLL_DDR.DPLL_MULT and DPLL_DIV to the desired values.
4. Configure M2 divider by setting CM_DIV_M2_DPLL_DDR.DPLL_CLKOUT_DIV to the desired value.
5. Switch over to lock mode by setting CM_CLKMODE_DPLL_DDR.DPLL_EN to 0x7.
6. Wait for CM_IDLEST_DPLL_DDR.ST_DPLL_CLK = 1 to ensure PLL is locked (CM_IDLEST_DPLL_DDR.ST_MN_BYPASS should also change to 0 to denote the PLL is out of bypass mode).

Note: M2 divider can also be changed on-the-fly (i.e., there is no need to put the PLL in bypass and back to lock mode). After changing CM_DIV_M2_DPLL_DDR.DPLL_CLKOUT_DIV, check CM_DIV_M2_DPLL_DDR.DPLL_CLKOUT_DIVCHACK for a toggle (a change from 0 to 1 or 1 to 0) to see if the change was acknowledged by the PLL.

6.6.12 EXTDEV PLL Description

This PLL output feeds to CLKOUTs. It will generate clocks for external devices such as a modem. The lock frequency depends on the external clock device. The PRCM must support a fractional multiplier for this PLL.

Figure 6-19. EXTDEV PLL Structure



(A): CLKOUT mux logic. See Section 6.6.14, *CLKOUT Signals*.

6.6.12.1 Configuring the EXTDEV PLL

The SELFREQSCO tieoff value must be defined in the range of 500 to 2000 MHz to give the best jitter.

6.6.13 PLL Bypass Modes

When an active PLL is not required, the PLL can be configured into a bypass mode to reduce power consumption. Each bypass mode has unique power and latency characteristics, which should be considered when selecting the appropriate operating mode. The supported modes are handled through the CLKMODE_DPLL_X.DPLL_EN register bitfield.

The ADPLLS module supports three different bypass modes through their internal MNBypass mode, external Low Power Idle bypass mode, and Fast Relock bypass mode. The PLLs are in the MNBypass mode after power-on-reset and can be configured by software to enter the other bypass modes for power-down. The MNBypass and Low Power Idle bypass modes gate internal clocks and turn off the analog blocks of the DPLL to reduce power consumption, but at a cost to relock latency. The Fast Relock bypass mode keeps the DPLL analog bias and LDO active and provides the lowest latency relock period. When the Core PLL is configured in bypass mode, the HSDIVIDER enters bypass mode and the CLKINBYPASS input is driven on the M4, M5, and M6 outputs. CLKINBYPASS defaults to the master oscillator input (typically 24 MHz).

The ADPLLLJ module supports two different bypass modes through their internal MNBypass mode and their external Low Power Idle bypass mode. Fast Relock bypass is not supported for ADPLLLJ. The PER PLL can use the Low Power Idle bypass mode. When the internal bypass mode is selected, the CLKOUT output is driven by CLKINP/(N2+1) where N2 is driven by the PRCM. CLKINP defaults to the master oscillator input (typically 24 MHz).

Table 6-25 describes the relock times and power consumption for the PLL bypass modes, for nominal process at 30°C. Frequency Lock Time is the maximum latency from the bypass mode to when the internal normal clk frequency is within +/-1% of the final output frequency, of Active and Locked mode. Phase Lock Time is the maximum latency from the bypass mode to when, internally, the phase difference between FBCLK and REFCLK is less than 6–12% of the REFCLK period for 96 continuous REFCLKs, of Active and Locked mode.

Table 6-25. Latency and Power for PLL Bypass Modes

	MN Bypass	**Low Power Bypass**	**Fast Relock Bypass**
Frequency Relock Time	1.9us + 350REFCLKs	1.9us + 70REFCLKs	0.05us + 70REFCLKs
Phase Relock Time	1.9us + 500REFCLKs	1.9us + 120REFCLKs	0.05us + 120REFCLKs

If temperature is steady when entering into and coming out of the bypass modes, the lock time for relock is significantly lower compared to the initial lock time. However, if the temperature drift exceeds 10°C, the frequency and phase relock times increase (see Table 6-26). Once the module is locked, it can tolerate any amount of change in temperature within the operating temperature range. Because the typical rate of change in temperature is very slow compared to the loop BW (~REFCLK/50), the loop will be able to track changes in temperature.

Additional relock time will be taken if exit of bypass mode is triggered without the DPLL being fully in bypass mode.

Table 6-26. Effects of Temperature Drift on Relock

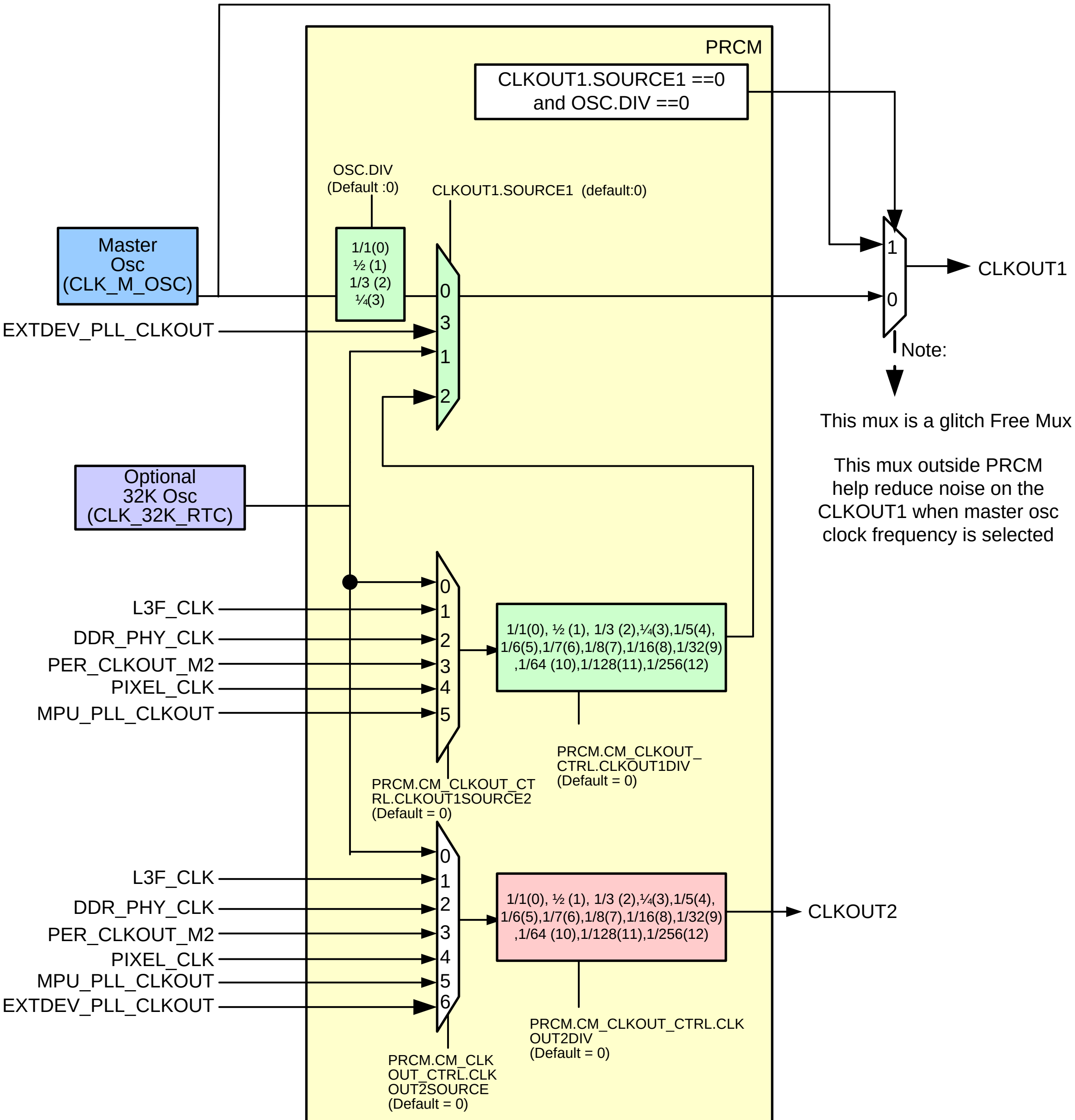
Temperature Drift (°C)	**Low Power Bypass**		**Fast Relock Bypass**	
	Frequency Relock	**Phase Relock**	**Frequency Relock**	**Phase Relock**
ΔT <= 10	1.9us + 70REFCLKs	1.9us + 120REFCLKs	0.05us + 70REFCLKs	0.05us + 120REFCLKs
ΔT > 10	1.9us + 100REFCLKs	1.9us + 150REFCLKs	0.05us + 100REFCLKs	0.05us + 150REFCLKs

6.6.14 CLKOUT Signals

The CLKOUT1 and CLKOUT2 signals go device pads and should mainly be used as debug testpoints.

Figure 6-20 shows the different clock sources coming from the PLL or master oscillator and their routing to CLKOUT1 and CLKOUT2 pads. The CLKOUT1 and CLKOUT2 signals go to device pads and can be used as source clocks for FPGAs or other system devices.

Figure 6-20. CLKOUT Architecture



This mux is a glitch Free Mux

This mux outside PRCM help reduce noise on the CLKOUT1 when master osc clock frequency is selected

6.6.15 32-kHz Clock Structure

The 32-kHz crystal oscillator is used by the RTCSS. The device also contains an on-chip RC oscillator. The RC oscillator is not configurable but may be enabled or disabled through the Control Module RCOSC_CTRL register. The 32-kHz clocks are summarized in Table 6-27.

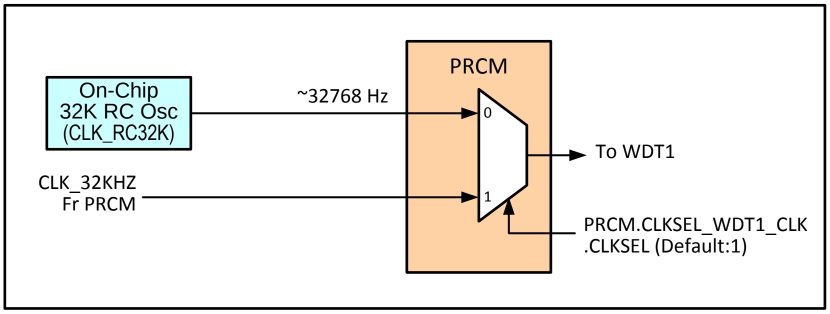
Table 6-27. 32-kHz Clock Summary

Clock Name	Source	Accuracy
CLK_32KHZ	Divide down of PER PLL output (PLL uses Master Osc).	32768 Hz Precise
CLK_32K_RC	Internal RC Oscillator.	16 to 60 kHz
CLK_32K_RTC	External 32768 Hz crystal with internal 32K Osc or external 32768-Hz clock.	32768 Hz Precise
CLK_32K_MOSC	Divide down of Master Oscillator Crystal Frequency	~ 32768 Hz

6.6.15.1 Watchdog Timer Clocking

The RC oscillator is inaccurate and can vary in frequency from 16 to 60 kHz. The clock options for watchdog1 are shown in Figure 6-21.

Figure 6-21. Watchdog Timer Clock Selection

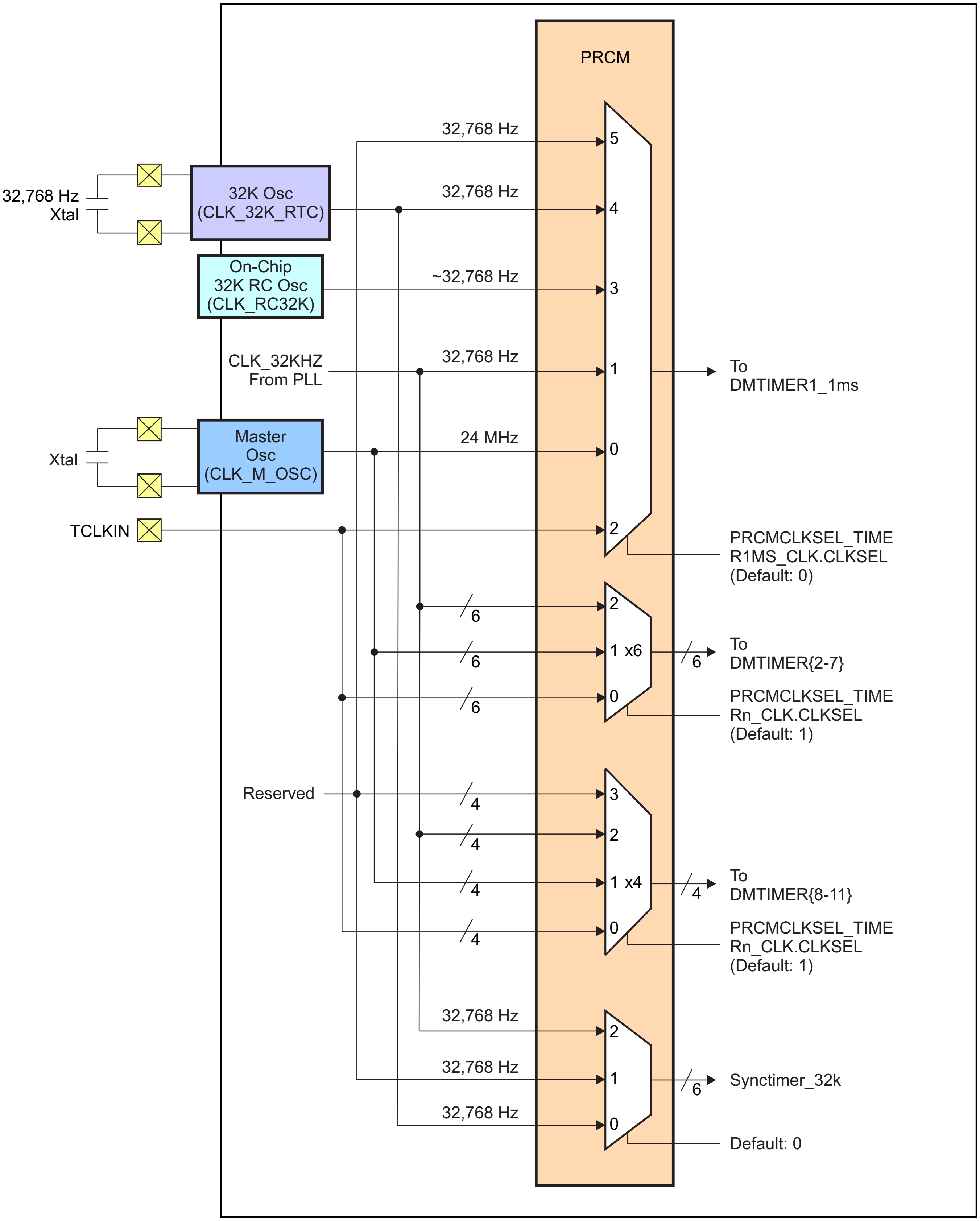


6.6.15.2 Timer Clocking

The clock selections for the other Timer modules are shown in Figure 6-22. CLK_32KHZ, the master oscillator, and the external pin (TCLKIN) are optional clocks available for all timers, which may be selected based on end use application.

DMTIMER1 is implemented using the dmtimer_1ms module, which is capable of generating an accurate 1ms tick using a 32.768-kHz clock. During low power modes, the master oscillator is disabled. CLK_32KHZ is not available in this scenario since it is sourced from the master oscillator-based PER PLL. Hence, in low power modes, DMTIMER1 in the WKUP domain can use the 32K clock sources from the one of the 32-kHz oscillators for timer-based wakeup. The synctimer32K is a free running timer which can be used for time reference by the operating system.

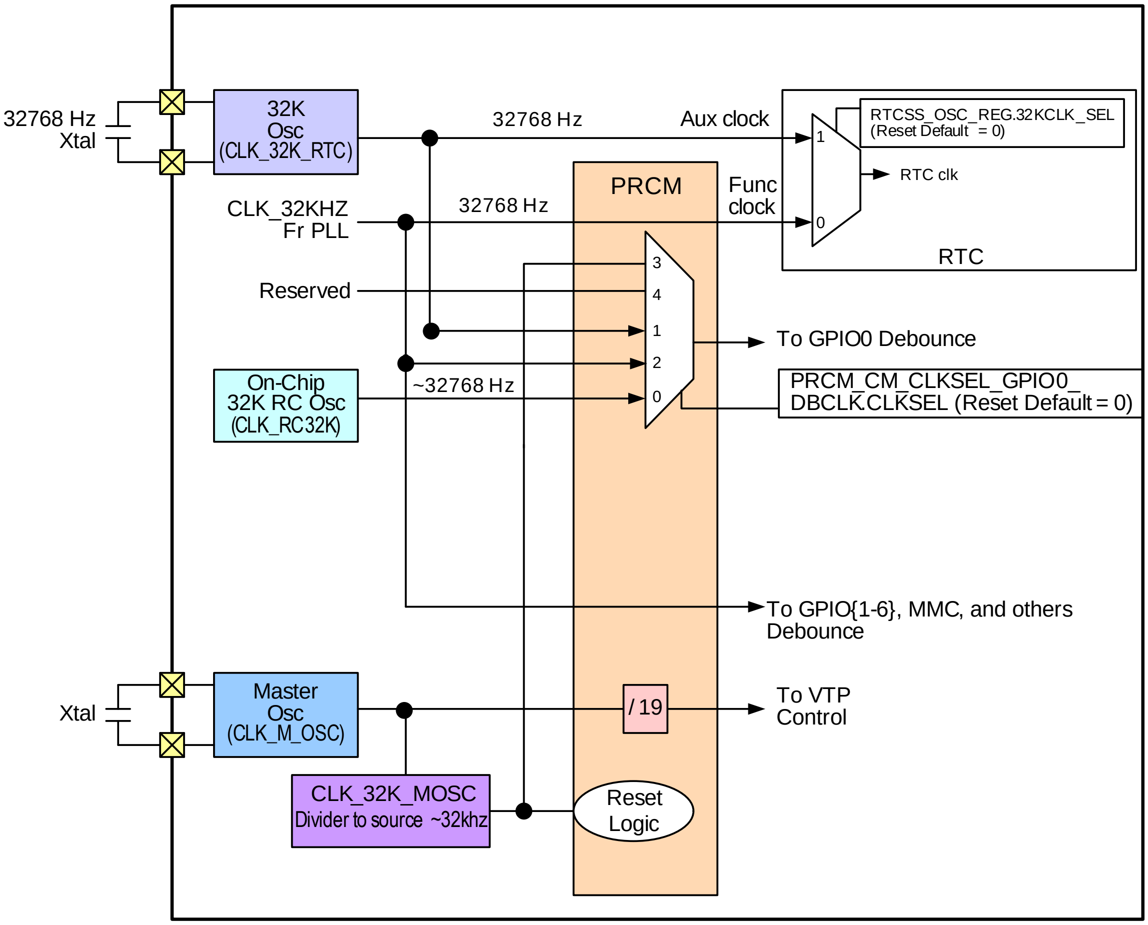
Figure 6-22. Timer Clock Selection



6.6.15.3 RTC, VTP, Debounce and Reset Clocking

Figure 6-23 shows the clocking for RTC, DDR VTP macro, debounce clocks for GPIO, and the reset logic of PRCM. In low power modes, the debounce for GPIO0 in WKUP domain can use the accurate 32768-Hz crystal oscillator or the inaccurate (16 to 60 kHz) 32K RC oscillator when the master oscillator is powered down.

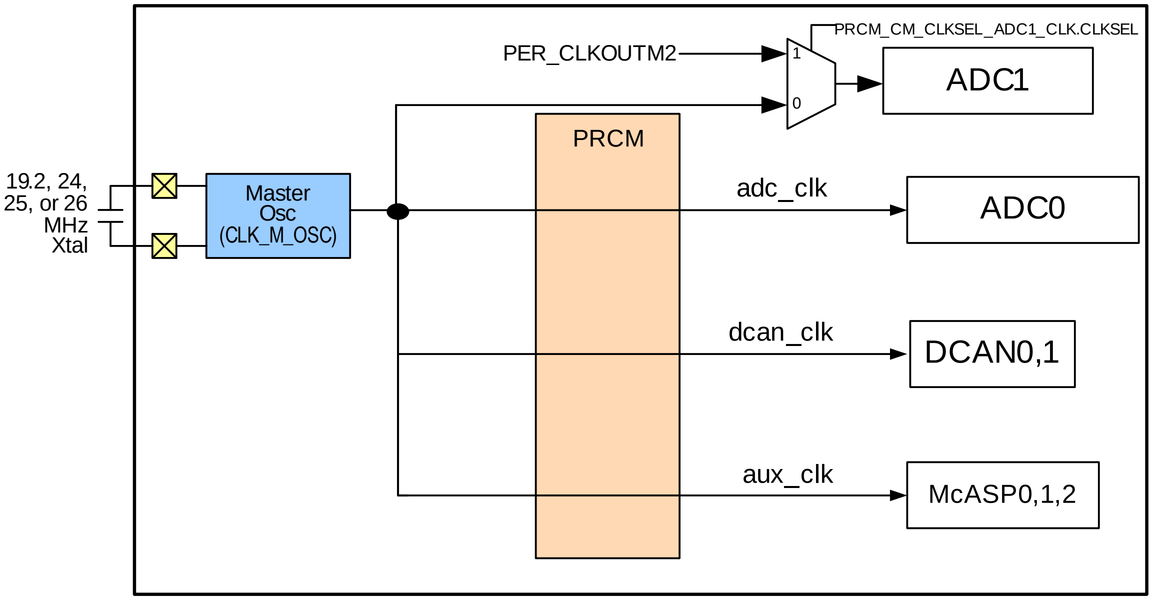
Figure 6-23. RTC, VTP and Debounce Clock Selection



6.6.16 ADC0, ADC1, DCAN, and McASP Clocking

Figure 6-24 shows the functional clock sources for ADC0, ADC1, DCAN, and McASP.

Figure 6-24. ADC0, ADC1, DCAN, and McASP Clock Selection



6.7 Reset Management

6.7.1 Overview

The PRCM manages the resets to all power domains inside the device and generates a single reset output signal through device pin, WARMRSTn, for external use. The PRCM has no knowledge of or control over resets generated locally within a module, e.g., via the OCP configuration register bit: PeripheralName_SYSCONFIG.SoftReset.

All PRM reset outputs are asynchronously asserted. These outputs are active-low except for the PLL resets. Deassertion is synchronous to the clock which runs a counter used to stall, or delay, reset de-assertion upon source deactivation. This clock will be CLK_M_OSC used by all the reset managers. All modules receiving a PRCM generated reset are expected to treat the reset as asynchronous and implement local re-synchronization upon de-activation as needed.

One or more Reset Managers are required per power domain. Independent management of multiple reset domains is required to meet the reset sequencing requirements of all modules in the power domain

6.7.2 Reset Concepts and Definitions

The PRCM collects many sources of reset. Here below is a list of qualifiers of the source of reset:

- Cold reset: it affects all the logic in a given entity
- Warm reset: it is a partial reset which doesn't affect all the logic in a given entity
- Global reset: it affects the entire device
- Local reset: it affects part of the device (1 power domain for example)
- S/W reset: it is initiated by software
- H/W reset: it is hardware driven

Each reset source is specified as being a cold or warm type. Cold types are synonymous with power-on-reset (POR) types. Such sources are applied globally within each receiving entity (i.e., sub-system, module, macro-cell) upon assertion. Cold reset events include: device power-up, power-domain power-up, and eFuse programming failures.

Warm reset types are not necessarily applied globally within each receiving entity. A module may use a warm reset to reset a subset of its logic. This is often done to speed-up reset recovery time, i.e., the time to transition to a safe operating state, compared to the time required upon receipt of a cold reset. Warm reset events include: software initiated per power-domain, watchdog timeout, externally triggered, and emulation initiated.

Reset sources, warm or cold types, intended for device-wide effect are classified as global sources. Reset sources intended for regional effect are classified as local sources.

Each Reset Manager provides two reset outputs. One is a cold reset generated from the group of global and local cold reset sources it receives. The other is a warm+cold reset generated from the combined groups of, global and local, cold and warm reset sources it receives.

The Reset Manager asserts one, or both, of its reset outputs asynchronously upon reset source assertion. Reset deassertion is extended beyond the time the source gets de-asserted. The reset manager will then extend the active period of the reset outputs beyond the release of the reset source, according to the PRCM’s internal constraints and device’s constraints. Some reset durations can be software-configured. Most (but not all) reset sources are logged by PRCM’s reset status registers. The same reset output can generally be activated by several reset sources and the same reset source can generally activate several reset outputs. All the reset signals output of the PRCM are active low. Several conventions are used in this document for signal and port names. They include:

- "_RST" in a signal or port name is used to denote reset signal.
- "_PWRON_RST" in a signal or port name is used to denote a cold reset source

6.7.3 Global Power On Reset (Cold Reset)

There are several cold reset sources. See Table 6-28 for a summary of the different reset sources.

6.7.3.1 Power On Reset (PORz)

The source of power on reset is PORz signal on the device. Everything on device is reset with assertion of power on reset. This reset is non-blockable. PORz can be driven by external power management devices or power supervisor circuitry. During power-up, when power supplies to the device are ramping up, PORz needs to be driven Low. When the ramp-up is complete and supplies reach their steady-state values, PORz need to be driven High. During normal operation when any of the device power supplies are turned OFF, PORz must be driven Low.

6.7.3.2 PORz Sequence

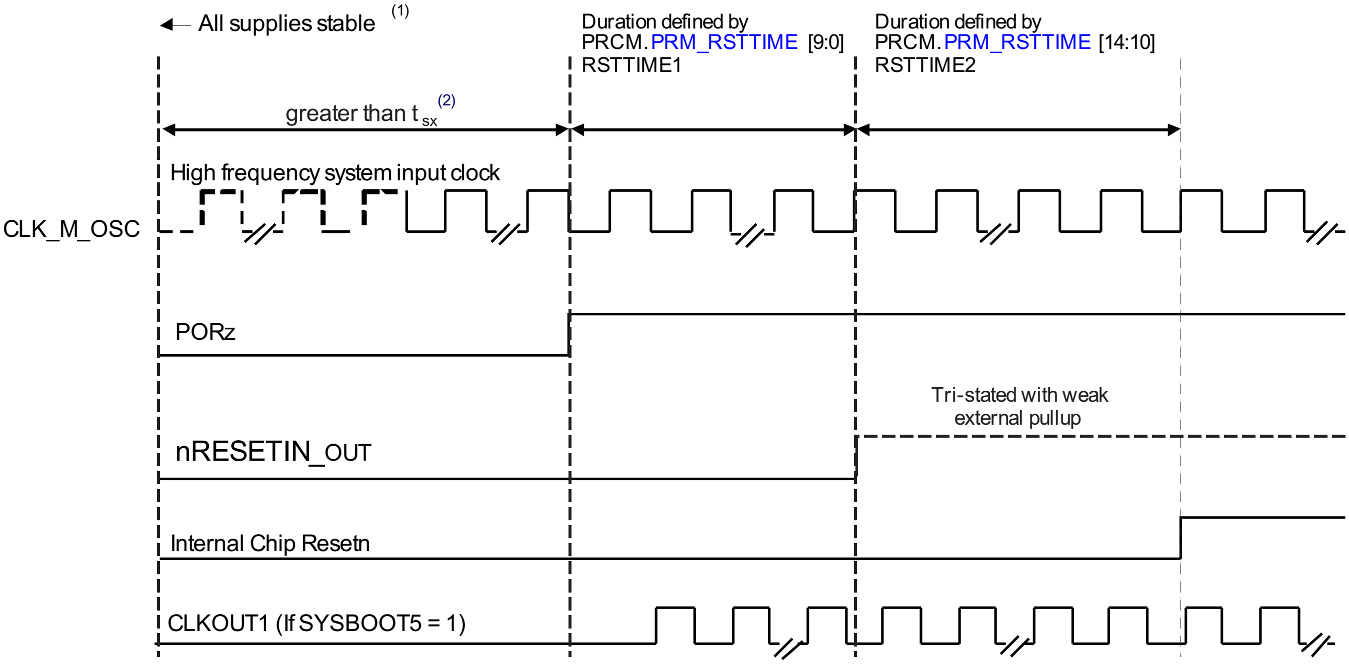
1. PORz pin at chip boundary gets asserted (goes low). Note: The state of nRESETIN_OUT during PORz assertion should be a don’t care, it should not affect PORz (only implication is if they are both asserted and nRESETIN_OUT is deasserted after PORz you will get re-latching of boot config pins and may see warm nRESETIN_OUT flag set in PRCM versus POR).
2. The processor drives Warm Reset Out (nRESETIN_OUT pin) to Low. All other IOs are tristated with pull values, as defined in the device datasheet.
3. When power comes up, PORz value will propagate to the PRCM.
4. PRCM will fan out reset to the complete chip and all logic which uses async reset will get reset. nRESETIN_OUT will go low to indicate reset-in-progress.
5. External clocks will start toggling and the PRCM will propagate these clocks to the chip while keeping PLLs in bypass mode.
6. All logic using sync reset will be reset.
7. When all power and clocks to the chip are stable, PORz must be de-asserted (this is handled externally by the PMIC or Supervisor Circuitry).
8. Boot configuration pins are latched when PORz is de-asserted (on rising edge of PORz). (1)
9. IO cell controls from peripherals for all the IOs with a few exceptions (see datasheet for details) are driven by the GPIO peripheral. GPIO puts all IOs in input mode with a few exceptions (see datasheet).

---

(1) In order to simplify end system design, it is desired that SysBoot configuration inputs have no hold time requirement relative the the PORz riding edge. In order to achieve this 0 ns hold time, the inputs need to be sampled continuously during the PORz active period and the final sampled value prior to the (last) rising edge latched in the register.

10. PRCM internal state machines will be released after RSTTIME1 (based on 32K MOSC divided clock).
11. FuseFarms reset will be de-asserted to start eFuse scanning.
12. nRESETIN_OUT is de-asserted after PRCM.RSTTIME1 time is met.
13. As soon as nRESETIN_OUT is de-asserted (system warm reset out):
    a. The RSTTIME2 count starts for other domain resets.
    b. The RSTTIME2 count starts for MPU pwron reset release. After the counter expires, MPU pwron reset is released. This is followed by Reset done from the MPU and then the MPU_RST is de-asserted.
14. Internal Resets to all peripherals without local reset will be de-asserted.
15. Once the device finishes booting, all remaining peripherals will see reset de-assertion.

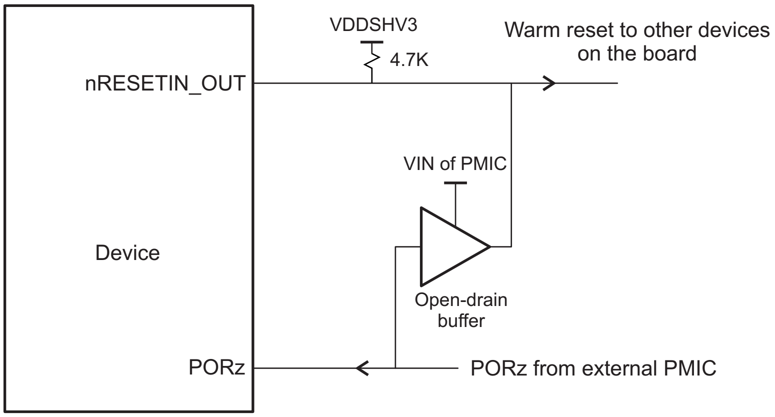
Figure 6-25. PORz



(1) nRESETIN_OUT is not defined (can either be driven low or pulled up high) until all supplies are fully ramped up. For nRESETIN_OUT to maintain a valid low state until the supplies are ramped, an external buffer should be implemented, as shown in Figure 6-26.

(2) For information on t_{sx}, see *AM437x ARM Cortex-A9* Processors (literature number SPRS851)

Figure 6-26. External Buffer for nRESETIN_OUT



6.7.3.3 Bad Device Reset (BAD_DEVICE_RST)

This reset is asserted whenever the DEVICE_TYPE encodes an unsupported device type, such as the code for a "bad" device.

6.7.3.4 Global Cold Software Reset (GLOBAL_COLD_SW_RST)

The source for GLOBAL_COLD_SW_RST is generated internally by the PRM. It is activated upon setting the PRM_RSTCTRL.RST_GLOBAL_COLD_SW bit in the PRM memory map. This bit is self-clearing, i.e., it is automatically cleared by the hardware.

6.7.4 Global Warm Reset

All warm reset events must be logged in a register (PRCM.PRM_RSTST), which is isolated from warm reset. After reboot, software can identify the source of the reset and clear the bit.

6.7.4.1 External Warm Reset

nRESETIN_OUT is a bidirectional warm reset signal. As an input, it is typically used by an external source as a device reset. Refer to Table 8-24 for a summary of the differences between a warm reset and cold reset. Some of these differences are:

- The warm reset can be blocked to the EMAC switch and its reference clock source PLL using the RESET_ISO register in the Control Module.
- The warm reset assumes that clocks and power to the chip are stable from assertion through deassertion, whereas during the cold reset, the power supplies can become stable during assertion
- Some PRCM and Control module registers are warm reset insensitive and maintain their value throughout a warm reset
- SYSBOOT pins are not latched with a warm reset. The device will boot with the SYSBOOT values from the previous cold reset.
- Most debug subsystem logic is not affected by warm reset. This allows you to maintain any debug sessions throughout a warm reset event.
- PLLs are not affected by warm reset

As an output, nRESETIN_OUT can be used to reset external devices. nRESET_OUT will drive low during a cold reset or an internally generated warm reset. After completion of a cold or warm reset, nRESETIN_OUT will continue to drive low for a period defined by PRM_RSTTIME.RSTTIME1. RSTTIME1 is a timer that counts down to zero at a rate equal to the high frequency system input clock CLK_M_OSC. This allows external devices to be held in reset for some time after the device comes out of reset.

Caution must be used when implementing the nRESETIN_OUT as an bi-directional reset signal. Because of the short maximum time allowed using RSTTIME1, it does not supply an adequate debounce time for an external push button circuit. The processor could potentially start running while external components are still in reset. It is recommended that this signal be used as input only (do not connect to other devices as a reset) to implement a push button reset circuit to the device, or an output only to be able to reset other devices after an device reset completes.

6.7.4.1.1 Warm Reset Input/Reset Output (nRESETIN_OUT)

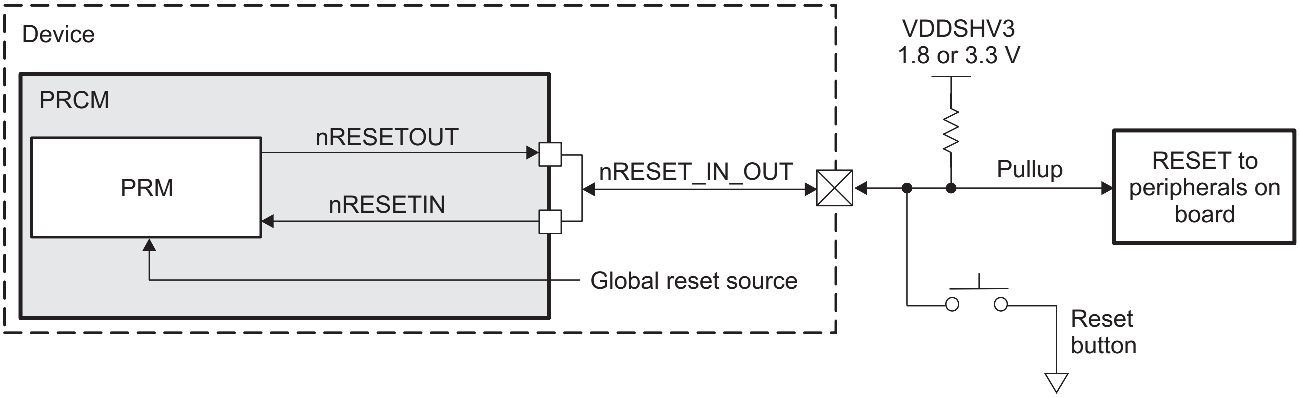
Any global reset source (internal or external) causes nRESETIN_OUT to be driven and maintained at the boundary of the device for at least the amount of time configured in the PRCM.PRM_RSTTIME. RSTTIME1 bit field. This ensures that the device and its related peripherals are reset together.

The RSTTIME1 counter must use a slower clock (such as 32-kHz clock) and the default count value is set to a 6-clock period. Software can later change the default value to a desirable setting based on the system requirements. This new value will be effective for subsequent global warm resets. However, if there is a global cold reset, the RSTTIME1 default count value will be reset to a 6-clock period.

The nRESETIN_OUT output buffer is configured as an open-drain; consequently, an external pull-up resistor is required.

After the de-assertion, the bi-directional pin nRESETIN_OUT is tri-stated to allow for assertion from off chip source (externally).

Figure 6-27. External System Reset



Note: It is recommended to implement warm reset as an input only (for example, push button) or an output only (to reset external peripherals), not both.

The device will have one pin nRESETIN_OUT which reflects chip reset status. This output will always assert asynchronously when any chip watchdog timer reset occurs if any of the following reset events occurs:

- POR (only internal stretched portion of reset event after bootstrap is latched)
- External Warm reset (nRESETIN_OUT pin, only internal stretched portion of reset event after bootstrap is latched)
- Emulation reset (Cold or warm from ICEPICK)
- Reset requestor
- SW cold/warm reset

This output will remain asserted as long as PRCM keeps reset to the host processor asserted.

Note: TRST does not cause RSTOUTn assertion

6.7.4.1.2 Warm Reset Sequence

1. nRESETIN_OUT pin at chip boundary gets asserted (goes low). NOTE: For Warm Reset sequence to work as described, it is expected that PORz pin is always inactive, otherwise you will get PORz functionality as described in previous section.
2. All IOs (except test and emulation) will go to tri-state immediately.
3. Chip clocks are not affected as both PLL and dividers are intact.
4. After the programmable timer expires, nRESETIN_OUT is de-asserted and tri-stated. At this point, if nRESETIN_OUT is still active (driven LOW) by an external source, then the devices continues to be in warm reset until nRESETIN_OUT is de-asserted (Pulled HIGH) externally.
5. After external warm-reset source is de-asserted, all internal reset to the chip will be released after PRCM.RSTTIME2 time is met.
6. Note that all peripherals with local CPUs will have local reset asserted by default at Warm Reset and reset de-assertion would require host processor to write to respective registers in PRCM.

Figure 6-28 shows the nRESETIN_OUT waveform when using nRESETIN_OUT as warm reset source. For the duration when external warm reset switch is closed, both the device and chip will be driving zero.

Figure 6-28. Warm Reset Sequence (External Warm Reset Source)

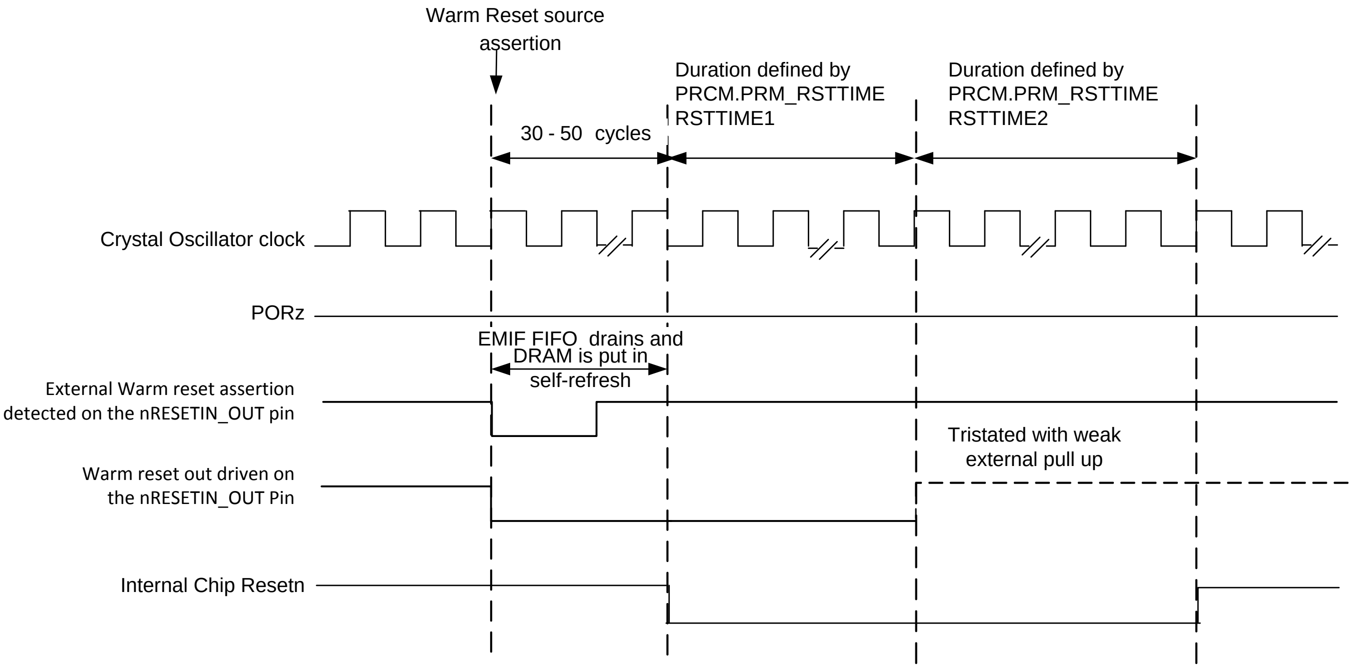
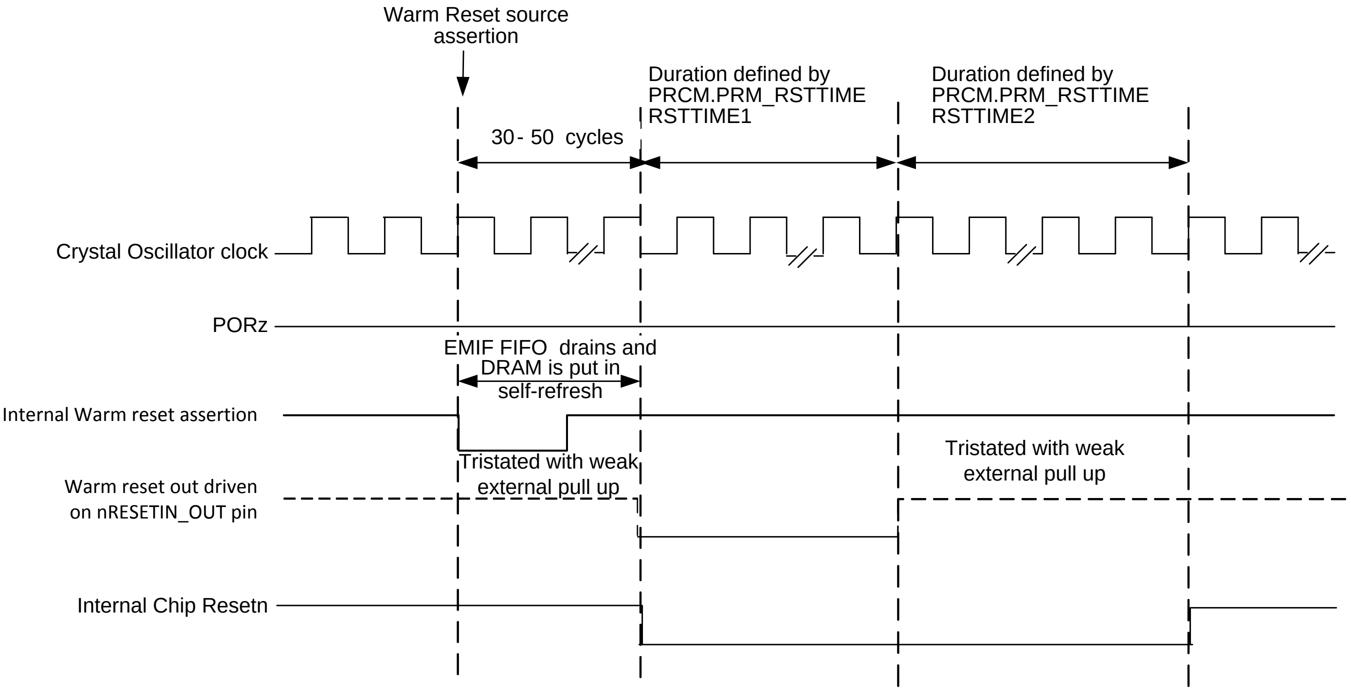


Figure 6-29 shows the nRESETIN_OUT waveform when any one of the warm reset sources captured except using nRESETIN_OUT itself as warm reset source.

Note: PRCM.PRM_RSTTIME1 and PRCM.PRM_RSTTIME2 use a 32-kHz clock derived from the master crystal oscillator clock.

Figure 6-29. Warm Reset Sequence (Internal Warm Reset Source)



For applications not needing a warm reset input function, the warm reset input can be disabled by a software programmable MMR bit. During device boot-up, software can program this MMR bit to mask/disable Warm reset input function.

6.7.4.2 Watchdog Timer (WDT_RST)

WDT_RST is generated from internal watchdog timer modules. Activation is triggered by a timeout event. The reset is not blockable.

6.7.4.3 Global Warm Software Reset (GLOBAL_SW_WARM_RST)

GLOBAL_WARM_SW_RST is internally generated by the PRCM. Activation is triggered upon setting memory-mapped register bit, PRM_RSTCTRL. RST_GLOBAL_WARM_SW. This bit is self-clearing, which means it is automatically cleared by the hardware.

6.7.4.4 Test Reset (TRSTz)

This reset is triggered from TRSTz pin on JTAG interface. This is a non-blockable reset and it resets test and emulation logic.

NOTE: A PORz reset assertion should cause entire device to reset including all test and emulation logic regardless of the state of TRSTz Therefore, PORz assertion will achieve full reset of the device even if TRSTz pin is pulled permanently high and no special toggling of TRSTz pin is required during power ramp to achieve full POR reset to the device. Further, it is acceptable for TRSTz input to be pulled permanently low during normal functional usage of the device in the end-system to ensure that all test and emulation logic is kept in reset.

6.7.5 Reset Characteristics

The following table shows characteristic of each reset source.

Table 6-28. Reset Sources

	Cold Reset Sources			**Warm Reset sources**			
Characteristic	Pin PORz	SW Cold Reset	Bad Device	Pin Warm Reset	Watchdog Timer	SW Warm Reset	TRSTz
Boot pins latched	Y	N	N	N	N	N	N
Resets Standard Efuses	Y	N	N	N	N	N	N
Resets Customer Efuses	Y	Y	Y	Y	Y	Y	N
DRAM contents preserved	N	N	N	N(1)	N(1)	N(1)	Y
Resets PLLs	Y	Y	Y	N	N	N	N
Resets Clock Dividers	Y	Y	Y	N	N	N	N
PLLs enter bypass mode(2)	Y	Y	Y	Y	Y	Y	N
Reset source blockable by emulation	N	N	N	Y	Y	Y	N
Resets test and emulation logic	Y	Y	Y	N	Y	N	Y
Resets Chip Functional Logic	Y	Y	Y	Y	Y	Y	N
Puts IOs in Tri-state	Y	Y	Y	Y	Y	Y	N
Resets Pinmux Registers	Y	Y	Y(3)	Y(3)	Y	Y(3)	N
Reset out Assertion (nRESETIN_OUT Pin)	Y	Y	Y(3)	Y(3)	Y	Y(3)	N

(1) DRAM contents are not guaranteed to be preserved across all Warm resets. The ROM software does not utilize this feature of DRAM content preservation, hence, the processor reboots like a cold boot for warm resets.

(2) CORE PLL is an exception when EMAC switch reset isolation is enabled

(3) Some special IOs/Muxing registers like test, emulation, GEMAC Switch (when under reset isolation mode), and more related will not be affected under warm reset conditions.

6.7.6 EMAC Switch Reset Isolation

The device will support reset isolation for the Ethernet Switch peripheral. This allows the device to undergo a warm reset without disrupting the switch or traffic being routed through the switch during the reset condition.

If configured by registers in the control module (CPSW Reset Isolation Register: RESET_ISO[0]) that EMAC reset isolation is active, then behavior is as follows:

Any warm reset source (except the software warm reset) will be blocked to the EMAC switch logic in the peripheral (the peripheral has ISO_MAIN_ARST_N input to support such isolation) and to PLL (and its control bits) which is sourcing the EMAC switch clocks as required by the peripheral (50- or 125 MHz reference clocks). Also, the EMAC switch related IO pins must retain their pin muxing and not glitch (continuously controlled by the EMAC switch peripheral) by blocking reset to the controlling MMR bits.

If configured by registers in the control module that EMAC reset isolation is NOT active (default state), then the warm reset sources are allowed to propagate as normal including to the EMAC Switch peripheral (both reset inputs to the peripheral).

All cold or POR resets will always propagate to the EMAC switch peripheral as normal (as otherwise defined in this document).

6.7.7 Reset Priority

If more than one of these reset sources are asserted simultaneously then the following priority order should be used:

1. POR
2. TRSTz
3. External warm reset
4. Emulation
5. Reset requestors
6. Software resets

6.7.8 Trace Functionality Across Reset

Other than the Cold Reset Sources shown and TRSTz, no other resets (such as global warm resets and local resets) affect trace functionality. The debug subsystem must implement required reset isolation for the trace logic. The I/Os and muxing control (if any) for trace I/Os should not get affected by any other reset (all DebugSS EMU pins). Since PLLs are reset only on Global Cold Resets and are isolated with other resets (such as global warm resets and local resets), clocks are ensured to be stable.

6.7.9 RTC PORz

This processor supports RTC-only mode by supplying dedicated power to the RTC module. The RTC module has a dedicated PORz signal (RTC_PORz) to reset RTC logic and circuitry during powerup. RTC_PORz is expected to be driven low when the RTC power supply is ramping up. After the power supply reaches its stable value, the RTC_PORz can be de-asserted. The RTC module is not affected by the device PORz. Similarly RTC_PORz does not affect the device reset.

If RTC-only mode is not required, then PORz and RTC_PORz need to be shorted. For power-up sequencing with respect to RTC_PORz, see the device datasheet.

6.8 Power-Up/Down Sequence

Each power domain has a dedicated warm and cold reset. Warm reset gets asserted each time there is any warm reset source requesting a reset. Warm reset is also asserted when the power domain moves from ON to OFF state. Cold reset for the power domain is asserted in response to cold reset sources. When the domain moves from ON to OFF state, then a cold reset also gets asserted as this is similar to a power-up condition for that domain.

6.9 IO State

All IOs except for JTAG i/f and Reset output (and any special cases mentioned in pinlist) should have their output drivers tri-state and internal pulls enabled during assertion of all reset sources. JTAG i/f IO is affected only by TRSTz.

Note: The PRU-ICSS and wakeup processor are held under reset after global warm reset by assertion of software source of reset. Other domains are held under reset after global warm reset until the MPU software enables their respective interface clock.

6.10 Voltage and Power Domains

The following table shows how the device core logic is partitioned into two core logic voltage domains and four power domains. The table lists which voltage and power domain a functional module belongs.

Table 6-29. Core Logic Voltage and Power Domains

Logic Voltage Domain Name	Module
CORE	All Core Modules
RTC	RTC

6.10.1 Voltage Domains

The core logic is divided into two voltage domains: VDD_CORE and VDD_RTC.

6.10.2 Power Domains

In order to reduce power due to leakage, the core logic supply voltage to the power domains can be turned OFF with internal power switches. The internal power switches are controlled through memory mapped registers in the control module.

If all the modules within a power domain are not used, that power domain can be placed in the OFF state.

The following table shows the allowable combination power domain ON/OFF states and which power domains are switched via internal power switches. At power-on-reset, all domains except always-on will be in the power domain OFF state.

Table 6-30. Power Domain State Table

	POWER DOMAIN					
MODE	WAKEUP	MPU	GFX	PER	RTC	EFUSE
No Voltage Supply	N/A	N/A	N/A	N/A	N/A	N/A
Power On Reset	ON	OFF	OFF	OFF	OFF	OFF
ALL OTHER FUNCTIONAL MODES	ON	DON'T CARE	DON'T CARE	DON'T CARE	DON'T CARE	DON'T CARE
Internal Power Switch	NO	YES	YES	YES	YES	YES

6.11 Device Modules and Power Management Attributes List

Table 6-31. Power Domain of Various Modules

Power Supply	Power Domain	Modules OR Supply Destinations (sinks)
VDD_CORE	PD_EMU	Debug Subsystem
	PD_WKUP (Always On)	Wakeup Processor Subsystem
		PRCM
		Control Module
		GPIO0
		DMTimer0
		DMTimer_1ms
		SyncTimer32k
		UART0
		I2C0
		WDT1
		L4_WKUP
		Pinmux
		WKUP_DFTSS
		VDD of crystal oscillator
		RC Oscillator
		ADC0
		VDD of all chip-level Ios
		VDD of ADC0_AFE
		Miscellaneous logic in MPU
		Switch cells from power domains in VDD_CORE

Table 6-31. Power Domain of Various Modules (continued)

Power Supply	Power Domain	Modules OR Supply Destinations (sinks)
VDD_CORE	PD_PER (Switchable)	**Infrastructure**
		L3
		L4_PER, L4_Fast
		EMIF4
		EDMA
		GPMC
		OCMC controller
		L3 / L4_PER / L4_Fast Peripherals
		PRU-ICSS0/1
		DSS
		Ethernet Switch
		USB Controller
		GPMC
		MMC0-2
		IEEE1500
		DMTIMER2-11
		UART1-5
		SPI0-4
		I2C1, 2
		DCAN0, 1
		McASP0, 1
		ePWMSS0-5
		GPIO1-4
		ELM
		Mailbox0-1, Spinlock
		OCP_WP
		VPFE0, 1
		DDR PHY
		HDQ1W
		QSPI
		Others
		DFTSS (Main)
		USB2PHYCORE (VDD/digital section)
		USB2PHYCM (VDD/digital section)
	PD_MPU_EMU (switchable)	MPU Emulation
	PD_PER_PLL (switchable)	PER DPLL digital logic
	PD_DISP_PLL (switchable)	DISP DPLL digital logic
	PD_CORE_PLL (switchable)	CORE DPLL digital logic
	PD_DDR_PLL (switchable)	DDR DPLL digital logic
	PD_GFX (switchable)	SGX530
VDD_MPU	PD_MPU (switchable)	CPU, SCU, PL310, AXI2OCP, MA, Bridges, OCMC, DEBUG_MPU - CTI, CTM, PTM, Timestamp
	PD_AON_MPU	WKUPGEN, STBY_CTRL
	PD_MPU_DPLL (switchable)	MPU DPLL digital logic

Table 6-31. Power Domain of Various Modules (continued)

Power Supply	Power Domain	Modules OR Supply Destinations (sinks)
VDD_RTC	PD_RTC (non-switchable)	RTC
		VDD for 32 768 Hz Crystal Osc
		VDD for IO for the alarm pin

6.11.1 Power Domain Power Down Sequence

The following sequence of steps happen during the power down of a power domain

All peripherals (belonging to a power domain) with STANDBY interface will assert STANDBY. STANDBY assertion should get triggered by a peripheral based on its activity on OCP initiator port. The peripheral should assert STANDBY whenever initiator port is IDLE. Some of the peripherals may not have this feature and they will require SW write to standby-mode register to get STANDBY assertion from the peripheral.

1. SW will request all modules in given power domain to go to disable state by programming module control register inside PRCM.
2. PRCM will start and wait for completion of power management handshake with peripherals (IdleReq/IdleAck).
3. PRCM will gate-off all the clocks to the power domain.
4. SW will request all clock domains in given power domain to go to “force sleep” mode by programming functional clock domain register in PRCM. Note that PRCM has already gated-off clocks and this register programming may look redundant.
5. SW will request PRCM to take this power domain to OFF state by programming PWRSTCTRL register. Note that this step can be skipped if PWRSTCTRL is permanently programmed to OFF state. When this is done, functional clock domain register decides when power domain will be taken to OFF state. Only reason not to have OFF state in PWRSTCTRL is to take power domain to just clock gate state without power gating.
6. PSCON specific to this power domain will assert isolation enable for the domain.
7. PRCM will assert warm and cold reset to the power domain.
8. PSCON will assert control signals to switch-off power using on-die switches.
9. On-die switches will send acknowledge back to PSCON.

6.11.2 Power Domain Power-Up Sequence

The following sequence of steps occurs during power-up of a power domain. This sequence is not relevant to always-on domain as this domain will never go to OFF state as long as the device is powered. This sequence will be repeated each time a domain is taken to ON state from OFF (including first time power-up). Note that some of the details are intentionally taken out here to simplify things.

There can be multiple reasons to start power-up sequence for a domain. For example it can be due to an interrupt from one of the peripherals which is powered-up.

1. SW will request required clock domains inside this power domain to go to force wake-up state by programming functional clock domain register.
2. PRCM will enable clocks to the required clock domains.
3. PSCON specific to this power domain will assert control signal to un-gate the power.
4. Once power is un-gated, on die switches will send acknowledge back to PSCON.
5. PRCM will de-assert cold and warm reset to the power domain.
6. PRCM will turn-off isolation cells.
7. SW will request PRCM to enable required module in the power domain by programming module control register.
8. PRCM will initiate and wait for completion of PM protocol to enable the modules (IdleReq/IdleAck).

6.12 Power Management Registers

6.12.1 PRCM_PRM_CEFUSE Registers

Table 6-32 lists the memory-mapped registers for the PRCM_PRM_CEFUSE. All register offset addresses not listed in Table 6-32 should be considered as reserved locations and the register contents should not be modified.

Table 6-32. PRCM_PRM_CEFUSE REGISTERS

Offset	Acronym	Register Name	Section
0h	PRCM_PM_CEFUSE_PWRSTCTRL		Section 6.12.1.1
4h	PRCM_PM_CEFUSE_PWRSTST		Section 6.12.1.2
24h	PRCM_RM_CEFUSE_CONTEXT		Section 6.12.1.3

6.12.1.1 PRCM_PM_CEFUSE_PWRSTCTRL Register (offset = 0h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_PM_CEFUSE_PWRSTCTRL is shown in Figure 6-30 and described in Table 6-33.

This register controls the CEFUSE power state to reach upon a domain sleep transition

Figure 6-30. PRCM_PM_CEFUSE_PWRSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED			LOWPOWERSTATECHANGE	RESERVED		POWERSTATE	
Rreturns0s-0h			R/WSpecial-0h	Rreturns0s-0h		R/W-0h	

Table 6-33. PRCM_PM_CEFUSE_PWRSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	Rreturns0s	0h	
4	LOWPOWERSTATECHANGE	R/WSpecial	0h	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0h (R/W) = Do not request a low power state change. 1h (R/W) = Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.
3-2	RESERVED	Rreturns0s	0h	
1-0	POWERSTATE	R/W	0h	Power state control 0h (R/W) = OFF state 1h (R/W) = RESERVED 2h (R/W) = RESERVED 3h (R/W) = ON State

6.12.1.2 PRCM_PM_CEFUSE_PWRSTST Register (offset = 4h) [reset = 7h]

Register mask: FFFFFFFFh

PRCM_PM_CEFUSE_PWRSTST is shown in Figure 6-31 and described in Table 6-34.

This register provides a status on the current CEFUSE power domain state. [warm reset insensitive]

Figure 6-31. PRCM_PM_CEFUSE_PWRSTST Register

31	30	29	28	27	26	25	24
RESERVED						LASTPOWERSTATEENTERED	
Rreturns0s-0h						R/W1toSet-0h	

23	22	21	20	19	18	17	16
RESERVED			INTRANSITION	RESERVED			
Rreturns0s-0h			R-0h	Rreturns0s-0h			

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED					LOGICSTATEST	POWERSTATEST	
Rreturns0s-0h					R-1h	R-3h	

Table 6-34. PRCM_PM_CEFUSE_PWRSTST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	Rreturns0s	0h	
25-24	LASTPOWERSTATEENTERED	R/W1toSet	0h	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0h (R) = Power domain was previously OFF 1h (R) = Power domain was previously ON-ACTIVE
23-21	RESERVED	Rreturns0s	0h	
20	INTRANSITION	R	0h	Domain transition status 0h (R) = No on-going transition on power domain 1h (R) = Power domain transition is in progress.
19-3	RESERVED	Rreturns0s	0h	
2	LOGICSTATEST	R	1h	Logic state status 0h (R) = Logic in domain is OFF 1h (R) = Logic in domain is ON
1-0	POWERSTATEST	R	3h	Current power state status 0h (R) = Power domain is OFF 3h (R) = Power domain is ON-ACTIVE

6.12.1.3 PRCM_RM_CEFUSE_CONTEXT Register (offset = 24h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_CEFUSE_CONTEXT is shown in Figure 6-32 and described in Table 6-35.

This register contains dedicated CEFUSE module context statuses. [warm reset insensitive]

Figure 6-32. PRCM_RM_CEFUSE_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-35. PRCM_RM_CEFUSE_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CUST_EFUSE_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.2 PRCM_PRM_DEVICE Registers

Table 6-36 lists the memory-mapped registers for the PRCM_PRM_DEVICE. All register offset addresses not listed in Table 6-36 should be considered as reserved locations and the register contents should not be modified.

Table 6-36. PRCM_PRM_DEVICE REGISTERS

Offset	Acronym	Register Name	Section
0h	PRCM_PRM_RSTCTRL		Section 6.12.2.1
4h	PRCM_PRM_RSTST		Section 6.12.2.2
8h	PRCM_PRM_RSTTIME		Section 6.12.2.3
Ch	PRCM_PRM_SRAM_COUNT		Section 6.12.2.4
10h	PRCM_PRM_LDO_SRAM_CORE_SETUP		Section 6.12.2.5
14h	PRCM_PRM_LDO_SRAM_CORE_CTRL		Section 6.12.2.6
18h	PRCM_PRM_LDO_SRAM_MPU_SETUP		Section 6.12.2.7
1Ch	PRCM_PRM_LDO_SRAM_MPU_CTRL		Section 6.12.2.8
20h	PRCM_PRM_IO_COUNT		Section 6.12.2.9
24h	PRCM_PRM_IO_PMCTRL		Section 6.12.2.10
28h	PRCM_PRM_VC_VAL_BYPASS		Section 6.12.2.11

Table 6-36. PRCM_PRM_DEVICE REGISTERS (continued)

Offset	Acronym	Register Name	Section
30h	PRCM_PRM_EMIF_CTRL		Section 6.12.2.12

6.12.2.1 PRCM_PRM_RSTCTRL Register (offset = 0h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_PRM_RSTCTRL is shown in Figure 6-33 and described in Table 6-37.

Global software cold and warm reset control. This register is auto-cleared. Only write 1 is possible. A read returns 0 only.

Figure 6-33. PRCM_PRM_RSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED						RST_GLOBAL_COLD_SW	RST_GLOBAL_WARM_SW
Rreturns0s-0h						Rreturns0s/W-0h	Rreturns0s/W-0h

Table 6-37. PRCM_PRM_RSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	Rreturns0s	0h	
1	RST_GLOBAL_COLD_SW	Rreturns0s/W	0h	Global COLD software reset control. This bit is reset only upon a global cold source of reset. 0h (R/W) = 0X0 : Global COLD software reset is cleared. 1h (R/W) = 0X1 : Asserts a global COLD software reset. The software must ensure the SDRAM is properly put in sef-refresh mode before applying this reset.
0	RST_GLOBAL_WARM_SW	Rreturns0s/W	0h	Global WARM software reset control. This bit is reset upon any global source of reset (warm and cold). 0h (R/W) = 0X0 : Global warm software reset is cleared. 1h (R/W) = 0X1 : Asserts a global warm software reset.

6.12.2.2 PRCM_PRM_RSTST Register (offset = 4h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_PRM_RSTST is shown in Figure 6-34 and described in Table 6-38.

This register logs the global reset sources. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]

Figure 6-34. PRCM_PRM_RSTST Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED						ICEPICK_RST	RESERVED
Rreturns0s-0h						R/W1toClr-0h	Rreturns0s-0h

7	6	5	4	3	2	1	0
RESERVED		EXTERNAL_WARM_RST	WDT1_RST	RESERVED		GLOBAL_WARM_SW_RST	GLOBAL_COLD_RST
Rreturns0s-0h		R/W1toClr-0h	R/W1toClr-0h	R/W-0h		R/W1toClr-0h	R/W1toClr-1h

Table 6-38. PRCM_PRM_RSTST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	Rreturns0s	0h	
9	ICEPICK_RST	R/W1toClr	0h	IcePick reset event. This is a source of global warm reset initiated by the emulation. [warm reset insensitive] 0h (R/W) = 0X0 : No ICEPICK reset. 1h (R/W) = 0X1 : IcePick reset has occurred.
8-6	RESERVED	Rreturns0s	0h	
5	EXTERNAL_WARM_RST	R/W1toClr	0h	External warm reset event [warm reset insensitive] 0h (R/W) = 0X0 : No global warm reset. 1h (R/W) = 0X1 : Global external warm reset has occurred.
4	WDT1_RST	R/W1toClr	0h	Watchdog1 timer reset event. This is a source of global WARM reset. [warm reset insensitive] 0h (R/W) = 0X0 : No watchdog reset. 1h (R/W) = 0X1 : watchdog reset has occurred.
3-2	RESERVED	R/W	0h	
1	GLOBAL_WARM_SW_RST	R/W1toClr	0h	Global warm software reset event [warm reset insensitive] 0h (R/W) = 0X0 : No global warm SW reset 1h (R/W) = 0X1 : Global warm SW reset has occurred.
0	GLOBAL_COLD_RST	R/W1toClr	1h	Power-on (cold) reset event [warm reset insensitive] 0h (R/W) = 0X0 : No power-on reset. 1h (R/W) = 0X1 : Power-on reset has occurred.

6.12.2.3 PRCM_PRM_RSTTIME Register (offset = 8h) [reset = 4006h]

Register mask: FFFFFFFFh

PRCM_PRM_RSTTIME is shown in Figure 6-35 and described in Table 6-39.

Reset duration control. [warm reset insensitive]

Figure 6-35. PRCM_PRM_RSTTIME Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
Rreturns0s-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RSTTIME2					RSTTIME1									
Rreturns0s-0h	R/W-10h					R/W-6h									

Table 6-39. PRCM_PRM_RSTTIME Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	Rreturns0s	0h	
14-10	RSTTIME2	R/W	10h	(Power domain) reset duration 2 (number of RM.SYSCLK clock cycles)
9-0	RSTTIME1	R/W	6h	(Global) reset duration 1 (number of SYS_CLK clock cycles)

6.12.2.4 PRCM_PRM_SRAM_COUNT Register (offset = Ch) [reset = 78000017h]

Register mask: FFFFFFFFh

PRCM_PRM_SRAM_COUNT is shown in Figure 6-36 and described in Table 6-40.

Common setup for SRAM LDO transition counters. Applies to all voltage domains. [warm reset insensitive]

Figure 6-36. PRCM_PRM_SRAM_COUNT Register

31	30	29	28	27	26	25	24
STARTUP_COUNT							
R/W-78h							

23	22	21	20	19	18	17	16
SLPCNT_VALUE							
R/W-0h							

15	14	13	12	11	10	9	8
VSETUPCNT_VALUE							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED		PCHARGECNT_VALUE					
Rreturns0s-0h		R/W-17h					

Table 6-40. PRCM_PRM_SRAM_COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	STARTUP_COUNT	R/W	78h	Determines the start-up duration of SRAM and ABB LDO. The duration is computed as 16 x NbCycles of system clock cycles. Target is 50us.
23-16	SLPCNT_VALUE	R/W	0h	Delay between retention/off assertion of last SRAM bank and SRAMALLRET signal to LDO is driven high. Counting on system clock. Target is 2us.
15-8	VSETUPCNT_VALUE	R/W	0h	SRAM LDO rampup time from retention to active mode. The duration is computed as 8 x NbCycles of system clock cycles. Target is 30us.
7-6	RESERVED	Rreturns0s	0h	
5-0	PCHARGECNT_VALUE	R/W	17h	Delay between de-assertion of standby_rta_ret_on and standby_rta_ret_good. Counting on system clock. Target is 600ns.

6.12.2.5 PRCM_PRM_LDO_SRAM_CORE_SETUP Register (offset = 10h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_PRM_LDO_SRAM_CORE_SETUP is shown in Figure 6-37 and described in Table 6-41.

Setup of the SRAM LDO for CORE voltage domain. [warm reset insensitive]

Figure 6-37. PRCM_PRM_LDO_SRAM_CORE_SETUP Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							AIPOFF
Rreturns0s-0h							R/W-0h

7	6	5	4	3	2	1	0
ENFUNC5	ENFUNC4	ENFUNC3_EXPORT	ENFUNC2_EXPORT	ENFUNC1_EXPORT	ABBOFF_SLEEP_EXPORT	ABBOFF_ACT_EXPORT	DISABLE_RTA_EXPORT
R/W-0h	R/W-0h	R/WSpecial-0h	R/WSpecial-0h	R/WSpecial-0h	R/WSpecial-0h	R/WSpecial-0h	R/WSpecial-1h

Table 6-41. PRCM_PRM_LDO_SRAM_CORE_SETUP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	AIPOFF	R/W	0h	Override on AIPOFF input of SRAM LDO. 0h (R/W) = AIPOFF signal is not overriden 1h (R/W) = AIPOFF signal is overriden to '1'. Corresponding SRAM LDO is disabled and in HZ mode.
7	ENFUNC5	R/W	0h	ENFUNC5 input of SRAM LDO. 0h (R/W) = Active to retention is a one step transfer 1h (R/W) = Active to retention is a two steps transfer
6	ENFUNC4	R/W	0h	ENFUNC4 input of SRAM LDO. 0h (R/W) = One external clock is supplied 1h (R/W) = No external clock is supplied
5	ENFUNC3_EXPORT	R/WSpecial	0h	ENFUNC3 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0h (R/W) = Sub regulation is disabled 1h (R/W) = Sub regulation is enabled
4	ENFUNC2_EXPORT	R/WSpecial	0h	ENFUNC2 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0h (R/W) = External cap is used 1h (R/W) = External cap is not used
3	ENFUNC1_EXPORT	R/WSpecial	0h	ENFUNC1 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0h (R/W) = Short circuit protection is disabled 1h (R/W) = Short circuit protection is enabled

Table 6-41. PRCM_PRM_LDO_SRAM_CORE_SETUP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	ABBOFF_SLEEP_EXPORT	R/WSpecial	0h	Determines whether SRAMNWA is supplied by VDDS or VDDAR during deep-sleep. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0h (R/W) = SRAMNWA supplied with VDDS 1h (R/W) = SRAMNWA supplied with VDDAR
1	ABBOFF_ACT_EXPORT	R/WSpecial	0h	Determines whether SRAMNWA is supplied by VDDS or VDDAR during active mode. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0h (R/W) = SRAMNWA supplied with VDDS 1h (R/W) = SRAMNWA supplied with VDDAR
0	DISABLE_RTA_EXPORT	R/WSpecial	1h	Control for HD memory RTA feature. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. Note : This feature is not used. 0h (R/W) = HD memory RTA feature is enabled 1h (R/W) = HD memory RTA feature is disabled

6.12.2.6 PRCM_PRM_LDO_SRAM_CORE_CTRL Register (offset = 14h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_PRM_LDO_SRAM_CORE_CTRL is shown in Figure 6-38 and described in Table 6-42.

Control and status of the SRAM LDO for CORE voltage domain. [warm reset insensitive]

Figure 6-38. PRCM_PRM_LDO_SRAM_CORE_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED						SRAM_IN_TRANSITION	SRAMLDO_STS
Rreturns0s-0h						R-0h	R-0h

7	6	5	4	3	2	1	0
RESERVED							RETMODE_EN
Rreturns0s-0h							R/W-0h

Table 6-42. PRCM_PRM_LDO_SRAM_CORE_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	Rreturns0s	0h	
9	SRAM_IN_TRANSITION	R	0h	Status indicating SRAM LDO state machine state. 0h (R) = SRAM LDO state machine is stable 1h (R) = SRAM LDO state machine is in transition state
8	SRAMLDO_STS	R	0h	SRAMLDO status 0h (R) = SRAMLDO is in ACTIVE mode. 1h (R) = SRAMLDO is on RETENTION mode.
7-1	RESERVED	Rreturns0s	0h	
0	RETMODE_EN	R/W	0h	Control if the SRAM LDO retention mode is used or not. 0h (R/W) = SRAM LDO is not allowed to go to RET mode 1h (R/W) = SRAM LDO go to RET mode when all memory of voltage domain are OFF or RET

6.12.2.7 PRCM_PRM_LDO_SRAM_MPU_SETUP Register (offset = 18h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_PRM_LDO_SRAM_MPU_SETUP is shown in Figure 6-39 and described in Table 6-43.

Setup of the SRAM LDO for MPU voltage domain. [warm reset insensitive]

Figure 6-39. PRCM_PRM_LDO_SRAM_MPU_SETUP Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							AIPOFF
Rreturns0s-0h							R/W-0h

7	6	5	4	3	2	1	0
ENFUNC5	ENFUNC4	ENFUNC3_EXPORT	ENFUNC2_EXPORT	ENFUNC1_EXPORT	ABBOFF_SLEEP_EXPORT	ABBOFF_ACT_EXPORT	DISABLE_RTA_EXPORT
R/W-0h	R/W-0h	R/WSpecial-0h	R/WSpecial-0h	R/WSpecial-0h	R/WSpecial-0h	R/WSpecial-0h	R/WSpecial-0h

Table 6-43. PRCM_PRM_LDO_SRAM_MPU_SETUP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	AIPOFF	R/W	0h	Override on AIPOFF input of SRAM LDO. 0h (R/W) = AIPOFF signal is not overriden 1h (R/W) = AIPOFF signal is overriden to '1'. Corresponding SRAM LDO is disabled and in HZ mode.
7	ENFUNC5	R/W	0h	ENFUNC5 input of SRAM LDO. 0h (R/W) = Active to retention is a one step transfer 1h (R/W) = Active to retention is a two steps transfer
6	ENFUNC4	R/W	0h	ENFUNC4 input of SRAM LDO. 0h (R/W) = One external clock is supplied 1h (R/W) = No external clock is supplied
5	ENFUNC3_EXPORT	R/WSpecial	0h	ENFUNC3 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0h (R/W) = Sub regulation is disabled 1h (R/W) = Sub regulation is enabled
4	ENFUNC2_EXPORT	R/WSpecial	0h	ENFUNC2 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0h (R/W) = External cap is used 1h (R/W) = External cap is not used
3	ENFUNC1_EXPORT	R/WSpecial	0h	ENFUNC1 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0h (R/W) = Short circuit protection is disabled 1h (R/W) = Short circuit protection is enabled

Table 6-43. PRCM_PRM_LDO_SRAM_MPU_SETUP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	ABBOFF_SLEEP_EXPORT	R/WSpecial	0h	Determines whether SRAMNWA is supplied by VDDS or VDDAR during deep-sleep. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0h (R/W) = SRAMNWA supplied with VDDS 1h (R/W) = SRAMNWA supplied with VDDAR
1	ABBOFF_ACT_EXPORT	R/WSpecial	0h	Determines whether SRAMNWA is supplied by VDDS or VDDAR during active mode. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0h (R/W) = SRAMNWA supplied with VDDS 1h (R/W) = SRAMNWA supplied with VDDAR
0	DISABLE_RTA_EXPORT	R/WSpecial	0h	Control for HD memory RTA feature. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0h (R/W) = HD memory RTA feature is enabled 1h (R/W) = HD memory RTA feature is disabled

6.12.2.8 PRCM_PRM_LDO_SRAM_MPU_CTRL Register (offset = 1Ch) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_PRM_LDO_SRAM_MPU_CTRL is shown in Figure 6-40 and described in Table 6-44.

Control and status of the SRAM LDO for MPU voltage domain. [warm reset insensitive]

Figure 6-40. PRCM_PRM_LDO_SRAM_MPU_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED						SRAM_IN_TRANSITION	SRAMLDO_STS
Rreturns0s-0h						R-0h	R-0h

7	6	5	4	3	2	1	0
RESERVED							RETMODE_EN
Rreturns0s-0h							R/W-0h

Table 6-44. PRCM_PRM_LDO_SRAM_MPU_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	Rreturns0s	0h	
9	SRAM_IN_TRANSITION	R	0h	Status indicating SRAM LDO state machine state. 0h (R) = SRAM LDO state machine is stable 1h (R) = SRAM LDO state machine is in transition state
8	SRAMLDO_STS	R	0h	SRAMLDO status 0h (R) = SRAMLDO is in ACTIVE mode. 1h (R) = SRAMLDO is on RETENTION mode.
7-1	RESERVED	Rreturns0s	0h	
0	RETMODE_EN	R/W	0h	Control if the SRAM LDO retention mode is used or not. 0h (R/W) = SRAM LDO is not allowed to go to RET mode 1h (R/W) = SRAM LDO go to RET mode when all memory of voltage domain are OFF or RET

6.12.2.9 PRCM_PRM_IO_COUNT Register (offset = 20h) [reset = 3Ah]

Register mask: FFFFFFFFh

PRCM_PRM_IO_COUNT is shown in Figure 6-41 and described in Table 6-45.

This register allows controlling EMIF IO isolation removal setup. [warm reset insensitive]

Figure 6-41. PRCM_PRM_IO_COUNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
Rreturns0s-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ISO_2_ON_TIME							
Rreturns0s-0h								R/W-3Ah							

Table 6-45. PRCM_PRM_IO_COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	Rreturns0s	0h	
7-0	ISO_2_ON_TIME	R/W	3Ah	Determines the setup time of the DDR IOs going out of isolation. Counting on the system clock. Target is 1.5us.

6.12.2.10 PRCM_PRM_IO_PMCTRL Register (offset = 24h) [reset = 20h]

Register mask: FFFFFFFFh

PRCM_PRM_IO_PMCTRL is shown in Figure 6-42 and described in Table 6-46.

This register allows controlling power management features of the IOs.

Figure 6-42. PRCM_PRM_IO_PMCTRL Register

31	30	29	28	27	26	25	24
RESERVED						IO_ISO_STS	IO_ISO_CTRL
Rreturns0s-0h						R-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED							GLOBAL_WUEN
Rreturns0s-0h							R/W-0h

15	14	13	12	11	10	9	8
RESERVED						WUCLK_STS	WUCLK_CTRL
Rreturns0s-0h						R-0h	R/W-0h

7	6	5	4	3	2	1	0
RESERVED		IO_ON_STS	ISOOVR_EXTEND	RESERVED		ISOCLK_STS	ISOCLK_OVERRIDE
Rreturns0s-0h		R-1h	R/W-0h	Rreturns0s-0h		R-0h	R/W-0h

Table 6-46. PRCM_PRM_IO_PMCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	Rreturns0s	0h	
25	IO_ISO_STS	R	0h	IO ISO Status. 0h (R) = IO isolation not active. 1h (R) = IO isolation active
24	IO_ISO_CTRL	R/W	0h	IO ISO control. writing this bit to '1' will kick off IO isolation. 0h (R/W) = Turn off the IO isolation. 1h (R/W) = Turn ON the IO isolation
23-17	RESERVED	Rreturns0s	0h	
16	GLOBAL_WUEN	R/W	0h	Global IO wakeup enable. This is a gating condition to all individual IO WUEN coming from control module. Gating is done in the Spinner logic. 0h (R/W) = All individual IO WUEN are gated in the Spinner logic (overriden to 0). 1h (R/W) = All individual IO WUEN from control module are going to IOs.
15-10	RESERVED	Rreturns0s	0h	
9	WUCLK_STS	R	0h	Gives value of WUCLKOUT signal coming back from IO pad ring.
8	WUCLK_CTRL	R/W	0h	Direct control on WUCLKIN signal to IO pad ring. 0h (R/W) = WUCLKIN signal is driven to 0. IO wakeup daisy chain is functional as well as IO whose wakeup feature is enabled. 1h (R/W) = WUCLKIN signal is driven to 1. IO wakeup daisy chain is reset and is latching current pad states and WUEN inputs.
7-6	RESERVED	Rreturns0s	0h	
5	IO_ON_STS	R	1h	Gives the functional status of the IO ring. 0h (R) = Part or all of the IOs are not in the ON state, that is are in isolation state. 1h (R) = All IOs are in the ON state.

Table 6-46. PRCM_PRM_IO_PMCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ISOOVR_EXTEND	R/W	0h	Control non-EMIF IO isolation extension. 0h (R/W) = Non-EMIF IO isolation is not extended. "EMIF_ON" IO transition happens as soon as automatic restore is completed. 1h (R/W) = Non-EMIF IO isolation is extended. "EMIF_ON" IO transition is stalled.
3-2	RESERVED	Rreturns0s	0h	
1	ISOCLK_STS	R	0h	Gives value of ISOCLKOUT signal coming back from IO pad ring.
0	ISOCLK_OVERRIDE	R/W	0h	Override control on ISOCLKIN signal to IO pad ring.When not overriden, this signal is controlled by hardware only. 0h (R/W) = ISOCLKIN signal is not overriden. 1h (R/W) = ISOCLKIN signal is overriden to active value ('1').

6.12.2.11 PRCM_PRM_VC_VAL_BYPASS Register (offset = 28h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_PRM_VC_VAL_BYPASS is shown in Figure 6-43 and described in Table 6-47.

This MMR has flag to indicate OPP change to EMIF to allow read/write leveling.

Figure 6-43. PRCM_PRM_VC_VAL_BYPASS Register

31	30	29	28	27	26	25	24
RESERVED						OPP_CHANGE_EMIF_LVL	RESERVED
Rreturns0s-0h						R/W-0h	Rreturns0s-0h

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							
Rreturns0s-0h							

Table 6-47. PRCM_PRM_VC_VAL_BYPASS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	Rreturns0s	0h	
25	OPP_CHANGE_EMIF_LVL	R/W	0h	This bit controls read-write leveling of EMIF memories (DDR3). It must be set in case OPP voltage change is done. 0h (R/W) = Enable leveling 1h (R/W) = disable leveling
24-0	RESERVED	Rreturns0s	0h	

6.12.2.12 PRCM_PRM_EMIF_CTRL Register (offset = 30h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_PRM_EMIF_CTRL is shown in Figure 6-44 and described in Table 6-48.

This register controls EMIF controller low power configurations.

Figure 6-44. PRCM_PRM_EMIF_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							EMIF_DEVOFF
Rreturns0s-0h							R/W-0h

Table 6-48. PRCM_PRM_EMIF_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	Reserved
0	EMIF_DEVOFF	R/W	0h	EMIF Controller DeepSleep Mode Enable. This bit should be programmed to '1' (ON) before going into DeepSleep. Must be cleared to '0' (OFF) after wakeup and EMIF configuration is completed. 0h (R/W) = EMIF Controller DeepSleep Mode is Disabled. 1h (R/W) = EMIF Controller DeepSleep Mode is Enabled.

6.12.3 PRCM_PRM_GFX Registers

Table 6-49 lists the memory-mapped registers for the PRCM_PRM_GFX. All register offset addresses not listed in Table 6-49 should be considered as reserved locations and the register contents should not be modified.

Table 6-49. PRCM_PRM_GFX REGISTERS

Offset	Acronym	Register Name	Section
0h	PRCM_PRM_PM_GFX_PWRSTCTRL		Section 6.12.3.1
4h	PRCM_PRM_PM_GFX_PWRSTST		Section 6.12.3.2
10h	PRCM_PRM_RM_GFX_RSTCTRL		Section 6.12.3.3
14h	PRCM_PRM_RM_GFX_RSTST		Section 6.12.3.4
24h	PRCM_PRM_RM_GFX_CONTEXT		Section 6.12.3.5

6.12.3.1 PRCM_PRM_PM_GFX_PWRSTCTRL Register (offset = 0h) [reset = 30100h]

Register mask: FFFFFFFFh

PRCM_PRM_PM_GFX_PWRSTCTRL is shown in Figure 6-45 and described in Table 6-50.

This register controls the GFX power state to reach upon a domain sleep transition.

Figure 6-45. PRCM_PRM_PM_GFX_PWRSTCTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">Rreturns0s-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">GFX_MEM_ONSTATE</td></tr>
<tr><td colspan="6">Rreturns0s-0h</td><td colspan="2">Rreturns1s-3h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>GFX_MEM_RETSTATE</td></tr>
<tr><td colspan="7">Rreturns0s-0h</td><td>R/W-1h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">RESERVED</td><td>LOWPOWERSTATECHANGE</td><td colspan="2">RESERVED</td><td colspan="2">POWERSTATE</td></tr>
<tr><td colspan="3">Rreturns0s-0h</td><td>R/WSpecial-0h</td><td colspan="2">Rreturns0s-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 6-50. PRCM_PRM_PM_GFX_PWRSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	Rreturns0s	0h	
17-16	GFX_MEM_ONSTATE	Rreturns1s	3h	GFX memory state when domain is ON. 3h (R) = Memory bank is on when the domain is ON.
15-9	RESERVED	Rreturns0s	0h	
8	GFX_MEM_RETSTATE	R/W	1h	GFX_MEM bank state when domain is retention 0h (R/W) = Memory is off when the domain is in the RETENTION state. 1h (R/W) = Memory is retained when domain is in RETENTION state.
7-5	RESERVED	Rreturns0s	0h	
4	LOWPOWERSTATECHANGE	R/WSpecial	0h	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0h (R/W) = Do not request a low power state change. 1h (R/W) = Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.
3-2	RESERVED	Rreturns0s	0h	
1-0	POWERSTATE	R/W	0h	Power state control 0h (R/W) = OFF State 1h (R) = RESERVED 2h (R) = RESERVED 3h (R/W) = ON State

6.12.3.2 PRCM_PRM_PM_GFX_PWRSTST Register (offset = 4h) [reset = 17h]

Register mask: FFFFFFFFh

PRCM_PRM_PM_GFX_PWRSTST is shown in Figure 6-46 and described in Table 6-51.

This register provides a status on the current GFX power domain state. [warm reset insensitive]

Figure 6-46. PRCM_PRM_PM_GFX_PWRSTST Register

31	30	29	28	27	26	25	24
RESERVED						LASTPOWERSTATEENTERED	
Rreturns0s-0h						R/W1toSet-0h	

23	22	21	20	19	18	17	16
RESERVED			INTRANSITION	RESERVED			
Rreturns0s-0h			R-0h	Rreturns0s-0h			

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED		GFX_MEM_STATEST		RESERVED	LOGICSTATEST	POWERSTATEST	
Rreturns0s-0h		R-1h		Rreturns0s-0h	R-1h	R-3h	

Table 6-51. PRCM_PRM_PM_GFX_PWRSTST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	Rreturns0s	0h	
25-24	LASTPOWERSTATEENTERED	R/W1toSet	0h	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0h (R) = Power domain was previously OFF 1h (R) = Power domain was previously in RETENTION 2h (R) = Power domain was previously ON-INACTIVE 3h (R) = Power domain was previously ON-ACTIVE
23-21	RESERVED	Rreturns0s	0h	
20	INTRANSITION	R	0h	Domain transition status 0h (R) = No on-going transition on power domain 1h (R) = Power domain transition is in progress.
19-6	RESERVED	Rreturns0s	0h	
5-4	GFX_MEM_STATEST	R	1h	GFX memory state status 0h (R) = Memory is OFF 1h (R) = Memory is in RETENTION. 2h (R) = Reserved 3h (R) = Memory is ON
3	RESERVED	Rreturns0s	0h	
2	LOGICSTATEST	R	1h	Logic state status 0h (R) = Logic in domain is OFF 1h (R) = Logic in domain is ON
1-0	POWERSTATEST	R	3h	Current Power State Status 0h (R) = OFF State [warm reset insensitive] 3h (R) = ON State [warm reset insensitive]

6.12.3.3 PRCM_PRM_RM_GFX_RSTCTRL Register (offset = 10h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_PRM_RM_GFX_RSTCTRL is shown in Figure 6-47 and described in Table 6-52.

This register controls the release of the GFX Domain resets.

Figure 6-47. PRCM_PRM_RM_GFX_RSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							
15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED		RESERVED	GFX_RST
Rreturns0s-0h				Rreturns0s-0h		Rreturns0s-0h	R/W-1h

Table 6-52. PRCM_PRM_RM_GFX_RSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	Rreturns0s	0h	
3-2	RESERVED	Rreturns0s	0h	
1	RESERVED	Rreturns0s	0h	
0	GFX_RST	R/W	1h	GFX domain local reset control 0h (R/W) = Reset is cleared for the GFX Domain (SGX530) 1h (R/W) = Reset is asserted for the GFX Domain (SGX 530)

6.12.3.4 PRCM_PRM_RM_GFX_RSTST Register (offset = 14h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_PRM_RM_GFX_RSTST is shown in Figure 6-48 and described in Table 6-53.

This register logs the different reset sources of the GFX domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]

Figure 6-48. PRCM_PRM_RM_GFX_RSTST Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED				RESERVED			
Rreturns0s-0h				Rreturns0s-0h			

7	6	5	4	3	2	1	0
RESERVED						RESERVED	GFX_RST
Rreturns0s-0h						Rreturns0s-0h	R/W1toClr-0h

Table 6-53. PRCM_PRM_RM_GFX_RSTST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	Rreturns0s	0h	
11-2	RESERVED	Rreturns0s	0h	
1	RESERVED	Rreturns0s	0h	
0	GFX_RST	R/W1toClr	0h	GFX Domain Logic Reset 0h (R/W) = No SW reset occured 1h (R/W) = GFX Domain Logic has been reset upon SW reset

6.12.3.5 PRCM_PRM_RM_GFX_CONTEXT Register (offset = 24h) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_PRM_RM_GFX_CONTEXT is shown in Figure 6-49 and described in Table 6-54.

This register contains dedicated GFX context statuses. [warm reset insensitive]

Figure 6-49. PRCM_PRM_RM_GFX_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_GFX_MEM
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-54. PRCM_PRM_RM_GFX_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_GFX_MEM	R/W1toClr	1h	Specify if memory-based context in GFX_MEM memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of GFX_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.4 PRM_MPU Registers

Table 6-55 lists the memory-mapped registers for the PRM_MPU. All register offset addresses not listed in Table 6-55 should be considered as reserved locations and the register contents should not be modified.

Table 6-55. PRM_MPU Registers

Offset	Acronym	Register Name	Section
0h	PRCM_PM_MPU_PWRSTCTRL		Section 6.12.4.1
4h	PRCM_PM_MPU_PWRSTST		Section 6.12.4.2
14h	PRCM_RM_MPU_RSTST		Section 6.12.4.3
24h	PRCM_RM_MPU_CONTEXT		Section 6.12.4.4

6.12.4.1 PRCM_PM_MPU_PWRSTCTRL Register (offset = 0h) [reset = 3F0707h]

Register mask: FFFFFFFFh

PRCM_PM_MPU_PWRSTCTRL is shown in Figure 6-50 and described in Table 6-56.

This register controls the MPU power state to reach upon mpu domain sleep transition

Figure 6-50. PRCM_PM_MPU_PWRSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	22	21	20	19	18	17	16
RESERVED		MPU_L2_ONSTATE		MPU_L1_ONSTATE		MPU_RAM_ONSTATE	
Rreturns0s-0h		Rreturns1s-3h		Rreturns1s-3h		Rreturns1s-3h	
15	14	13	12	11	10	9	8
RESERVED					MPU_L2_RETSTATE	MPU_L1_RETSTATE	MPU_RAM_RETSTATE
R-0h					R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
RESERVED			LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE	
Rreturns0s-0h			R/WSpecial-0h	Rreturns0s-0h	R/W-1h	R/W-3h	

Table 6-56. PRCM_PM_MPU_PWRSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	Rreturns0s	0h	
21-20	MPU_L2_ONSTATE	Rreturns1s	3h	Default power domain memory state when domain is ON. 3h (R) = Memory bank is on when the domain is ON.
19-18	MPU_L1_ONSTATE	Rreturns1s	3h	Default power domain memory state when domain is ON. 3h (R) = Memory bank is on when the domain is ON.
17-16	MPU_RAM_ONSTATE	Rreturns1s	3h	Default power domain memory state when domain is ON. 3h (R) = Memory bank is on when the domain is ON.
15-11	RESERVED	R	0h	
10	MPU_L2_RETSTATE	R/W	1h	L2 bank state when domain is retention. 0h (R/W) = Memory is off when the domain is in the RETENTION state. 1h (R/W) = Memory is retained when domain is in RETENTION state.
9	MPU_L1_RETSTATE	R/W	1h	L1 bank state when domain is retention. 0h (R/W) = Memory is off when the domain is in the RETENTION state. 1h (R/W) = Memory is retained when domain is in RETENTION state.
8	MPU_RAM_RETSTATE	R/W	1h	MPU RAM bank state when domain is retention. 0h (R/W) = Memory is off when the domain is in the RETENTION state. 1h (R/W) = Memory is retained when domain is in RETENTION state.
7-5	RESERVED	Rreturns0s	0h	

Table 6-56. PRCM_PM_MPU_PWRSTCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	LOWPOWERSTATECHANGE	R/WSpecial	0h	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0h (R/W) = Do not request a low power state change. 1h (R/W) = Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.
3	RESERVED	Rreturns0s	0h	
2	LOGICRETSTATE	R/W	1h	Logic state when power domain is RETENTION 0h (R) = Only retention registers are retained and remaing logic is off when the domain is in RETENTION state. 1h (R/W) = Whole logic is retained when domain is in RETENTION state.
1-0	POWERSTATE	R/W	3h	Power state control 0h (R/W) = OFF State 1h (R/W) = RETENTION state 2h (R/W) = RESERVED 3h (R/W) = ON State

6.12.4.2 PRCM_PM_MPU_PWRSTST Register (offset = 4h) [reset = 157h]

Register mask: FFFFFFFFh

PRCM_PM_MPU_PWRSTST is shown in Figure 6-51 and described in Table 6-57.

This register provides a status on the current MPU power domain state0. [warm reset insensitive]

Figure 6-51. PRCM_PM_MPU_PWRSTST Register

31	30	29	28	27	26	25	24
RESERVED						LASTPOWERSTATEENTERED	
Rreturns0s-0h						R/W1toSet-0h	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED			INTRANSITION	RESERVED			
Rreturns0s-0h			R-0h	Rreturns0s-0h			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED						MPU_L2_STATEST	
Rreturns0s-0h						R-1h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MPU_L1_STATEST		MPU_RAM_STATEST		RESERVED	LOGICSTATEST	POWERSTATEST	
R-1h		R-1h		Rreturns0s-0h	R-1h	R-3h	

Table 6-57. PRCM_PM_MPU_PWRSTST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	Rreturns0s	0h	
25-24	LASTPOWERSTATEENTERED	R/W1toSet	0h	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0h (R) = Power domain was previously OFF 1h (R) = Power domain was previously in RETENTION 2h (R) = Power domain was previously ON-INACTIVE 3h (R) = Power domain was previously ON-ACTIVE
23-21	RESERVED	Rreturns0s	0h	
20	INTRANSITION	R	0h	Domain transition status 0h (R) = No on-going transition on power domain 1h (R) = Power domain transition is in progress.
19-10	RESERVED	Rreturns0s	0h	
9-8	MPU_L2_STATEST	R	1h	MPU L2 memory state status 0h (R) = Memory is OFF 1h (R) = Memory is in RETENTION. 2h (R) = Reserved 3h (R) = Memory is ON
7-6	MPU_L1_STATEST	R	1h	MPU L1 memory state status 0h (R) = Memory is OFF 1h (R) = Memory is in RETENTION. 2h (R) = Reserved 3h (R) = Memory is ON
5-4	MPU_RAM_STATEST	R	1h	MPU_RAM memory state status 0h (R) = Memory is OFF 1h (R) = Memory is in RETENTION. 2h (R) = Reserved 3h (R) = Memory is ON
3	RESERVED	Rreturns0s	0h	

Table 6-57. PRCM_PM_MPU_PWRSTST Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	LOGICSTATEST	R	1h	Logic state status 0h (R) = Logic in domain is OFF 1h (R) = Logic in domain is ON
1-0	POWERSTATEST	R	3h	Current Power State Status 0h (R) = OFF State [warm reset insensitive] 1h (R) = RETENTION state 3h (R) = ON State [warm reset insensitive]

6.12.4.3 PRCM_RM_MPU_RSTST Register (offset = 14h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_RM_MPU_RSTST is shown in Figure 6-52 and described in Table 6-58.

This register logs the different reset sources of the ALWON domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]

Figure 6-52. PRCM_RM_MPU_RSTST Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED						
Rreturns0s-0h	Rreturns0s-0h						
7	6	5	4	3	2	1	0
RESERVED	ICECRUSHER_MPU_RST	EMULATION_MPU_RST	RESERVED	RESERVED	RESERVED	RESERVED	
Rreturns0s-0h	R/W1toClr-0h	R/W1toClr-0h	Rreturns0s-0h	Rreturns0s-0h	Rreturns0s-0h	Rreturns0s-0h	

Table 6-58. PRCM_RM_MPU_RSTST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	Rreturns0s	0h	
14-8	RESERVED	Rreturns0s	0h	
7	RESERVED	Rreturns0s	0h	
6	ICECRUSHER_MPU_RST	R/W1toClr	0h	MPU Processor has been reset due to MPU ICECRUSHER1 reset event 0h (R/W) = No icecrusher reset 1h (R/W) = MPU Processor has been reset upon icecursher reset
5	EMULATION_MPU_RST	R/W1toClr	0h	MPU Processor has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0h (R/W) = No emulation reset 1h (R/W) = MPU Processor has been reset upon emulation reset
4	RESERVED	Rreturns0s	0h	
3	RESERVED	Rreturns0s	0h	
2	RESERVED	Rreturns0s	0h	
1-0	RESERVED	Rreturns0s	0h	

6.12.4.4 PRCM_RM_MPU_CONTEXT Register (offset = 24h) [reset = 701h]

Register mask: FFFFFFFFh

PRCM_RM_MPU_CONTEXT is shown in Figure 6-53 and described in Table 6-59.

This register contains dedicated MPU context statuses.
[warm reset insensitive]

Figure 6-53. PRCM_RM_MPU_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
Rreturns0s-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED					LOSTMEM_MPU_L2	LOSTMEM_MPU_L1	LOSTMEM_MPU_RAM
Rreturns0s-0h					R/W1toClr-1h	R/W1toClr-1h	R/W1toClr-1h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-59. PRCM_RM_MPU_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	Rreturns0s	0h	
10	LOSTMEM_MPU_L2	R/W1toClr	1h	Specify if memory-based context in MPU_L2 memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
9	LOSTMEM_MPU_L1	R/W1toClr	1h	Specify if memory-based context in MPU_L1 memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
8	LOSTMEM_MPU_RAM	R/W1toClr	1h	Specify if memory-based context in MPU_RAM memory bank has been lost due to a previous power transition or other reset source (not affected by a global warm reset). 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of MPU_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5 PRCM_PRM_PER Registers

Table 6-60 lists the memory-mapped registers for the PRCM_PRM_PER. All register offset addresses not listed in Table 6-60 should be considered as reserved locations and the register contents should not be modified.

Table 6-60. PRCM_PRM_PER REGISTERS

Offset	Acronym	Register Name	Section
0h	PRCM_PM_PER_PWRSTCTRL		Section 6.12.5.1
4h	PRCM_PM_PER_PWRSTST		Section 6.12.5.2
10h	PRCM_RM_PER_RSTCTRL		Section 6.12.5.3
14h	PRCM_RM_PER_RSTST		Section 6.12.5.4
24h	PRCM_RM_PER_L3_CONTEXT		Section 6.12.5.5
44h	PRCM_RM_PER_L3_INSTR_CONTEXT		Section 6.12.5.6
54h	PRCM_RM_PER_OCMCRAM_CONTEXT		Section 6.12.5.7
6Ch	PRCM_RM_PER_VPFE0_CONTEXT		Section 6.12.5.8
74h	PRCM_RM_PER_VPFE1_CONTEXT		Section 6.12.5.9
7Ch	PRCM_RM_PER_TPCC_CONTEXT		Section 6.12.5.10
84h	PRCM_RM_PER_TPTC0_CONTEXT		Section 6.12.5.11
8Ch	PRCM_RM_PER_TPTC1_CONTEXT		Section 6.12.5.12
94h	PRCM_RM_PER_TPTC2_CONTEXT		Section 6.12.5.13
9Ch	PRCM_RM_PER_DLL_AGING_CONTEXT		Section 6.12.5.14
A4h	PRCM_RM_PER_L4HS_CONTEXT		Section 6.12.5.15
224h	PRCM_RM_PER_GPMC_CONTEXT		Section 6.12.5.16
234h	PRCM_RM_PER_ADC1_CONTEXT		Section 6.12.5.17
23Ch	PRCM_RM_PER_MCASP0_CONTEXT		Section 6.12.5.18
244h	PRCM_RM_PER_MCASP1_CONTEXT		Section 6.12.5.19
24Ch	PRCM_RM_PER_MMC2_CONTEXT		Section 6.12.5.20
25Ch	PRCM_RM_PER_QSPI_CONTEXT		Section 6.12.5.21
264h	PRCM_RM_PER_USB_OTG_SS0_CONTEXT		Section 6.12.5.22
26Ch	PRCM_RM_PER_USB_OTG_SS1_CONTEXT		Section 6.12.5.23
324h	PRCM_RM_PER_PRU_ICSS_CONTEXT		Section 6.12.5.24
424h	PRCM_RM_PER_L4LS_CONTEXT		Section 6.12.5.25
42Ch	PRCM_RM_PER_DCAN0_CONTEXT		Section 6.12.5.26
434h	PRCM_RM_PER_DCAN1_CONTEXT		Section 6.12.5.27
43Ch	PRCM_RM_PER_PWMSS0_CONTEXT		Section 6.12.5.28
444h	PRCM_RM_PER_PWMSS1_CONTEXT		Section 6.12.5.29
44Ch	PRCM_RM_PER_PWMSS2_CONTEXT		Section 6.12.5.30
454h	PRCM_RM_PER_PWMSS3_CONTEXT		Section 6.12.5.31
45Ch	PRCM_RM_PER_PWMSS4_CONTEXT		Section 6.12.5.32
464h	PRCM_RM_PER_PWMSS5_CONTEXT		Section 6.12.5.33
46Ch	PRCM_RM_PER_ELM_CONTEXT		Section 6.12.5.34
47Ch	PRCM_RM_PER_GPIO1_CONTEXT		Section 6.12.5.35
484h	PRCM_RM_PER_GPIO2_CONTEXT		Section 6.12.5.36
48Ch	PRCM_RM_PER_GPIO3_CONTEXT		Section 6.12.5.37
494h	PRCM_RM_PER_GPIO4_CONTEXT		Section 6.12.5.38
49Ch	PRCM_RM_PER_GPIO5_CONTEXT		Section 6.12.5.39
4A4h	PRCM_RM_PER_HDQ1W_CONTEXT		Section 6.12.5.40
4ACh	PRCM_RM_PER_I2C1_CONTEXT		Section 6.12.5.41
4B4h	PRCM_RM_PER_I2C2_CONTEXT		Section 6.12.5.42
4BCh	PRCM_RM_PER_MAILBOX0_CONTEXT		Section 6.12.5.43

Table 6-60. PRCM_PRM_PER REGISTERS (continued)

Offset	Acronym	Register Name	Section
4C4h	PRCM_RM_PER_MMC0_CONTEXT		Section 6.12.5.44
4CCh	PRCM_RM_PER_MMC1_CONTEXT		Section 6.12.5.45
504h	PRCM_RM_PER_SPI0_CONTEXT		Section 6.12.5.46
50Ch	PRCM_RM_PER_SPI1_CONTEXT		Section 6.12.5.47
514h	PRCM_RM_PER_SPI2_CONTEXT		Section 6.12.5.48
51Ch	PRCM_RM_PER_SPI3_CONTEXT		Section 6.12.5.49
524h	PRCM_RM_PER_SPI4_CONTEXT		Section 6.12.5.50
52Ch	PRCM_RM_PER_SPINLOCK_CONTEXT		Section 6.12.5.51
534h	PRCM_RM_PER_TIMER2_CONTEXT		Section 6.12.5.52
53Ch	PRCM_RM_PER_TIMER3_CONTEXT		Section 6.12.5.53
544h	PRCM_RM_PER_TIMER4_CONTEXT		Section 6.12.5.54
54Ch	PRCM_RM_PER_TIMER5_CONTEXT		Section 6.12.5.55
554h	PRCM_RM_PER_TIMER6_CONTEXT		Section 6.12.5.56
55Ch	PRCM_RM_PER_TIMER7_CONTEXT		Section 6.12.5.57
564h	PRCM_RM_PER_TIMER8_CONTEXT		Section 6.12.5.58
56Ch	PRCM_RM_PER_TIMER9_CONTEXT		Section 6.12.5.59
574h	PRCM_RM_PER_TIMER10_CONTEXT		Section 6.12.5.60
57Ch	PRCM_RM_PER_TIMER11_CONTEXT		Section 6.12.5.61
584h	PRCM_RM_PER_UART1_CONTEXT		Section 6.12.5.62
58Ch	PRCM_RM_PER_UART2_CONTEXT		Section 6.12.5.63
594h	PRCM_RM_PER_UART3_CONTEXT		Section 6.12.5.64
59Ch	PRCM_RM_PER_UART4_CONTEXT		Section 6.12.5.65
5A4h	PRCM_RM_PER_UART5_CONTEXT		Section 6.12.5.66
5BCh	PRCM_RM_PER_USBPHYOCP2SCP0_CONTEXT		Section 6.12.5.67
5C4h	PRCM_RM_PER_USBPHYOCP2SCP1_CONTEXT		Section 6.12.5.68
724h	PRCM_RM_PER_EMIF_CONTEXT		Section 6.12.5.69
72Ch	PRCM_RM_PER_DLL_CONTEXT		Section 6.12.5.70
A24h	PRCM_RM_PER_DSS_CONTEXT		Section 6.12.5.71
B24h	PRCM_RM_PER_CPGMAC0_CONTEXT		Section 6.12.5.72
C24h	PRCM_RM_PER_OCPWP_CONTEXT		Section 6.12.5.73

6.12.5.1 PRCM_PM_PER_PWRSTCTRL Register (offset = 0h) [reset = FF0F07h]

Register mask: FFFFFFFFh

PRCM_PM_PER_PWRSTCTRL is shown in Figure 6-54 and described in Table 6-61.

Controls the power state of PER power domain

Figure 6-54. PRCM_PM_PER_PWRSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RAM2_MEM_ONSTATE		RAM1_MEM_ONSTATE		PER_MEM_ONSTATE		PRU_ICSS_MEM_ONSTATE	
Rreturns1s-3h		Rreturns1s-3h		Rreturns1s-3h		Rreturns1s-3h	

15	14	13	12	11	10	9	8
RESERVED				RAM2_MEM_RETSTATE	RAM1_MEM_RETSTATE	PER_MEM_RETSTATE	PRU_ICSS_MEM_RETSTATE
R-0h				R/W-1h	R/W-1h	R/W-1h	R/W-1h

7	6	5	4	3	2	1	0
RESERVED			LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE	
R-0h			R/WSpecial-0h	Rreturns0s-0h	R/W-1h	R/W-3h	

Table 6-61. PRCM_PM_PER_PWRSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	Rreturns0s	0h	
23-22	RAM2_MEM_ONSTATE	Rreturns1s	3h	OCMC RAM Group 2 (Last 192KB) memory on state 3h (R) = Memory is ON
21-20	RAM1_MEM_ONSTATE	Rreturns1s	3h	OCMC RAM Group 1 (First 64KB) memory on state 3h (R) = Memory is ON
19-18	PER_MEM_ONSTATE	Rreturns1s	3h	Other memories in PER Domain ON state 3h (R) = Memory is ON
17-16	PRU_ICSS_MEM_ONSTATE	Rreturns1s	3h	PRU-ICSS memory ON state 3h (R) = Memory is ON
15-12	RESERVED	R	0h	
11	RAM2_MEM_RETSTATE	R/W	1h	RAM2_MEM[OCMC RAM Group 2 (Last 192KB)] bank state when domain is retention. 0h (R/W) = Memory is off when the domain is in the RETENTION state. 1h (R/W) = Memory is retained when domain is in RETENTION state.
10	RAM1_MEM_RETSTATE	R/W	1h	RAM1_MEM[OCMC RAM Group 1 (First 64KB)] bank state when domain is retention. 0h (R/W) = Memory is off when the domain is in the RETENTION state. 1h (R/W) = Memory is retained when domain is in RETENTION state.
9	PER_MEM_RETSTATE	R/W	1h	PER_MEM bank state when domain is retention. 0h (R/W) = Memory is off when the domain is in the RETENTION state. 1h (R/W) = Memory is retained when domain is in RETENTION state.

Table 6-61. PRCM_PM_PER_PWRSTCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	PRU_ICSS_MEM_RETSTATE	R/W	1h	PRU-ICSS bank state when domain is retention. 0h (R/W) = Memory is off when the domain is in the RETENTION state. 1h (R/W) = Memory is retained when domain is in RETENTION state.
7-5	RESERVED	R	0h	
4	LOWPOWERSTATECHANGE	R/WSpecial	0h	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0h (R/W) = Do not request a low power state change. 1h (R/W) = Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.
3	RESERVED	Rreturns0s	0h	
2	LOGICRETSTATE	R/W	1h	Logic state when power domain is RETENTION 0h (R) = Only retention registers are retained and remaing logic is off when the domain is in RETENTION state. 1h (R/W) = Whole logic is retained when domain is in RETENTION state.
1-0	POWERSTATE	R/W	3h	PER domain power state control 0h (R/W) = OFF State 1h (R/W) = RETENTION state 2h (R) = RESERVED 3h (R/W) = ON State

6.12.5.2 PRCM_PM_PER_PWRSTST Register (offset = 4h) [reset = FF7h]

Register mask: FFFFFFFFh

PRCM_PM_PER_PWRSTST is shown in Figure 6-55 and described in Table 6-62.

This register provides a status on the current PER power domain state. [warm reset insensitive]

Figure 6-55. PRCM_PM_PER_PWRSTST Register

31	30	29	28	27	26	25	24
RESERVED						LASTPOWERSTATEENTERED	
Rreturns0s-0h						R/W1toSet-0h	

23	22	21	20	19	18	17	16
RESERVED			INTRANSITION	RESERVED			
Rreturns0s-0h			R-0h	Rreturns0s-0h			

15	14	13	12	11	10	9	8
RESERVED				RAM2_MEM_STATEST		RAM1_MEM_STATEST	
Rreturns0s-0h				R-3h		R-3h	

7	6	5	4	3	2	1	0
PER_MEM_STATEST		PRU_ICSS_MEM_STATEST		RESERVED	LOGICSTATEST	POWERSTATEST	
R-3h		R-3h		Rreturns0s-0h	R-1h	R-3h	

Table 6-62. PRCM_PM_PER_PWRSTST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	Rreturns0s	0h	
25-24	LASTPOWERSTATEENTERED	R/W1toSet	0h	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0h (R) = Power domain was previously OFF 1h (R) = Power domain was previously in RETENTION 2h (R) = Power domain was previously ON-INACTIVE 3h (R) = Power domain was previously ON-ACTIVE
23-21	RESERVED	Rreturns0s	0h	
20	INTRANSITION	R	0h	Domain transition status 0h (R) = No on-going transition on power domain 1h (R) = Power domain transition is in progress.
19-12	RESERVED	Rreturns0s	0h	
11-10	RAM2_MEM_STATEST	R	3h	OCMC RAM Group 1 (Last 192KB) memory state status 0h (R) = Memory is OFF 1h (R) = Memory is in RETENTION. 2h (R) = Reserved 3h (R) = Memory is ON
9-8	RAM1_MEM_STATEST	R	3h	OCMC RAM Group 1 (First 64KB) memory state status 0h (R) = Memory is OFF 1h (R) = Memory is in RETENTION. 2h (R) = Reserved 3h (R) = Memory is ON
7-6	PER_MEM_STATEST	R	3h	PER domain memory state status 0h (R) = Memory is OFF 1h (R) = Memory is in RETENTION. 2h (R) = Reserved 3h (R) = Memory is ON

Table 6-62. PRCM_PM_PER_PWRSTST Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	PRU_ICSS_MEM_STATEST	R	3h	PRU-ICSS memory state status 0h (R) = Memory is OFF 1h (R) = Memory is in RETENTION. 2h (R) = Reserved 3h (R) = Memory is ON
3	RESERVED	Rreturns0s	0h	
2	LOGICSTATEST	R	1h	Logic state status 0h (R) = Logic in domain is OFF 1h (R) = Logic in domain is ON
1-0	POWERSTATEST	R	3h	Current Power State Status 0h (R) = OFF State 1h (R) = RETENTION state 3h (R) = ON State

6.12.5.3 PRCM_RM_PER_RSTCTRL Register (offset = 10h) [reset = 3h]

Register mask: FFFFFFFFh

PRCM_RM_PER_RSTCTRL is shown in Figure 6-56 and described in Table 6-63.

This register controls the release of the PER Domain resets.

Figure 6-56. PRCM_RM_PER_RSTCTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">Rreturns0s-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">Rreturns0s-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">Rreturns0s-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="3">RESERVED</td><td>ASSERT</td><td>PRU_ICSS_LRST</td><td>RESERVED</td></tr>
<tr><td colspan="2">Rreturns0s-0h</td><td colspan="3">Rreturns0s-0h</td><td>Rreturns0s-0h</td><td>R/W-1h</td><td>Rreturns0s-1h</td></tr>
</table>

Table 6-63. PRCM_RM_PER_RSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	Rreturns0s	0h	
5-3	RESERVED	Rreturns0s	0h	
2	ASSERT	Rreturns0s	0h	
1	PRU_ICSS_LRST	R/W	1h	PER domain PRU-ICSS local reset control 0h (R/W) = Reset is cleared for the PRU-ICSS 1h (R/W) = Reset is asserted for the PRU-ICSS
0	RESERVED	Rreturns0s	1h	

6.12.5.4 PRCM_RM_PER_RSTST Register (offset = 14h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_RM_PER_RSTST is shown in Figure 6-57 and described in Table 6-64.

This register logs the different reset sources of the PER domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]

Figure 6-57. PRCM_RM_PER_RSTST Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							PRU_ICSS_LRST
Rreturns0s-0h							R/W1toClr-0h

Table 6-64. PRCM_RM_PER_RSTST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	PRU_ICSS_LRST	R/W1toClr	0h	PRU-ICSS Processor software reset status. 0h (R/W) = No reset 1h (R/W) = PRU-ICSS Processor has been reset upon SW reset

6.12.5.5 PRCM_RM_PER_L3_CONTEXT Register (offset = 24h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_L3_CONTEXT is shown in Figure 6-58 and described in Table 6-65.

This register contains dedicated L3 module context statuses. [warm reset insensitive]

Figure 6-58. PRCM_RM_PER_L3_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-65. PRCM_RM_PER_L3_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.6 PRCM_RM_PER_L3_INSTR_CONTEXT Register (offset = 44h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_L3_INSTR_CONTEXT is shown in Figure 6-59 and described in Table 6-66.

This register contains dedicated L3_INSTR module context statuses. [warm reset insensitive]

Figure 6-59. PRCM_RM_PER_L3_INSTR_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-66. PRCM_RM_PER_L3_INSTR_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.7 PRCM_RM_PER_OCMCRAM_CONTEXT Register (offset = 54h) [reset = 301h]

Register mask: FFFFFFFFh

PRCM_RM_PER_OCMCRAM_CONTEXT is shown in Figure 6-60 and described in Table 6-67.

This register contains dedicated OCMCRAM module context statuses. [warm reset insensitive]

Figure 6-60. PRCM_RM_PER_OCMCRAM_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED						LOSTMEM_RAM2_MEM	LOSTMEM_RAM1_MEM
Rreturns0s-0h						R/W1toClr-1h	R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-67. PRCM_RM_PER_OCMCRAM_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	Rreturns0s	0h	
9	LOSTMEM_RAM2_MEM	R/W1toClr	1h	Specify if memory-based context in RAM2_MEM memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
8	LOSTMEM_RAM1_MEM	R/W1toClr	1h	Specify if memory-based context in RAM1_MEM memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.8 PRCM_RM_PER_VPFE0_CONTEXT Register (offset = 6Ch) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_VPFE0_CONTEXT is shown in Figure 6-61 and described in Table 6-68.

This register contains dedicated VPFE0 module context statuses. [warm reset insensitive]

Figure 6-61. PRCM_RM_PER_VPFE0_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-68. PRCM_RM_PER_VPFE0_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.9 PRCM_RM_PER_VPFE1_CONTEXT Register (offset = 74h) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_VPFE1_CONTEXT is shown in Figure 6-62 and described in Table 6-69.

This register contains dedicated VPFE1 module context statuses. [warm reset insensitive]

Figure 6-62. PRCM_RM_PER_VPFE1_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-69. PRCM_RM_PER_VPFE1_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.10 PRCM_RM_PER_TPCC_CONTEXT Register (offset = 7Ch) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_TPCC_CONTEXT is shown in Figure 6-63 and described in Table 6-70.

This register contains dedicated TPCC module context statuses. [warm reset insensitive]

Figure 6-63. PRCM_RM_PER_TPCC_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-70. PRCM_RM_PER_TPCC_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.11 PRCM_RM_PER_TPTC0_CONTEXT Register (offset = 84h) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_TPTC0_CONTEXT is shown in Figure 6-64 and described in Table 6-71.

This register contains dedicated TPTC0 module context statuses. [warm reset insensitive]

Figure 6-64. PRCM_RM_PER_TPTC0_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-71. PRCM_RM_PER_TPTC0_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.12 PRCM_RM_PER_TPTC1_CONTEXT Register (offset = 8Ch) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_TPTC1_CONTEXT is shown in Figure 6-65 and described in Table 6-72.

This register contains dedicated TPTC1 module context statuses. [warm reset insensitive]

Figure 6-65. PRCM_RM_PER_TPTC1_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-72. PRCM_RM_PER_TPTC1_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.13 PRCM_RM_PER_TPTC2_CONTEXT Register (offset = 94h) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_TPTC2_CONTEXT is shown in Figure 6-66 and described in Table 6-73.

This register contains dedicated TPTC2 module context statuses. [warm reset insensitive]

Figure 6-66. PRCM_RM_PER_TPTC2_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-73. PRCM_RM_PER_TPTC2_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.14 PRCM_RM_PER_DLL_AGING_CONTEXT Register (offset = 9Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_DLL_AGING_CONTEXT is shown in Figure 6-67 and described in Table 6-74.

This register contains dedicated DLL_AGING module context statuses. [warm reset insensitive]

Figure 6-67. PRCM_RM_PER_DLL_AGING_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-74. PRCM_RM_PER_DLL_AGING_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.15 PRCM_RM_PER_L4HS_CONTEXT Register (offset = A4h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_L4HS_CONTEXT is shown in Figure 6-68 and described in Table 6-75.

This register contains dedicated L4HS module context statuses. [warm reset insensitive]

Figure 6-68. PRCM_RM_PER_L4HS_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-75. PRCM_RM_PER_L4HS_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.16 PRCM_RM_PER_GPMC_CONTEXT Register (offset = 224h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_GPMC_CONTEXT is shown in Figure 6-69 and described in Table 6-76.

This register contains dedicated GPMC module context statuses. [warm reset insensitive]

Figure 6-69. PRCM_RM_PER_GPMC_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-76. PRCM_RM_PER_GPMC_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.17 PRCM_RM_PER_ADC1_CONTEXT Register (offset = 234h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_ADC1_CONTEXT is shown in Figure 6-70 and described in Table 6-77.

This register contains dedicated ADC1 module context statuses. [warm reset insensitive]

Figure 6-70. PRCM_RM_PER_ADC1_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-77. PRCM_RM_PER_ADC1_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.18 PRCM_RM_PER_MCASP0_CONTEXT Register (offset = 23Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_MCASP0_CONTEXT is shown in Figure 6-71 and described in Table 6-78.

This register contains dedicated MCASP0 module context statuses. [warm reset insensitive]

Figure 6-71. PRCM_RM_PER_MCASP0_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-78. PRCM_RM_PER_MCASP0_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.19 PRCM_RM_PER_MCASP1_CONTEXT Register (offset = 244h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_MCASP1_CONTEXT is shown in Figure 6-72 and described in Table 6-79.

This register contains dedicated MCASP1 module context statuses. [warm reset insensitive]

Figure 6-72. PRCM_RM_PER_MCASP1_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-79. PRCM_RM_PER_MCASP1_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.20 PRCM_RM_PER_MMC2_CONTEXT Register (offset = 24Ch) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_MMC2_CONTEXT is shown in Figure 6-73 and described in Table 6-80.

This register contains dedicated MMC2 module context statuses. [warm reset insensitive]

Figure 6-73. PRCM_RM_PER_MMC2_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-80. PRCM_RM_PER_MMC2_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.21 PRCM_RM_PER_QSPI_CONTEXT Register (offset = 25Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_QSPI_CONTEXT is shown in Figure 6-74 and described in Table 6-81.

This register contains dedicated QSPI module context statuses. [warm reset insensitive]

Figure 6-74. PRCM_RM_PER_QSPI_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-81. PRCM_RM_PER_QSPI_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.22 PRCM_RM_PER_USB_OTG_SS0_CONTEXT Register (offset = 264h) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_USB_OTG_SS0_CONTEXT is shown in Figure 6-75 and described in Table 6-82.

This register contains dedicated USB_OTG0 context statuses. [warm reset insensitive]

Figure 6-75. PRCM_RM_PER_USB_OTG_SS0_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-82. PRCM_RM_PER_USB_OTG_SS0_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in _BANK1 memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.23 PRCM_RM_PER_USB_OTG_SS1_CONTEXT Register (offset = 26Ch) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_USB_OTG_SS1_CONTEXT is shown in Figure 6-76 and described in Table 6-83.

This register contains dedicated USB_OTG0 context statuses. [warm reset insensitive]

Figure 6-76. PRCM_RM_PER_USB_OTG_SS1_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-83. PRCM_RM_PER_USB_OTG_SS1_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in _BANK1 memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.24 PRCM_RM_PER_PRU_ICSS_CONTEXT Register (offset = 324h) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_PRU_ICSS_CONTEXT is shown in Figure 6-77 and described in Table 6-84.

This register contains dedicated ICSS module context statuses. [warm reset insensitive]

Figure 6-77. PRCM_RM_PER_PRU_ICSS_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-84. PRCM_RM_PER_PRU_ICSS_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.25 PRCM_RM_PER_L4LS_CONTEXT Register (offset = 424h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_L4LS_CONTEXT is shown in Figure 6-78 and described in Table 6-85.

This register contains dedicated L4LS module context statuses. [warm reset insensitive]

Figure 6-78. PRCM_RM_PER_L4LS_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-85. PRCM_RM_PER_L4LS_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.26 PRCM_RM_PER_DCAN0_CONTEXT Register (offset = 42Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_DCAN0_CONTEXT is shown in Figure 6-79 and described in Table 6-86.

This register contains dedicated DCAN0 module context statuses. [warm reset insensitive]

Figure 6-79. PRCM_RM_PER_DCAN0_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-86. PRCM_RM_PER_DCAN0_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.27 PRCM_RM_PER_DCAN1_CONTEXT Register (offset = 434h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_DCAN1_CONTEXT is shown in Figure 6-80 and described in Table 6-87.

This register contains dedicated DCAN1 module context statuses. [warm reset insensitive]

Figure 6-80. PRCM_RM_PER_DCAN1_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-87. PRCM_RM_PER_DCAN1_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.28 PRCM_RM_PER_PWMSS0_CONTEXT Register (offset = 43Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_PWMSS0_CONTEXT is shown in Figure 6-81 and described in Table 6-88.

This register contains dedicated PWMSS0 module context statuses. [warm reset insensitive]

Figure 6-81. PRCM_RM_PER_PWMSS0_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-88. PRCM_RM_PER_PWMSS0_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.29 PRCM_RM_PER_PWMSS1_CONTEXT Register (offset = 444h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_PWMSS1_CONTEXT is shown in Figure 6-82 and described in Table 6-89.

This register contains dedicated PWMSS1 module context statuses. [warm reset insensitive]

Figure 6-82. PRCM_RM_PER_PWMSS1_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-89. PRCM_RM_PER_PWMSS1_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.30 PRCM_RM_PER_PWMSS2_CONTEXT Register (offset = 44Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_PWMSS2_CONTEXT is shown in Figure 6-83 and described in Table 6-90.

This register contains dedicated PWMSS2 module context statuses. [warm reset insensitive]

Figure 6-83. PRCM_RM_PER_PWMSS2_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-90. PRCM_RM_PER_PWMSS2_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.31 PRCM_RM_PER_PWMSS3_CONTEXT Register (offset = 454h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_PWMSS3_CONTEXT is shown in Figure 6-84 and described in Table 6-91.

This register contains dedicated PWMSS3 module context statuses. [warm reset insensitive]

Figure 6-84. PRCM_RM_PER_PWMSS3_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-91. PRCM_RM_PER_PWMSS3_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.32 PRCM_RM_PER_PWMSS4_CONTEXT Register (offset = 45Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_PWMSS4_CONTEXT is shown in Figure 6-85 and described in Table 6-92.

This register contains dedicated PWMSS4 module context statuses. [warm reset insensitive]

Figure 6-85. PRCM_RM_PER_PWMSS4_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-92. PRCM_RM_PER_PWMSS4_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.33 PRCM_RM_PER_PWMSS5_CONTEXT Register (offset = 464h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_PWMSS5_CONTEXT is shown in Figure 6-86 and described in Table 6-93.

This register contains dedicated PWMSS5 module context statuses. [warm reset insensitive]

Figure 6-86. PRCM_RM_PER_PWMSS5_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-93. PRCM_RM_PER_PWMSS5_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.34 PRCM_RM_PER_ELM_CONTEXT Register (offset = 46Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_ELM_CONTEXT is shown in Figure 6-87 and described in Table 6-94.

This register contains dedicated ELM module context statuses. [warm reset insensitive]

Figure 6-87. PRCM_RM_PER_ELM_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
Rreturns0s-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
Rreturns0s-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-94. PRCM_RM_PER_ELM_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.35 PRCM_RM_PER_GPIO1_CONTEXT Register (offset = 47Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_GPIO1_CONTEXT is shown in Figure 6-88 and described in Table 6-95.

This register contains dedicated GPIO1 module context statuses. [warm reset insensitive]

Figure 6-88. PRCM_RM_PER_GPIO1_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-95. PRCM_RM_PER_GPIO1_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.36 PRCM_RM_PER_GPIO2_CONTEXT Register (offset = 484h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_GPIO2_CONTEXT is shown in Figure 6-89 and described in Table 6-96.

This register contains dedicated GPIO2 module context statuses. [warm reset insensitive]

Figure 6-89. PRCM_RM_PER_GPIO2_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-96. PRCM_RM_PER_GPIO2_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.37 PRCM_RM_PER_GPIO3_CONTEXT Register (offset = 48Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_GPIO3_CONTEXT is shown in Figure 6-90 and described in Table 6-97.

This register contains dedicated GPIO3 module context statuses. [warm reset insensitive]

Figure 6-90. PRCM_RM_PER_GPIO3_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-97. PRCM_RM_PER_GPIO3_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.38 PRCM_RM_PER_GPIO4_CONTEXT Register (offset = 494h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_GPIO4_CONTEXT is shown in Figure 6-91 and described in Table 6-98.

This register contains dedicated GPIO4 module context statuses. [warm reset insensitive]

Figure 6-91. PRCM_RM_PER_GPIO4_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-98. PRCM_RM_PER_GPIO4_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.39 PRCM_RM_PER_GPIO5_CONTEXT Register (offset = 49Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_GPIO5_CONTEXT is shown in Figure 6-92 and described in Table 6-99.

This register contains dedicated GPIO5 module context statuses. [warm reset insensitive]

Figure 6-92. PRCM_RM_PER_GPIO5_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-99. PRCM_RM_PER_GPIO5_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.40 PRCM_RM_PER_HDQ1W_CONTEXT Register (offset = 4A4h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_HDQ1W_CONTEXT is shown in Figure 6-93 and described in Table 6-100.

This register contains dedicated HDQ1W module context statuses. [warm reset insensitive]

Figure 6-93. PRCM_RM_PER_HDQ1W_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-100. PRCM_RM_PER_HDQ1W_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.41 PRCM_RM_PER_I2C1_CONTEXT Register (offset = 4ACh) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_I2C1_CONTEXT is shown in Figure 6-94 and described in Table 6-101.

This register contains dedicated I2C1 module context statuses. [warm reset insensitive]

Figure 6-94. PRCM_RM_PER_I2C1_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-101. PRCM_RM_PER_I2C1_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.42 PRCM_RM_PER_I2C2_CONTEXT Register (offset = 4B4h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_I2C2_CONTEXT is shown in Figure 6-95 and described in Table 6-102.

This register contains dedicated I2C2 module context statuses. [warm reset insensitive]

Figure 6-95. PRCM_RM_PER_I2C2_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-102. PRCM_RM_PER_I2C2_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.43 PRCM_RM_PER_MAILBOX0_CONTEXT Register (offset = 4BCh) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_MAILBOX0_CONTEXT is shown in Figure 6-96 and described in Table 6-103.

This register contains dedicated MAILBOX0 module context statuses. [warm reset insensitive]

Figure 6-96. PRCM_RM_PER_MAILBOX0_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-103. PRCM_RM_PER_MAILBOX0_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.44 PRCM_RM_PER_MMC0_CONTEXT Register (offset = 4C4h) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_MMC0_CONTEXT is shown in Figure 6-97 and described in Table 6-104.

This register contains dedicated MMC0 module context statuses. [warm reset insensitive]

Figure 6-97. PRCM_RM_PER_MMC0_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-104. PRCM_RM_PER_MMC0_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.45 PRCM_RM_PER_MMC1_CONTEXT Register (offset = 4CCh) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_MMC1_CONTEXT is shown in Figure 6-98 and described in Table 6-105.

This register contains dedicated MMC1 module context statuses. [warm reset insensitive]

Figure 6-98. PRCM_RM_PER_MMC1_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-105. PRCM_RM_PER_MMC1_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.46 PRCM_RM_PER_SPI0_CONTEXT Register (offset = 504h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_SPI0_CONTEXT is shown in Figure 6-99 and described in Table 6-106.

This register contains dedicated SPI0 module context statuses. [warm reset insensitive]

Figure 6-99. PRCM_RM_PER_SPI0_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-106. PRCM_RM_PER_SPI0_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.47 PRCM_RM_PER_SPI1_CONTEXT Register (offset = 50Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_SPI1_CONTEXT is shown in Figure 6-100 and described in Table 6-107.

This register contains dedicated SPI1 module context statuses. [warm reset insensitive]

Figure 6-100. PRCM_RM_PER_SPI1_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-107. PRCM_RM_PER_SPI1_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.48 PRCM_RM_PER_SPI2_CONTEXT Register (offset = 514h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_SPI2_CONTEXT is shown in Figure 6-101 and described in Table 6-108.

This register contains dedicated SPI2 module context statuses. [warm reset insensitive]

Figure 6-101. PRCM_RM_PER_SPI2_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-108. PRCM_RM_PER_SPI2_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.49 PRCM_RM_PER_SPI3_CONTEXT Register (offset = 51Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_SPI3_CONTEXT is shown in Figure 6-102 and described in Table 6-109.

This register contains dedicated SPI3 module context statuses. [warm reset insensitive]

Figure 6-102. PRCM_RM_PER_SPI3_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-109. PRCM_RM_PER_SPI3_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.50 PRCM_RM_PER_SPI4_CONTEXT Register (offset = 524h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_SPI4_CONTEXT is shown in Figure 6-103 and described in Table 6-110.

This register contains dedicated SPI4 module context statuses. [warm reset insensitive]

Figure 6-103. PRCM_RM_PER_SPI4_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-110. PRCM_RM_PER_SPI4_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.51 PRCM_RM_PER_SPINLOCK_CONTEXT Register (offset = 52Ch) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_SPINLOCK_CONTEXT is shown in Figure 6-104 and described in Table 6-111.

This register contains dedicated SPINLOCK module context statuses. [warm reset insensitive]

Figure 6-104. PRCM_RM_PER_SPINLOCK_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-111. PRCM_RM_PER_SPINLOCK_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.52 PRCM_RM_PER_TIMER2_CONTEXT Register (offset = 534h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_TIMER2_CONTEXT is shown in Figure 6-105 and described in Table 6-112.

This register contains dedicated TIMER2 module context statuses. [warm reset insensitive]

Figure 6-105. PRCM_RM_PER_TIMER2_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-112. PRCM_RM_PER_TIMER2_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.53 PRCM_RM_PER_TIMER3_CONTEXT Register (offset = 53Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_TIMER3_CONTEXT is shown in Figure 6-106 and described in Table 6-113.

This register contains dedicated TIMER3 module context statuses. [warm reset insensitive]

Figure 6-106. PRCM_RM_PER_TIMER3_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
Rreturns0s-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
Rreturns0s-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-113. PRCM_RM_PER_TIMER3_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.54 PRCM_RM_PER_TIMER4_CONTEXT Register (offset = 544h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_TIMER4_CONTEXT is shown in Figure 6-107 and described in Table 6-114.

This register contains dedicated TIMER4 module context statuses. [warm reset insensitive]

Figure 6-107. PRCM_RM_PER_TIMER4_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-114. PRCM_RM_PER_TIMER4_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.55 PRCM_RM_PER_TIMER5_CONTEXT Register (offset = 54Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_TIMER5_CONTEXT is shown in Figure 6-108 and described in Table 6-115.

This register contains dedicated TIMER5 module context statuses. [warm reset insensitive]

Figure 6-108. PRCM_RM_PER_TIMER5_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-115. PRCM_RM_PER_TIMER5_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.56 PRCM_RM_PER_TIMER6_CONTEXT Register (offset = 554h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_TIMER6_CONTEXT is shown in Figure 6-109 and described in Table 6-116.

This register contains dedicated TIMER6 module context statuses. [warm reset insensitive]

Figure 6-109. PRCM_RM_PER_TIMER6_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-116. PRCM_RM_PER_TIMER6_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.57 PRCM_RM_PER_TIMER7_CONTEXT Register (offset = 55Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_TIMER7_CONTEXT is shown in Figure 6-110 and described in Table 6-117.

This register contains dedicated TIMER7 module context statuses. [warm reset insensitive]

Figure 6-110. PRCM_RM_PER_TIMER7_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
Rreturns0s-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
Rreturns0s-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-117. PRCM_RM_PER_TIMER7_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.58 PRCM_RM_PER_TIMER8_CONTEXT Register (offset = 564h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_TIMER8_CONTEXT is shown in Figure 6-111 and described in Table 6-118.

This register contains dedicated TIMER8 module context statuses. [warm reset insensitive]

Figure 6-111. PRCM_RM_PER_TIMER8_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-118. PRCM_RM_PER_TIMER8_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.59 PRCM_RM_PER_TIMER9_CONTEXT Register (offset = 56Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_TIMER9_CONTEXT is shown in Figure 6-112 and described in Table 6-119.

This register contains dedicated TIMER9 module context statuses. [warm reset insensitive]

Figure 6-112. PRCM_RM_PER_TIMER9_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-119. PRCM_RM_PER_TIMER9_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.60 PRCM_RM_PER_TIMER10_CONTEXT Register (offset = 574h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_TIMER10_CONTEXT is shown in Figure 6-113 and described in Table 6-120.

This register contains dedicated TIMER10 module context statuses. [warm reset insensitive]

Figure 6-113. PRCM_RM_PER_TIMER10_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-120. PRCM_RM_PER_TIMER10_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.61 PRCM_RM_PER_TIMER11_CONTEXT Register (offset = 57Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_TIMER11_CONTEXT is shown in Figure 6-114 and described in Table 6-121.

This register contains dedicated TIMER11 module context statuses. [warm reset insensitive]

Figure 6-114. PRCM_RM_PER_TIMER11_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-121. PRCM_RM_PER_TIMER11_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.62 PRCM_RM_PER_UART1_CONTEXT Register (offset = 584h) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_UART1_CONTEXT is shown in Figure 6-115 and described in Table 6-122.

This register contains dedicated UART1 module context statuses. [warm reset insensitive]

Figure 6-115. PRCM_RM_PER_UART1_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-122. PRCM_RM_PER_UART1_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.63 PRCM_RM_PER_UART2_CONTEXT Register (offset = 58Ch) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_UART2_CONTEXT is shown in Figure 6-116 and described in Table 6-123.

This register contains dedicated UART2 module context statuses. [warm reset insensitive]

Figure 6-116. PRCM_RM_PER_UART2_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-123. PRCM_RM_PER_UART2_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.64 PRCM_RM_PER_UART3_CONTEXT Register (offset = 594h) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_UART3_CONTEXT is shown in Figure 6-117 and described in Table 6-124.

This register contains dedicated UART3 module context statuses. [warm reset insensitive]

Figure 6-117. PRCM_RM_PER_UART3_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-124. PRCM_RM_PER_UART3_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.65 PRCM_RM_PER_UART4_CONTEXT Register (offset = 59Ch) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_UART4_CONTEXT is shown in Figure 6-118 and described in Table 6-125.

This register contains dedicated UART4 module context statuses. [warm reset insensitive]

Figure 6-118. PRCM_RM_PER_UART4_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-125. PRCM_RM_PER_UART4_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.66 PRCM_RM_PER_UART5_CONTEXT Register (offset = 5A4h) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_UART5_CONTEXT is shown in Figure 6-119 and described in Table 6-126.

This register contains dedicated UART5 module context statuses. [warm reset insensitive]

Figure 6-119. PRCM_RM_PER_UART5_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-126. PRCM_RM_PER_UART5_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.67 PRCM_RM_PER_USBPHYOCP2SCP0_CONTEXT Register (offset = 5BCh) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_USBPHYOCP2SCP0_CONTEXT is shown in Figure 6-120 and described in Table 6-127.

This register contains dedicated USBPHYOCP2SCP0 context statuses. [warm reset insensitive]

Figure 6-120. PRCM_RM_PER_USBPHYOCP2SCP0_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-127. PRCM_RM_PER_USBPHYOCP2SCP0_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3_INIT_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.68 PRCM_RM_PER_USBPHYOCP2SCP1_CONTEXT Register (offset = 5C4h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_USBPHYOCP2SCP1_CONTEXT is shown in Figure 6-121 and described in Table 6-128.

This register contains dedicated USBPHYOCP2SCP0 context statuses. [warm reset insensitive]

Figure 6-121. PRCM_RM_PER_USBPHYOCP2SCP1_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-128. PRCM_RM_PER_USBPHYOCP2SCP1_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3_INIT_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.69 PRCM_RM_PER_EMIF_CONTEXT Register (offset = 724h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_EMIF_CONTEXT is shown in Figure 6-122 and described in Table 6-129.

This register contains dedicated EMIF module context statuses. [warm reset insensitive]

Figure 6-122. PRCM_RM_PER_EMIF_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-129. PRCM_RM_PER_EMIF_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.70 PRCM_RM_PER_DLL_CONTEXT Register (offset = 72Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_DLL_CONTEXT is shown in Figure 6-123 and described in Table 6-130.

This register contains dedicated DLL module context statuses. [warm reset insensitive]

Figure 6-123. PRCM_RM_PER_DLL_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-130. PRCM_RM_PER_DLL_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.71 PRCM_RM_PER_DSS_CONTEXT Register (offset = A24h) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_DSS_CONTEXT is shown in Figure 6-124 and described in Table 6-131.

This register contains dedicated DSS module context statuses. [warm reset insensitive]

Figure 6-124. PRCM_RM_PER_DSS_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
Rreturns0s-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							LOSTMEM_DSS_MEM
Rreturns0s-0h							R/W1toClr-1h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-131. PRCM_RM_PER_DSS_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_DSS_MEM	R/W1toClr	1h	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.72 PRCM_RM_PER_CPGMAC0_CONTEXT Register (offset = B24h) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_PER_CPGMAC0_CONTEXT is shown in Figure 6-125 and described in Table 6-132.

This register contains dedicated CPGMAC0 module context statuses. [warm reset insensitive]

Figure 6-125. PRCM_RM_PER_CPGMAC0_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-132. PRCM_RM_PER_CPGMAC0_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.5.73 PRCM_RM_PER_OCPWP_CONTEXT Register (offset = C24h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_PER_OCPWP_CONTEXT is shown in Figure 6-126 and described in Table 6-133.

This register contains dedicated OCPWP module context statuses. [warm reset insensitive]

Figure 6-126. PRCM_RM_PER_OCPWP_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-133. PRCM_RM_PER_OCPWP_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of PER_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.6 PRCM_PRM_RTC Registers

Table 6-134 lists the memory-mapped registers for the PRCM_PRM_RTC. All register offset addresses not listed in Table 6-134 should be considered as reserved locations and the register contents should not be modified.

Table 6-134. PRCM_PRM_RTC REGISTERS

Offset	Acronym	Register Name	Section
24h	PRCM_RM_RTC_CONTEXT		Section 6.12.6.1

6.12.6.1 PRCM_RM_RTC_CONTEXT Register (offset = 24h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_RTC_CONTEXT is shown in Figure 6-127 and described in Table 6-135.

This register contains dedicated RTC module context statuses. [warm reset insensitive]

Figure 6-127. PRCM_RM_RTC_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-135. PRCM_RM_RTC_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of RTC_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.7 PRM_WKUP Registers

Table 6-136 lists the memory-mapped registers for the PRM_WKUP. All register offset addresses not listed in Table 6-136 should be considered as reserved locations and the register contents should not be modified.

Table 6-136. PRM_WKUP Registers

Offset	Acronym	Register Name	Section
10h	PRCM_RM_WKUP_RSTCTRL		Section 6.12.7.1
14h	PRCM_RM_WKUP_RSTST		Section 6.12.7.2
24h	PRCM_RM_WKUP_DBGSS_CONTEXT		Section 6.12.7.3
124h	PRCM_RM_WKUP_ADC0_CONTEXT		Section 6.12.7.4
224h	PRCM_RM_WKUP_L4WKUP_CONTEXT		Section 6.12.7.5
22Ch	PRCM_RM_WKUP_PROC_CONTEXT		Section 6.12.7.6
234h	PRCM_RM_WKUP_SYNCTIMER_CONTEXT		Section 6.12.7.7
324h	PRCM_RM_WKUP_TIMER0_CONTEXT		Section 6.12.7.8
32Ch	PRCM_RM_WKUP_TIMER1_CONTEXT		Section 6.12.7.9
33Ch	PRCM_RM_WKUP_WDT1_CONTEXT		Section 6.12.7.10
344h	PRCM_RM_WKUP_I2C0_CONTEXT		Section 6.12.7.11

Table 6-136. PRM_WKUP Registers (continued)

Offset	Acronym	Register Name	Section
34Ch	PRCM_RM_WKUP_UART0_CONTEXT		Section 6.12.7.12
36Ch	PRCM_RM_WKUP_GPIO0_CONTEXT		Section 6.12.7.13

6.12.7.1 PRCM_RM_WKUP_RSTCTRL Register (Offset = 10h) [reset = 8h]

Register mask: FFFFFFFFh

PRCM_RM_WKUP_RSTCTRL is shown in Figure 6-128 and described in Table 6-137.

This register controls the release of the ALWAYS ON Domain resets.

Figure 6-128. PRCM_RM_WKUP_RSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED		RESERVED		WKUP_PROC_LRST	RESERVED		
Rreturns0s-0h		Rreturns0s-0h		R/W-1h	Rreturns0s-0h		

Table 6-137. PRCM_RM_WKUP_RSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	Rreturns0s	0h	
5-4	RESERVED	Rreturns0s	0h	
3	WKUP_PROC_LRST	R/W	1h	Assert Reset to WKUP_PROC 0h (R/W) = Reset is cleared for the Wakeup Processor 1h (R/W) = Reset is asserted for the Wakeup Processor by the A9
2-0	RESERVED	Rreturns0s	0h	

6.12.7.2 PRCM_RM_WKUP_RSTST Register (Offset = 14h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_RM_WKUP_RSTST is shown in Figure 6-129 and described in Table 6-138.

This register logs the different reset sources of the ALWON domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]

Figure 6-129. PRCM_RM_WKUP_RSTST Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							
15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							
7	6	5	4	3	2	1	0
ICECRUSHER_WKUP_PROC_RST	EMULATION_WKUP_PROC_RST	WKUP_PROC_LRST	RESERVED				
R/W1toClr-0h	R/W1toClr-0h	R/W1toClr-0h	Rreturns0s-0h				

Table 6-138. PRCM_RM_WKUP_RSTST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	Rreturns0s	0h	
7	ICECRUSHER_WKUP_PROC_RST	R/W1toClr	0h	Wakeup Processor has been reset due to Wakeup Processor ICECRUSHER1 reset event 0h (R/W) = No reset 1h (R/W) = Wakeup Processor has been reset
6	EMULATION_WKUP_PROC_RST	R/W1toClr	0h	Wakeup Processor has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0h (R/W) = No reset 1h (R/W) = Wakeup Processor has been reset
5	WKUP_PROC_LRST	R/W1toClr	0h	Wakeup Processor has been reset 0h (R/W) = No reset 1h (R/W) = Wakeup Processor has been reset
4-0	RESERVED	Rreturns0s	0h	

6.12.7.3 PRCM_RM_WKUP_DBGSS_CONTEXT Register (Offset = 24h) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_WKUP_DBGSS_CONTEXT is shown in Figure 6-130 and described in Table 6-139.

This register contains dedicated DEBUGSS module context statuses.
[warm reset insensitive]

Figure 6-130. PRCM_RM_WKUP_DBGSS_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							
15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_DBGSS_MEM
Rreturns0s-0h							R/W1toClr-1h
7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-139. PRCM_RM_WKUP_DBGSS_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_DBGSS_MEM	R/W1toClr	1h	Specify if memory-based context in DEBUGSS_MEM memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of EMU_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.7.4 PRCM_RM_WKUP_ADC0_CONTEXT Register (Offset = 124h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_WKUP_ADC0_CONTEXT is shown in Figure 6-131 and described in Table 6-140.

This register contains dedicated ADC0 module context statuses.

Figure 6-131. PRCM_RM_WKUP_ADC0_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							
15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							
7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-140. PRCM_RM_WKUP_ADC0_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUP_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.7.5 PRCM_RM_WKUP_L4WKUP_CONTEXT Register (Offset = 224h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_WKUP_L4WKUP_CONTEXT is shown in Figure 6-132 and described in Table 6-141.

This register contains dedicated L4WKUP module context statuses.
[warm reset insensitive]

Figure 6-132. PRCM_RM_WKUP_L4WKUP_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							
15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							
7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-141. PRCM_RM_WKUP_L4WKUP_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUP_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.7.6 PRCM_RM_WKUP_PROC_CONTEXT Register (Offset = 22Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_WKUP_PROC_CONTEXT is shown in Figure 6-133 and described in Table 6-142.

This register contains dedicated WKUP_M3 module context statuses.
[warm reset insensitive]

Figure 6-133. PRCM_RM_WKUP_PROC_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							
15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							
7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-142. PRCM_RM_WKUP_PROC_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUP_PROC_LRST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.7.7 PRCM_RM_WKUP_SYNCTIMER_CONTEXT Register (Offset = 234h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_WKUP_SYNCTIMER_CONTEXT is shown in Figure 6-134 and described in Table 6-143.

This register contains dedicated SYNCTIMER module context statuses.
[warm reset insensitive]

Figure 6-134. PRCM_RM_WKUP_SYNCTIMER_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
Rreturns0s-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
Rreturns0s-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-143. PRCM_RM_WKUP_SYNCTIMER_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUP_SYS_PWRON_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.7.8 PRCM_RM_WKUP_TIMER0_CONTEXT Register (Offset = 324h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_WKUP_TIMER0_CONTEXT is shown in Figure 6-135 and described in Table 6-144.

This register contains dedicated TIMER0 module context statuses.
[warm reset insensitive]

Figure 6-135. PRCM_RM_WKUP_TIMER0_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
Rreturns0s-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
Rreturns0s-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-144. PRCM_RM_WKUP_TIMER0_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUP_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.7.9 PRCM_RM_WKUP_TIMER1_CONTEXT Register (Offset = 32Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_WKUP_TIMER1_CONTEXT is shown in Figure 6-136 and described in Table 6-145.

This register contains dedicated TIMER1 module context statuses.
[warm reset insensitive]

Figure 6-136. PRCM_RM_WKUP_TIMER1_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
Rreturns0s-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
Rreturns0s-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-145. PRCM_RM_WKUP_TIMER1_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUP_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.7.10 PRCM_RM_WKUP_WDT1_CONTEXT Register (Offset = 33Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_WKUP_WDT1_CONTEXT is shown in Figure 6-137 and described in Table 6-146.

This register contains dedicated WDT1 module context statuses.
[warm reset insensitive]

Figure 6-137. PRCM_RM_WKUP_WDT1_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
Rreturns0s-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
Rreturns0s-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-146. PRCM_RM_WKUP_WDT1_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUP_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.7.11 PRCM_RM_WKUP_I2C0_CONTEXT Register (Offset = 344h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_WKUP_I2C0_CONTEXT is shown in Figure 6-138 and described in Table 6-147.

This register contains dedicated I2C0 module context statuses.
[warm reset insensitive]

Figure 6-138. PRCM_RM_WKUP_I2C0_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							
15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							
7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-147. PRCM_RM_WKUP_I2C0_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUP_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.7.12 PRCM_RM_WKUP_UART0_CONTEXT Register (Offset = 34Ch) [reset = 101h]

Register mask: FFFFFFFFh

PRCM_RM_WKUP_UART0_CONTEXT is shown in Figure 6-139 and described in Table 6-148.

This register contains dedicated UART0 module context statuses.
[warm reset insensitive]

Figure 6-139. PRCM_RM_WKUP_UART0_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							
15	14	13	12	11	10	9	8
RESERVED							LOSTMEM_RETAINED_BANK
Rreturns0s-0h							R/W1toClr-1h
7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-148. PRCM_RM_WKUP_UART0_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	Rreturns0s	0h	
8	LOSTMEM_RETAINED_BANK	R/W1toClr	1h	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost
7-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUP_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.7.13 PRCM_RM_WKUP_GPIO0_CONTEXT Register (Offset = 36Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_RM_WKUP_GPIO0_CONTEXT is shown in Figure 6-140 and described in Table 6-149.

This register contains dedicated GPIO0 module context statuses.
[warm reset insensitive]

Figure 6-140. PRCM_RM_WKUP_GPIO0_CONTEXT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							
15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							
7	6	5	4	3	2	1	0
RESERVED							LOSTCONTEXT_DFF
Rreturns0s-0h							R/W1toClr-1h

Table 6-149. PRCM_RM_WKUP_GPIO0_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	Rreturns0s	0h	
0	LOSTCONTEXT_DFF	R/W1toClr	1h	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUP_DOM_RST signal) 0h (R/W) = Context has been maintained 1h (R/W) = Context has been lost

6.12.8 PRCM_PRM_IRQ Registers

Table 6-150 lists the memory-mapped registers for the PRCM_PRM_IRQ. All register offset addresses not listed in Table 6-150 should be considered as reserved locations and the register contents should not be modified.

Table 6-150. PRCM_PRM_IRQ Registers

Offset	Acronym	Register Name	Section
0h	PRCM_REVISION		Section 6.12.8.1
4h	PRCM_PRM_IRQSTS_MPU		Section 6.12.8.2
8h	PRCM_PRM_IRQEN_MPU		Section 6.12.8.3
Ch	PRCM_PRM_IRQSTS_WKUP_PROC		Section 6.12.8.4
10h	PRCM_PRM_IRQEN_WKUP_PROC		Section 6.12.8.5

6.12.8.1 PRCM_REVISION Register (Offset = 0h) [reset = 40000400h]

Register mask: FFFFFFFFh

PRCM_REVISION is shown in Figure 6-141 and described in Table 6-151.

Return to Summary Table.

This register contains the IP revision code for the PRCM

Figure 6-141. PRCM_REVISION Register

31	30	29	28	27	26	25	24
SCHEME		RESERVED		FUNC			
Rreturns-1h		R-0-0h		R-0-0h			
23	22	21	20	19	18	17	16
FUNC							
R-0-0h							
15	14	13	12	11	10	9	8
R_RTL					X_MAJOR		
R-0-0h					Rreturns-4h		
7	6	5	4	3	2	1	0
CUSTOM		Y_MINOR					
R-0-0h		Rreturns-0h					

Table 6-151. PRCM_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	Rreturns	1h	
29-28	RESERVED	R-0	0h	
27-16	FUNC	R-0	0h	
15-11	R_RTL	R-0	0h	
10-8	X_MAJOR	Rreturns	4h	
7-6	CUSTOM	R-0	0h	
5-0	Y_MINOR	Rreturns	0h	

6.12.8.2 PRCM_PRM_IRQSTS_MPU Register (Offset = 4h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_PRM_IRQSTS_MPU is shown in Figure 6-142 and described in Table 6-152.

Return to Summary Table.

This register provides status on MPU interrupt events. An event is logged whether interrupt generation for the event is enabled or not. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.

Figure 6-142. PRCM_PRM_IRQSTS_MPU Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							DPLL_EXTDEV_RECAL_ST
R-0-0h							R/W1C-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DPLL_PER_RECAL_ST	DPLL_DDR_RECAL_ST	DPLL_DISP_RECAL_ST	DPLL_CORE_RECAL_ST	DPLL_MPU_RECAL_ST	FORCEWKUP_ST	IO_ST	TRANSITION_ST
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							FREQ_UPDATE_ST
R-0-0h							R/W1C-0h

Table 6-152. PRCM_PRM_IRQSTS_MPU Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R-0	0h	
16	DPLL_EXTDEV_RECAL_ST	R/W1C	0h	interrupt status for extdev dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
15	DPLL_PER_RECAL_ST	R/W1C	0h	interrupt status for usb dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
14	DPLL_DDR_RECAL_ST	R/W1C	0h	interrupt status for ddr dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
13	DPLL_DISP_RECAL_ST	R/W1C	0h	interrupt status for disp dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
12	DPLL_CORE_RECAL_ST	R/W1C	0h	interrupt status for core dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
11	DPLL_MPU_RECAL_ST	R/W1C	0h	interrupt status for mpu dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
10	FORCEWKUP_ST	R/W1C	0h	Software supervised wakeup completed event interrupt status 0h (R/W) = No interrupt 1h (R/W) = Interrupt is pending

Table 6-152. PRCM_PRM_IRQSTS_MPU Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	IO_ST	R/W1C	0h	IO pad event interrupt status. 0h (R/W) = No interrupt 1h (R/W) = Interrupt is pending
8	TRANSITION_ST	R/W1C	0h	Software supervised transition completed event interrupt status (any domain) 0h (R/W) = No interrupt 1h (R/W) = Interrupt is pending
7-1	RESERVED	R-0	0h	
0	FREQ_UPDATE_ST	R/W1C	0h	Frequency Update interrupt status. 0h (R/W) = No interrupt 1h (R/W) = Interrupt is pending

6.12.8.3 PRCM_PRM_IRQEN_MPU Register (Offset = 8h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_PRM_IRQEN_MPU is shown in Figure 6-143 and described in Table 6-153.

Return to Summary Table.

This register is used to enable and disable events used to trigger MPU interrupt activation.

Figure 6-143. PRCM_PRM_IRQEN_MPU Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							DPLL_EXTDEV_RECAL_EN
R-0-0h							R/W-0h
15	14	13	12	11	10	9	8
DPLL_DISP_RECAL_EN	DPLL_DDR_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_CORE_RECAL_EN	DPLL_MPU_RECAL_EN	FORCEWKUP_EN	IO_EN	TRANSITION_EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED		RESERVED					FREQ_UPDATE_EN
R-0-0h		R-0-0h					R/W-0h

Table 6-153. PRCM_PRM_IRQEN_MPU Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R-0	0h	
16	DPLL_EXTDEV_RECAL_EN	R/W	0h	Interrupt enable for extdev dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
15	DPLL_DISP_RECAL_EN	R/W	0h	Interrupt enable for disp dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
14	DPLL_DDR_RECAL_EN	R/W	0h	Interrupt enable for ddr dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
13	DPLL_PER_RECAL_EN	R/W	0h	Interrupt enable for usb dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
12	DPLL_CORE_RECAL_EN	R/W	0h	Interrupt enable for core dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
11	DPLL_MPU_RECAL_EN	R/W	0h	Interrupt enable for mpu dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
10	FORCEWKUP_EN	R/W	0h	Software supervised Force Wakeup completed event interrupt enable 0h (R/W) = Interrupt is masked 1h (R/W) = Interrupt is enabled
9	IO_EN	R/W	0h	IO pad event interrupt enable 0h (R/W) = Interrupt is masked 1h (R/W) = Interrupt is enabled

Table 6-153. PRCM_PRM_IRQEN_MPU Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	TRANSITION_EN	R/W	0h	Software supervised transition completed event interrupt enable (any domain) 0h (R/W) = Interrupt is masked 1h (R/W) = Interrupt is enabled
7-6	RESERVED	R-0	0h	
5-1	RESERVED	R-0	0h	
0	FREQ_UPDATE_EN	R/W	0h	Frequency Update interrupt enable. 0h (R/W) = Interrupt is masked 1h (R/W) = Interrupt is enabled

6.12.8.4 PRCM_PRM_IRQSTS_WKUP_PROC Register (Offset = Ch) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_PRM_IRQSTS_WKUP_PROC is shown in Figure 6-144 and described in Table 6-154.

Return to Summary Table.

This register provides status on Wakeup Processor interrupt events. An event is logged whether interrupt generation for the event is enabled or not. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.

Figure 6-144. PRCM_PRM_IRQSTS_WKUP_PROC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							DPLL_EXTDEV_RECAL_ST
R-0-0h							R/W1C-0h
15	14	13	12	11	10	9	8
DPLL_PER_RECAL_ST	DPLL_DDR_RECAL_ST	DPLL_DISP_RECAL_ST	DPLL_CORE_RECAL_ST	DPLL_MPU_RECAL_ST	FORCEWKUP_ST	IO_ST	TRANSITION_ST
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
7	6	5	4	3	2	1	0
RESERVED							FREQ_UPDATE_ST
R-0-0h							R/W1C-0h

Table 6-154. PRCM_PRM_IRQSTS_WKUP_PROC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R-0	0h	
16	DPLL_EXTDEV_RECAL_ST	R/W1C	0h	interrupt status for extdev dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
15	DPLL_PER_RECAL_ST	R/W1C	0h	interrupt status for usb dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
14	DPLL_DDR_RECAL_ST	R/W1C	0h	interrupt status for ddr dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
13	DPLL_DISP_RECAL_ST	R/W1C	0h	interrupt status for disp dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
12	DPLL_CORE_RECAL_ST	R/W1C	0h	interrupt status for core dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
11	DPLL_MPU_RECAL_ST	R/W1C	0h	interrupt status for mpu dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
10	FORCEWKUP_ST	R/W1C	0h	Software supervised wakeup completed event interrupt status 0h (R/W) = No interrupt 1h (R/W) = Interrupt is pending

Table 6-154. PRCM_PRM_IRQSTS_WKUP_PROC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	IO_ST	R/W1C	0h	IO pad event interrupt status. 0h (R/W) = No interrupt 1h (R/W) = Interrupt is pending
8	TRANSITION_ST	R/W1C	0h	Software supervised transition completed event interrupt status (any domain) 0h (R/W) = No interrupt 1h (R/W) = Interrupt is pending
7-1	RESERVED	R-0	0h	
0	FREQ_UPDATE_ST	R/W1C	0h	Frequency Update interrupt status. 0h (R/W) = No interrupt 1h (R/W) = Interrupt is pending

6.12.8.5 PRCM_PRM_IRQEN_WKUP_PROC Register (Offset = 10h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_PRM_IRQEN_WKUP_PROC is shown in Figure 6-145 and described in Table 6-155.

Return to Summary Table.

This register is used to enable and disable events used to trigger Wakeup Processor interrupt activation.

Figure 6-145. PRCM_PRM_IRQEN_WKUP_PROC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							DPLL_EXTDEV_RECAL_EN
R-0-0h							R/W-0h
15	14	13	12	11	10	9	8
DPLL_DISP_RECAL_EN	DPLL_DDR_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_CORE_RECAL_EN	DPLL_MPU_RECAL_EN	FORCEWKUP_EN	IO_EN	TRANSITION_EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED		RESERVED					FREQ_UPDATE_EN
R-0-0h		R-0-0h					R/W-0h

Table 6-155. PRCM_PRM_IRQEN_WKUP_PROC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R-0	0h	
16	DPLL_EXTDEV_RECAL_EN	R/W	0h	Interrupt enable for extdev dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
15	DPLL_DISP_RECAL_EN	R/W	0h	Interrupt enable for disp dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
14	DPLL_DDR_RECAL_EN	R/W	0h	Interrupt enable for ddr dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
13	DPLL_PER_RECAL_EN	R/W	0h	Interrupt enable for usb dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
12	DPLL_CORE_RECAL_EN	R/W	0h	Interrupt enable for core dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
11	DPLL_MPU_RECAL_EN	R/W	0h	Interrupt enable for mpu dpll recaliberation 0h (R/W) = Disables dpll recaliberation 1h (R/W) = ENAbles dpll recaliberation
10	FORCEWKUP_EN	R/W	0h	Software supervised Froce Wakeup completed event interrupt enable 0h (R/W) = Interrupt is masked 1h (R/W) = Interrupt is enabled
9	IO_EN	R/W	0h	IO pad event interrupt enable 0h (R/W) = Interrupt is masked 1h (R/W) = Interrupt is enabled

Table 6-155. PRCM_PRM_IRQEN_WKUP_PROC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	TRANSITION_EN	R/W	0h	Software supervised transition completed event interrupt enable (any domain) 0h (R/W) = Interrupt is masked 1h (R/W) = Interrupt is enabled
7-6	RESERVED	R-0	0h	
5-1	RESERVED	R-0	0h	
0	FREQ_UPDATE_EN	R/W	0h	Frequency Update interrupt enable. 0h (R/W) = Interrupt is masked 1h (R/W) = Interrupt is enabled

6.13 Clock Module Registers

6.13.1 PRCM_CM_CEFUSE Registers

Table 6-156 lists the memory-mapped registers for the PRCM_CM_CEFUSE. All register offset addresses not listed in Table 6-156 should be considered as reserved locations and the register contents should not be modified.

Table 6-156. PRCM_CM_CEFUSE REGISTERS

Offset	Acronym	Register Name	Section
0h	PRCM_CM_CEFUSE_CLKSTCTRL		Section 6.13.1.1
20h	PRCM_CM_CEFUSE_CLKCTRL		Section 6.13.1.2

6.13.1.1 PRCM_CM_CEFUSE_CLKSTCTRL Register (offset = 0h) [reset = 2h]

Register mask: FFFFFFFFh

PRCM_CM_CEFUSE_CLKSTCTRL is shown in Figure 6-146 and described in Table 6-157.

This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

Figure 6-146. PRCM_CM_CEFUSE_CLKSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED						CLKACTIVITY_CUST_EFUSE_SYS_CLK	CLKACTIVITY_L4_CEFUSE_GICLK
Rreturns0s-0h						R-0h	R-0h

7	6	5	4	3	2	1	0
RESERVED						CLKTRCTRL	
Rreturns0s-0h						R/W-2h	

Table 6-157. PRCM_CM_CEFUSE_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	Rreturns0s	0h	
9	CLKACTIVITY_CUST_EFUSE_SYS_CLK	R	0h	This field indicates the state of the Cust_Efuse_SYSCLK clock input of the domain. [warm reset insensitive] 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
8	CLKACTIVITY_L4_CEFUSE_GICLK	R	0h	This field indicates the state of the L4_CEFUSE_GCLK clock input of the domain. [warm reset insensitive] 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
7-2	RESERVED	Rreturns0s	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the clock domain in customer efuse power domain. 0h (R/W) = NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1h (R/W) = SW_SLEEP: Start a software forced sleep transition on the domain. 2h (R/W) = SW_WKUP: Start a software forced wake-up transition on the domain. 3h (R/W) = Reserved

6.13.1.2 PRCM_CM_CEFUSE_CLKCTRL Register (offset = 20h) [reset = 30000h]

Register mask: FFFFFFFFh

PRCM_CM_CEFUSE_CLKCTRL is shown in Figure 6-147 and described in Table 6-158.

This register manages the CEFUSE clocks.

Figure 6-147. PRCM_CM_CEFUSE_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED						IDLEST	
Rreturns0s-0h						R-3h	

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
Rreturns0s-0h						R/W-0h	

Table 6-158. PRCM_CM_CEFUSE_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	Rreturns0s	0h	
17-16	IDLEST	R	3h	Module idle status. [warm reset insensitive] 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	Rreturns0s	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.2 PRCM_CM_DEVICE Registers

Table 6-159 lists the memory-mapped registers for the PRCM_CM_DEVICE. All register offset addresses not listed in Table 6-159 should be considered as reserved locations and the register contents should not be modified.

Table 6-159. PRCM_CM_DEVICE REGISTERS

Offset	Acronym	Register Name	Section
0h	PRCM_CM_CLKOUT1_CTRL		Section 6.13.2.1
4h	PRCM_CM_DLL_CTRL		Section 6.13.2.2
8h	PRCM_CM_CLKOUT2_CTRL		Section 6.13.2.3

6.13.2.1 PRCM_CM_CLKOUT1_CTRL Register (offset = 0h) [reset = 800000h]

Register mask: FFFFFFFFh

PRCM_CM_CLKOUT1_CTRL is shown in Figure 6-148 and described in Table 6-160.

This register provides the control over CLKOUT1 output

Figure 6-148. PRCM_CM_CLKOUT1_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							CLKOUT_32KSEL
R-0h							R/W-0h

23	22	21	20	19	18	17	16
CLKOUT1EN	RESERVED	CLKOUT1SEL0DIV		RESERVED		CLKOUT1SOURCE	
R/W-1h	R-0h	R/W-0h		R-0h		R/W-0h	

15	14	13	12	11	10	9	8
RESERVED					CLKOUT1SEL2DIV2		
Rreturns0s-0h					R/W-0h		

7	6	5	4	3	2	1	0
RESERVED	CLKOUT1SEL2DIV1			RESERVED	CLKOUT1SEL2SOURCE		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 6-160. PRCM_CM_CLKOUT1_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	
24	CLKOUT_32KSEL	R/W	0h	This bit controls the 32KHz clock source selection for CLKOUT1 and CLKOUT2 output clocks 0h (R/W) = Reserved. 1h (R/W) = Select 32KHz clock source.
23	CLKOUT1EN	R/W	1h	This bit controls the external clock CLKOUT1 activity 0h (R/W) = SYS_CLKOUT1 is disabled 1h (R/W) = SYS_CLKOUT1 is enabled
22	RESERVED	R	0h	
21-20	CLKOUT1SEL0DIV	R/W	0h	This field controls the external clock CLKOUT1 divison factor, when CLKOUT1SOURCE=SEL0 0h (R/W) = Divide CLKOUT1 by 1 1h (R/W) = Divide CLKOUT1 by 2 2h (R/W) = Divide CLKOUT1 by 3 3h (R/W) = Divide CLKOUT1 by 4
19-18	RESERVED	R	0h	
17-16	CLKOUT1SOURCE	R/W	0h	This field selects the external output CLKOUT1 clock source 0h (R/W) = Selects Master Osc[CLK_M_OSC]. 1h (R/W) = Select 32KHz clock source. 2h (R/W) = Selects clk based on CLKOUT1SEL2SOURCE, CLKOUT1SEL2DIV1 and CLKOUT1SEL2DIV2 3h (R/W) = Selects clock from DPLL_EXTDEV
15-11	RESERVED	Rreturns0s	0h	

Table 6-160. PRCM_CM_CLKOUT1_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-8	CLKOUT1SEL2DIV2	R/W	0h	This field controls the external clock CLKOUT1 first divison factor, when CLKOUT1SOURCE=SEL2 0h (R/W) = Divide clock further by 1 1h (R/W) = Divide clock further by 2 2h (R/W) = Divide clock further by 4 3h (R/W) = Divide clock further by 8 4h (R/W) = Divide clock further by 16 5h (R/W) = Divide clock further by 32 6h (R) = Reserved 7h (R) = Reserved
7	RESERVED	R	0h	
6-4	CLKOUT1SEL2DIV1	R/W	0h	This field controls the external clock CLKOUT1 first divison factor, when CLKOUT1SOURCE=SEL2 0h (R/W) = SYS_CLKOUT2/1 1h (R/W) = SYS_CLKOUT2/2 2h (R/W) = SYS_CLKOUT2/3 3h (R/W) = SYS_CLKOUT2/4 4h (R/W) = SYS_CLKOUT2/5 5h (R/W) = SYS_CLKOUT2/6 6h (R/W) = SYS_CLKOUT2/7 7h (R/W) = SYS_CLKOUT2/8
3	RESERVED	R	0h	
2-0	CLKOUT1SEL2SOURCE	R/W	0h	This field selects the CLKOUT1, when CLKOUT1SOURCE=SEL2, 0h (R/W) = Selects 32KHz clock source. Note: CLKOUT_32KSEL should be set to 1. 1h (R/W) = Selects L3F_CLK clock. 2h (R/W) = Selects DDR_PHY_Clk 3h (R/W) = Selects192Mhz clock from PER PLL[PER_CLKOUT_M2] 4h (R/W) = Selects LCD Pixel Clock[PIXEL_CLK] 5h (R/W) = Selects MPU_PLL_CLKOUT

6.13.2.2 PRCM_CM_DLL_CTRL Register (offset = 4h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_CM_DLL_CTRL is shown in Figure 6-149 and described in Table 6-161.

Special register for DLL control

Figure 6-149. PRCM_CM_DLL_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED					DLL_READYST	DLL_RESET	DLL_OVERRIDE
Rreturns0s-0h					R-0h	R/W-0h	R/W-1h

Table 6-161. PRCM_CM_DLL_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	Rreturns0s	0h	
2	DLL_READYST	R	0h	Gives the DLL ready status. It is the AND of individual DLL_READY signals. 0h (R) = 0 1h (R) = 1
1	DLL_RESET	R/W	0h	Controls DLL Reset. Applicable only for DVFS. This reset is to be used only during DVFS when DLL frequency is being changed 0h (R/W) = No Reset to DLL 1h (R/W) = Reset DLL
0	DLL_OVERRIDE	R/W	1h	Control if DLL lock and code outputs are overriden or not 0h (R/W) = Lock and code outputs are not overriden 1h (R/W) = Lock output is overriden to '1' and code output is overriden with a value coming from control module.

6.13.2.3 PRCM_CM_CLKOUT2_CTRL Register (offset = 8h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_CM_CLKOUT2_CTRL is shown in Figure 6-150 and described in Table 6-162.

This register provides the control over CLKOUT2 output

Figure 6-150. PRCM_CM_CLKOUT2_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							CLKOUT2EN
R-0h							R/W-0h

15	14	13	12	11	10	9	8
RESERVED					CLKOUT2POSTDIV		
R-0h					R/W-0h		

7	6	5	4	3	2	1	0
RESERVED	CLKOUT2DIV			RESERVED	CLKOUT2SOURCE		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 6-162. PRCM_CM_CLKOUT2_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	CLKOUT2EN	R/W	0h	This bit controls the external clock activity 0h (R/W) = SYS_CLKOUT2 is disabled 1h (R/W) = SYS_CLKOUT2 is enabled
15-11	RESERVED	R	0h	
10-8	CLKOUT2POSTDIV	R/W	0h	THis field controls the external clock CLKOUT2 post divison factor. This division factor will divide the clock coming out of divide controlled by CLKOUT2DIV bit-field 0h (R/W) = Divide clock further by 1 1h (R/W) = Divide clock further by 2 2h (R/W) = Divide clock further by 4 3h (R/W) = Divide clock further by 8 4h (R/W) = Divide clock further by 16 5h (R/W) = Divide clock further by 32 6h (R) = Reserved 7h (R) = Reserved
7	RESERVED	R	0h	
6-4	CLKOUT2DIV	R/W	0h	THis field controls the external clock CLKOUT2 divison factor 0h (R/W) = SYS_CLKOUT2/1 1h (R/W) = SYS_CLKOUT2/2 2h (R/W) = SYS_CLKOUT2/3 3h (R/W) = SYS_CLKOUT2/4 4h (R/W) = SYS_CLKOUT2/5 5h (R/W) = SYS_CLKOUT2/6 6h (R/W) = SYS_CLKOUT2/7 7h (R/W) = SYS_CLKOUT2/8
3	RESERVED	R	0h	

Table 6-162. PRCM_CM_CLKOUT2_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	CLKOUT2SOURCE	R/W	0h	This field selects the external output CLKOUT2 clock source 0h (R/W) = Selects 32KHz clock source 1h (R/W) = Selects L3F_CLK Clock 2h (R/W) = Selects DDR_PHY_Clk 3h (R/W) = Selects192Mhz clock from PER PLL[PER_CLKOUT_M2] 4h (R/W) = Selects LCD Pixel Clock[PIXEL_CLK] 5h (R/W) = Selects MPU_PLL_CLKOUT 6h (R/W) = Selects DPLL_EXTDEV clkout

6.13.3 CM_DPLL Registers

Table 6-163 lists the memory-mapped registers for the CM_DPLL. All register offset addresses not listed in Table 6-163 should be considered as reserved locations and the register contents should not be modified.

Table 6-163. CM_DPLL Registers

Offset	Acronym	Register Name	Section
0h	PRCM_CM_DPLL_DPLL_CLKSEL_TIMER1_CLK		Section 6.13.3.1
4h	PRCM_CM_DPLL_CLKSEL_TIMER2_CLK		Section 6.13.3.2
8h	PRCM_CM_DPLL_CLKSEL_TIMER3_CLK		Section 6.13.3.3
Ch	PRCM_CM_DPLL_CLKSEL_TIMER4_CLK		Section 6.13.3.4
10h	PRCM_CM_DPLL_CLKSEL_TIMER5_CLK		Section 6.13.3.5
14h	PRCM_CM_DPLL_CLKSEL_TIMER6_CLK		Section 6.13.3.6
18h	PRCM_CM_DPLL_CLKSEL_TIMER7_CLK		Section 6.13.3.7
1Ch	PRCM_CM_DPLL_CLKSEL_TIMER8_CLK		Section 6.13.3.8
20h	PRCM_CM_DPLL_CLKSEL_TIMER9_CLK		Section 6.13.3.9
24h	PRCM_CM_DPLL_CLKSEL_TIMER10_CLK		Section 6.13.3.10
28h	PRCM_CM_DPLL_CLKSEL_TIMER11_CLK		Section 6.13.3.11
2Ch	PRCM_CM_DPLL_CLKSEL_WDT1_CLK		Section 6.13.3.12
30h	PRCM_CM_DPLL_CLKSEL_SYNCTIMER_CLK		Section 6.13.3.13
34h	PRCM_CM_DPLL_CLKSEL_MAC_CLK		Section 6.13.3.14
38h	PRCM_CM_DPLL_CLKSEL_CPTS_RFT_CLK		Section 6.13.3.15
3Ch	PRCM_CM_DPLL_CLKSEL_GFX_FCLK		Section 6.13.3.16
40h	PRCM_CM_DPLL_CLKSEL_GPIO0_DBCLK		Section 6.13.3.17
48h	PRCM_CM_CLKSEL_PRU_ICSS_OCP_CLK		Section 6.13.3.18
4Ch	PRCM_CM_CLKSEL_ADC1_CLK		Section 6.13.3.19
50h	PRCM_CM_DPLL_CLKSEL_DLL_AGING_CLK		Section 6.13.3.20

Table 6-163. CM_DPLL Registers (continued)

Offset	Acronym	Register Name	Section
60h	PRCM_CM_DPLL_CLKSEL_USBPHY32KHZ_GCLK		Section 6.13.3.21

6.13.3.1 PRCM_CM_DPLL_DPLL_CLKSEL_TIMER1_CLK Register (Offset = 0h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_CM_DPLL_DPLL_CLKSEL_TIMER1_CLK is shown in Figure 6-151 and described in Table 6-164.

Return to Summary Table.

Selects the Mux select line for TIMER1 clock [warm reset insensitive]

Figure 6-151. PRCM_CM_DPLL_DPLL_CLKSEL_TIMER1_CLK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
R-0-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													CLKSEL		
R-0-0h													R/W-0h		

Table 6-164. PRCM_CM_DPLL_DPLL_CLKSEL_TIMER1_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RSVD	R-0	0h	
2-0	CLKSEL	R/W	0h	Selects the Mux select line for DMTIMER_1MS clock [warm reset insensitive] 0h (R/W) = Select CLK_M_OSC clock 1h (R/W) = Select CLK_32KHZ clock from PER_PLL divided clock 2h (R/W) = Select TCLKIN clock 3h (R/W) = Select CLK_RC32K clock 4h (R/W) = Selects the 32KHz clock from RTC 32KHz Crystal Osc 5h (R/W) = Reserved

6.13.3.2 PRCM_CM_DPLL_CLKSEL_TIMER2_CLK Register (Offset = 4h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_CM_DPLL_CLKSEL_TIMER2_CLK is shown in Figure 6-152 and described in Table 6-165.

Return to Summary Table.

Selects the Mux select line for TIMER2 clock
[warm reset insensitive]

Figure 6-152. PRCM_CM_DPLL_CLKSEL_TIMER2_CLK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														CLKSEL	
R-0-0h														R/W-1h	

Table 6-165. PRCM_CM_DPLL_CLKSEL_TIMER2_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	
1-0	CLKSEL	R/W	1h	Selects the Mux select line for TIMER2 clock [warm reset insensitive] 0h (R/W) = Select TCLKIN clock 1h (R/W) = Select CLK_M_OSC clock 2h (R/W) = Select CLK_32KHZ clock 3h (R/W) = Reserved

6.13.3.3 PRCM_CM_DPLL_CLKSEL_TIMER3_CLK Register (Offset = 8h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_CM_DPLL_CLKSEL_TIMER3_CLK is shown in Figure 6-153 and described in Table 6-166.

Return to Summary Table.

Selects the Mux select line for TIMER3 clock
[warm reset insensitive]

Figure 6-153. PRCM_CM_DPLL_CLKSEL_TIMER3_CLK Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="2">CLKSEL</td></tr>
<tr><td colspan="14">R-0-0h</td><td colspan="2">R/W-1h</td></tr>
</table>

Table 6-166. PRCM_CM_DPLL_CLKSEL_TIMER3_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	
1-0	CLKSEL	R/W	1h	Selects the Mux select line for TIMER3 clock [warm reset insensitive] 0h (R/W) = Select TCLKIN clock 1h (R/W) = Select CLK_M_OSC clock 2h (R/W) = Select CLK_32KHZ clock 3h (R/W) = Reserved

6.13.3.4 PRCM_CM_DPLL_CLKSEL_TIMER4_CLK Register (Offset = Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_CM_DPLL_CLKSEL_TIMER4_CLK is shown in Figure 6-154 and described in Table 6-167.

Return to Summary Table.

Selects the Mux select line for TIMER4 clock
[warm reset insensitive]

Figure 6-154. PRCM_CM_DPLL_CLKSEL_TIMER4_CLK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														CLKSEL	
R-0-0h														R/W-1h	

Table 6-167. PRCM_CM_DPLL_CLKSEL_TIMER4_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	
1-0	CLKSEL	R/W	1h	Selects the Mux select line for TIMER4 clock [warm reset insensitive] 0h (R/W) = Select TCLKIN clock 1h (R/W) = Select CLK_M_OSC clock 2h (R/W) = Select CLK_32KHZ clock 3h (R/W) = Reserved

6.13.3.5 PRCM_CM_DPLL_CLKSEL_TIMER5_CLK Register (Offset = 10h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_CM_DPLL_CLKSEL_TIMER5_CLK is shown in Figure 6-155 and described in Table 6-168.

Return to Summary Table.

Selects the Mux select line for TIMER5 clock
[warm reset insensitive]

Figure 6-155. PRCM_CM_DPLL_CLKSEL_TIMER5_CLK Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="2">CLKSEL</td></tr>
<tr><td colspan="14">R-0-0h</td><td colspan="2">R/W-1h</td></tr>
</table>

Table 6-168. PRCM_CM_DPLL_CLKSEL_TIMER5_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	
1-0	CLKSEL	R/W	1h	Selects the Mux select line for TIMER5 clock [warm reset insensitive] 0h (R/W) = Select TCLKIN clock 1h (R/W) = Select CLK_M_OSC clock 2h (R/W) = Select CLK_32KHZ clock 3h (R/W) = Reserved

6.13.3.6 PRCM_CM_DPLL_CLKSEL_TIMER6_CLK Register (Offset = 14h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_CM_DPLL_CLKSEL_TIMER6_CLK is shown in Figure 6-156 and described in Table 6-169.

Return to Summary Table.

Selects the Mux select line for TIMER6 clock
[warm reset insensitive]

Figure 6-156. PRCM_CM_DPLL_CLKSEL_TIMER6_CLK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED														CLKSEL	
R-0-0h														R/W-1h	

Table 6-169. PRCM_CM_DPLL_CLKSEL_TIMER6_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	
1-0	CLKSEL	R/W	1h	Selects the Mux select line for TIMER6 clock [warm reset insensitive] 0h (R/W) = Select TCLKIN clock 1h (R/W) = Select CLK_M_OSC clock 2h (R/W) = Select CLK_32KHZ clock 3h (R/W) = Reserved

6.13.3.7 PRCM_CM_DPLL_CLKSEL_TIMER7_CLK Register (Offset = 18h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_CM_DPLL_CLKSEL_TIMER7_CLK is shown in Figure 6-157 and described in Table 6-170.

Return to Summary Table.

Selects the Mux select line for TIMER7 clock
[warm reset insensitive]

Figure 6-157. PRCM_CM_DPLL_CLKSEL_TIMER7_CLK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED														CLKSEL	
R-0-0h														R/W-1h	

Table 6-170. PRCM_CM_DPLL_CLKSEL_TIMER7_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	
1-0	CLKSEL	R/W	1h	Selects the Mux select line for TIMER7 clock [warm reset insensitive] 0h (R/W) = Select TCLKIN clock 1h (R/W) = Select CLK_M_OSC clock 2h (R/W) = Select CLK_32KHZ clock 3h (R/W) = Reserved

6.13.3.8 PRCM_CM_DPLL_CLKSEL_TIMER8_CLK Register (Offset = 1Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_CM_DPLL_CLKSEL_TIMER8_CLK is shown in Figure 6-158 and described in Table 6-171.

Return to Summary Table.

Selects the Mux select line for TIMER8 clock
[warm reset insensitive]

Figure 6-158. PRCM_CM_DPLL_CLKSEL_TIMER8_CLK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
R-0-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														CLKSEL	
R-0-0h														R/W-1h	

Table 6-171. PRCM_CM_DPLL_CLKSEL_TIMER8_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RSVD	R-0	0h	
1-0	CLKSEL	R/W	1h	Selects the Mux select line for TIMER8 clock [warm reset insensitive] 0h (R/W) = Select TCLKIN clock 1h (R/W) = Select CLK_M_OSC clock 2h (R/W) = Select CLK_32KHZ clock 3h (R/W) = Reserved

6.13.3.9 PRCM_CM_DPLL_CLKSEL_TIMER9_CLK Register (Offset = 20h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_CM_DPLL_CLKSEL_TIMER9_CLK is shown in Figure 6-159 and described in Table 6-172.

Return to Summary Table.

Selects the Mux select line for TIMER9 clock
[warm reset insensitive]

Figure 6-159. PRCM_CM_DPLL_CLKSEL_TIMER9_CLK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
R-0-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														CLKSEL	
R-0-0h														R/W-1h	

Table 6-172. PRCM_CM_DPLL_CLKSEL_TIMER9_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RSVD	R-0	0h	
1-0	CLKSEL	R/W	1h	Selects the Mux select line for TIMER9 clock [warm reset insensitive] 0h (R/W) = Select TCLKIN clock 1h (R/W) = Select CLK_M_OSC clock 2h (R/W) = Select CLK_32KHZ clock 3h (R/W) = Reserved

6.13.3.10 PRCM_CM_DPLL_CLKSEL_TIMER10_CLK Register (Offset = 24h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_CM_DPLL_CLKSEL_TIMER10_CLK is shown in Figure 6-160 and described in Table 6-173.

Return to Summary Table.

Selects the Mux select line for TIMER10 clock
[warm reset insensitive]

Figure 6-160. PRCM_CM_DPLL_CLKSEL_TIMER10_CLK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
R-0-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														CLKSEL	
R-0-0h														R/W-1h	

Table 6-173. PRCM_CM_DPLL_CLKSEL_TIMER10_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RSVD	R-0	0h	
1-0	CLKSEL	R/W	1h	Selects the Mux select line for TIMER10 clock [warm reset insensitive] 0h (R/W) = Select TCLKIN clock 1h (R/W) = Select CLK_M_OSC clock 2h (R/W) = Select CLK_32KHZ clock 3h (R/W) = Reserved

6.13.3.11 PRCM_CM_DPLL_CLKSEL_TIMER11_CLK Register (Offset = 28h) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_CM_DPLL_CLKSEL_TIMER11_CLK is shown in Figure 6-161 and described in Table 6-174.

Return to Summary Table.

Selects the Mux select line for TIMER11 clock
[warm reset insensitive]

Figure 6-161. PRCM_CM_DPLL_CLKSEL_TIMER11_CLK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
R-0-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														CLKSEL	
R-0-0h														R/W-1h	

Table 6-174. PRCM_CM_DPLL_CLKSEL_TIMER11_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RSVD	R-0	0h	
1-0	CLKSEL	R/W	1h	Selects the Mux select line for TIMER11 clock [warm reset insensitive] 0h (R/W) = Select TCLKIN clock 1h (R/W) = Select CLK_M_OSC clock 2h (R/W) = Select CLK_32KHZ clock 3h (R/W) = Reserved

6.13.3.12 PRCM_CM_DPLL_CLKSEL_WDT1_CLK Register (Offset = 2Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_CM_DPLL_CLKSEL_WDT1_CLK is shown in Figure 6-162 and described in Table 6-175.

Return to Summary Table.

Selects the Mux select line for Watchdog1 clock [warm reset insensitive]

Figure 6-162. PRCM_CM_DPLL_CLKSEL_WDT1_CLK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							CLKSEL
R-0-0h							R/W-1h

Table 6-175. PRCM_CM_DPLL_CLKSEL_WDT1_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	
0	CLKSEL	R/W	1h	Selects the Mux select line for WDT1 clock [warm reset insensitive] 0h (R/W) = Select 32KHZ clock from RC Oscillator 1h (R/W) = Select 32KHZ from 32K Clock divider

6.13.3.13 PRCM_CM_DPLL_CLKSEL_SYNCTIMER_CLK Register (Offset = 30h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_CM_DPLL_CLKSEL_SYNCTIMER_CLK is shown in Figure 6-163 and described in Table 6-176.

Return to Summary Table.

Selects the Mux select line for SYNCTIMER clock [warm reset insensitive]

Figure 6-163. PRCM_CM_DPLL_CLKSEL_SYNCTIMER_CLK Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RSVD</td></tr>
<tr><td colspan="16">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RSVD</td><td colspan="2">CLKSEL</td></tr>
<tr><td colspan="14">R-0-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 6-176. PRCM_CM_DPLL_CLKSEL_SYNCTIMER_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RSVD	R-0	0h	
1-0	CLKSEL	R/W	0h	Selects the Mux select line for SYNCTIMER clock 0h (R/W) = Select RTC 32K clock 1h (R/W) = Reserved 2h (R/W) = Select PER PLL 32KHz clock 3h (R/W) = Reserved

6.13.3.14 PRCM_CM_DPLL_CLKSEL_MAC_CLK Register (Offset = 34h) [reset = 4h]

Register mask: FFFFFFFFh

PRCM_CM_DPLL_CLKSEL_MAC_CLK is shown in Figure 6-164 and described in Table 6-177.

Return to Summary Table.

Selects the clock divide ration for MII clock [warm reset insensitive]

Figure 6-164. PRCM_CM_DPLL_CLKSEL_MAC_CLK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					MII_CLK_SEL	RESERVED	
R-0-0h					R/W-1h	R-0-0h	

Table 6-177. PRCM_CM_DPLL_CLKSEL_MAC_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	
2	MII_CLK_SEL	R/W	1h	MII Clock Divider Selection. This bit is warm reset insensitive when CPSW RESET_ISO is enabled 0h (R/W) = Selects 1/2 divider of SYSCLK2 1h (R/W) = Selects 1/5 divide ratio of SYSCLK2
1-0	RESERVED	R-0	0h	

6.13.3.15 PRCM_CM_DPLL_CLKSEL_CPTS_RFT_CLK Register (Offset = 38h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_CM_DPLL_CLKSEL_CPTS_RFT_CLK is shown in Figure 6-165 and described in Table 6-178.

Return to Summary Table.

Selects the Mux select line for CPTS RFT clock [warm reset insensitive]

Figure 6-165. PRCM_CM_DPLL_CLKSEL_CPTS_RFT_CLK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														CLKSEL	
R-0-0h														R/W-0h	

Table 6-178. PRCM_CM_DPLL_CLKSEL_CPTS_RFT_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	
1-0	CLKSEL	R/W	0h	Selects the Mux select line for cpgmac rft clock [warm reset insensitive] 0h (R/W) = Select HSDIVIDER_CORE M4 output 1h (R/W) = Select HSDIVIDE_CORE M5 Output 2h (R/W) = Selects DISP PLL clock as CPTS RFT Clock

6.13.3.16 PRCM_CM_DPLL_CLKSEL_GFX_FCLK Register (Offset = 3Ch) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_CM_DPLL_CLKSEL_GFX_FCLK is shown in Figure 6-166 and described in Table 6-179.

Return to Summary Table.

Selects the divider value for GFX clock [warm reset insensitive]

Figure 6-166. PRCM_CM_DPLL_CLKSEL_GFX_FCLK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						CLKSEL_GFX_FCLK	CLKDIV_SEL_GFX_FCLK
R-0-0h						R/W-0h	R/W-0h

Table 6-179. PRCM_CM_DPLL_CLKSEL_GFX_FCLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	
1	CLKSEL_GFX_FCLK	R/W	0h	Selects the clock on gfx fclk [warm reset insensitive] 0h (R/W) = SGX FCLK is from CORE PLL (same as L3 clock) 1h (R/W) = SGX FCLK is from PER PLL (192 MHz clock)
0	CLKDIV_SEL_GFX_FCLK	R/W	0h	Selects the divider value on gfx fclk [warm reset insensitive] 0h (R/W) = SGX FCLK is same as L3 Clock or 192MHz Clock 1h (R/W) = SGX FCLK is L3 clock/2 or 192Mhz/2

6.13.3.17 PRCM_CM_DPLL_CLKSEL_GPIO0_DBCLK Register (Offset = 40h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_CM_DPLL_CLKSEL_GPIO0_DBCLK is shown in Figure 6-167 and described in Table 6-180.

Return to Summary Table.

Selects the Mux select line for GPIO0 debounce clock [warm reset insensitive]

Figure 6-167. PRCM_CM_DPLL_CLKSEL_GPIO0_DBCLK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
R-0-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RSVD													CLKSEL		
R-0-0h													R/W-0h		

Table 6-180. PRCM_CM_DPLL_CLKSEL_GPIO0_DBCLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RSVD	R-0	0h	
2-0	CLKSEL	R/W	0h	Selects the Mux select line for GPIO0 debounce clock [warm reset insensitive] 0h (R/W) = Select 32KHZ clock from RC Oscillator 1h (R/W) = Select 32KHZ from 32K Crystal Oscillator 2h (R/W) = Select 32KHz from PER_PLL (Source: 192MHz) Clock Divider 3h (R/W) = Select 32KHz from SYS_CLK (Source: main XTAL oscillator clock) Clock Divider 4h (R/W) = Reserved

6.13.3.18 PRCM_CM_CLKSEL_PRU_ICSS_OCP_CLK Register (Offset = 48h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_CM_CLKSEL_PRU_ICSS_OCP_CLK is shown in Figure 6-168 and described in Table 6-181.

Return to Summary Table.

Controls Mux Select of PRU-ICSS OCP clock mux

Figure 6-168. PRCM_CM_CLKSEL_PRU_ICSS_OCP_CLK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							CLKSEL
R-0-0h							R/W-0h

Table 6-181. PRCM_CM_CLKSEL_PRU_ICSS_OCP_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	
0	CLKSEL	R/W	0h	Controls Mux Select of PRU-ICSS OCP clock mux 0h (R/W) = Select L3F clock as OCP Clock of PRU-ICSS 1h (R/W) = Select DISP DPLL clock as OCP clock of PRU-ICSS

6.13.3.19 PRCM_CM_CLKSEL_ADC1_CLK Register (Offset = 4Ch) [reset = 1h]

Register mask: FFFFFFFFh

PRCM_CM_CLKSEL_ADC1_CLK is shown in Figure 6-169 and described in Table 6-182.

Return to Summary Table.

Selects the Mux select line for ADC1 clock [warm reset insensitive]

Figure 6-169. PRCM_CM_CLKSEL_ADC1_CLK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							CLKSEL
R-0-0h							R/W-1h

Table 6-182. PRCM_CM_CLKSEL_ADC1_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	
0	CLKSEL	R/W	1h	Selects the Mux select line for ADC1 clock [warm reset insensitive] 0h (R/W) = Select Main Crystal clock as ADC1 clock 1h (R/W) = Select 192MHz PER PLL clock output as ADC1 clock

6.13.3.20 PRCM_CM_DPLL_CLKSEL_DLL_AGING_CLK Register (Offset = 50h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_CM_DPLL_CLKSEL_DLL_AGING_CLK is shown in Figure 6-170 and described in Table 6-183.

Return to Summary Table.

Selects the clock divider for DLL_AGING module clock [warm reset insensitive]

Figure 6-170. PRCM_CM_DPLL_CLKSEL_DLL_AGING_CLK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														CLKSEL	
R-0-0h														R/W-0h	

Table 6-183. PRCM_CM_DPLL_CLKSEL_DLL_AGING_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	
1-0	CLKSEL	R/W	0h	Selects the divider value for generating the DLL_AGING clock 0h (R/W) = Divide by 8 1h (R/W) = Divide by 16 2h (R/W) = Divide by 32 3h (R) = Reserved

6.13.3.21 PRCM_CM_DPLL_CLKSEL_USBPHY32KHZ_GCLK Register (Offset = 60h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_CM_DPLL_CLKSEL_USBPHY32KHZ_GCLK is shown in Figure 6-171 and described in Table 6-184.

Return to Summary Table.

Selects the Mux select line for USBPHY 32KHZ clock [warm reset insensitive]

Figure 6-171. PRCM_CM_DPLL_CLKSEL_USBPHY32KHZ_GCLK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							CLKSEL
R-0-0h							R/W-0h

Table 6-184. PRCM_CM_DPLL_CLKSEL_USBPHY32KHZ_GCLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	
0	CLKSEL	R/W	0h	Selects the Mux select line for USBPHY 32KHZ clock 0h (R/W) = Select RTC 32K clock

6.13.4 PRCM_CM_GFX Registers

Table 6-185 lists the memory-mapped registers for the PRCM_CM_GFX. All register offset addresses not listed in Table 6-185 should be considered as reserved locations and the register contents should not be modified.

Table 6-185. PRCM_CM_GFX REGISTERS

Offset	Acronym	Register Name	Section
0h	PRCM_CM_GFX_L3_CLKSTCTRL		Section 6.13.4.1
20h	PRCM_CM_GFX_CLKCTRL		Section 6.13.4.2

6.13.4.1 PRCM_CM_GFX_L3_CLKSTCTRL Register (offset = 0h) [reset = 2h]

Register mask: FFFFFFFFh

PRCM_CM_GFX_L3_CLKSTCTRL is shown in Figure 6-172 and described in Table 6-186.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

Figure 6-172. PRCM_CM_GFX_L3_CLKSTCTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">RESERVED</td></tr>
<tr><td colspan="6">Rreturns0s-0h</td><td colspan="2">Rreturns0s-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">Rreturns0s-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="6">RESERVED</td><td>CLKACTIVITY_GFX_FCLK</td><td>CLKACTIVITY_GFX_L3_GCLK</td></tr>
<tr><td colspan="6">Rreturns0s-0h</td><td>R-0h</td><td>R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">CLKTRCTRL</td></tr>
<tr><td colspan="6">Rreturns0s-0h</td><td colspan="2">R/W-2h</td></tr>
</table>

Table 6-186. PRCM_CM_GFX_L3_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	Rreturns0s	0h	
25-10	RESERVED	Rreturns0s	0h	
9	CLKACTIVITY_GFX_FCLK	R	0h	This field indicates the state of the GFX_GCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
8	CLKACTIVITY_GFX_L3_GCLK	R	0h	This field indicates the state of the GFX_L3_GCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
7-2	RESERVED	Rreturns0s	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the GFX clock domain in GFX power domain. 0h (R/W) = NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1h (R/W) = SW_SLEEP: Start a software forced sleep transition on the domain. 2h (R/W) = SW_WKUP: Start a software forced wake-up transition on the domain. 3h (R/W) = Reserved

6.13.4.2 PRCM_CM_GFX_CLKCTRL Register (offset = 20h) [reset = 70000h]

Register mask: FFFFFFFFh

PRCM_CM_GFX_CLKCTRL is shown in Figure 6-173 and described in Table 6-187.

This register manages the GFX clocks.

Figure 6-173. PRCM_CM_GFX_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED					STBYST	IDLEST	
Rreturns0s-0h					R-1h	R-3h	

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
Rreturns0s-0h						R/W-0h	

Table 6-187. PRCM_CM_GFX_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	Rreturns0s	0h	
18	STBYST	R	1h	Module standby status. 0h (R) = Module is functional (not in standby) 1h (R) = Module is in standby
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	Rreturns0s	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.5 PRCM_CM_MPU Registers

Table 6-188 lists the memory-mapped registers for the PRCM_CM_MPU. All register offset addresses not listed in Table 6-188 should be considered as reserved locations and the register contents should not be modified.

Table 6-188. PRCM_CM_MPU REGISTERS

Offset	Acronym	Register Name	Section
0h	PRCM_CM_MPU_CLKSTCTRL		Section 6.13.5.1
20h	PRCM_CM_MPU_CLKCTRL		Section 6.13.5.2

6.13.5.1 PRCM_CM_MPU_CLKSTCTRL Register (offset = 0h) [reset = 102h]

Register mask: FFFFFFFFh

PRCM_CM_MPU_CLKSTCTRL is shown in Figure 6-174 and described in Table 6-189.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

Figure 6-174. PRCM_CM_MPU_CLKSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							CLKACTIVITY_ MPU_CLK
R-0h							R-1h

7	6	5	4	3	2	1	0
RESERVED						CLKTRCTRL	
Rreturns0s-0h						R/W-2h	

Table 6-189. PRCM_CM_MPU_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	CLKACTIVITY_MPU_CLK	R	1h	This field indicates the state of the MPU Clock 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
7-2	RESERVED	Rreturns0s	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the MPU clock domains. 0h (R/W) = NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1h (R/W) = SW_SLEEP: Start a software forced sleep transition on the domain. 2h (R/W) = SW_WKUP: Start a software forced wake-up transition on the domain. 3h (R/W) = HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.

6.13.5.2 PRCM_CM_MPU_CLKCTRL Register (offset = 20h) [reset = 40002h]

Register mask: FFFFFFFFh

PRCM_CM_MPU_CLKCTRL is shown in Figure 6-175 and described in Table 6-190.

This register manages the MPU clocks.

Figure 6-175. PRCM_CM_MPU_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED					STBYST	IDLEST	
Rreturns0s-0h					R-1h	R-0h	

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
Rreturns0s-0h						R/W-2h	

Table 6-190. PRCM_CM_MPU_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	Rreturns0s	0h	
18	STBYST	R	1h	Module standby status. 0h (R) = Module is functional (not in standby) 1h (R) = Module is in standby
17-16	IDLEST	R	0h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	Rreturns0s	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R/W) = Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. Module clocks may be gated according to the clock domain state. 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6 CM_PER Registers

Table 6-191 lists the memory-mapped registers for the CM_PER. All register offset addresses not listed in Table 6-191 should be considered as reserved locations and the register contents should not be modified.

Table 6-191. CM_PER Registers

Offset	Acronym	Register Name	Section
0h	PRCM_CM_PER_L3_CLKSTCTRL		Section 6.13.6.1
20h	PRCM_CM_PER_L3_CLKCTRL		Section 6.13.6.2
40h	PRCM_CM_PER_L3_INSTR_CLKCTRL		Section 6.13.6.3
50h	PRCM_CM_PER_OCMCRAM_CLKCTRL		Section 6.13.6.4
68h	PRCM_CM_PER_VPFE0_CLKCTRL		Section 6.13.6.5
70h	PRCM_CM_PER_VPFE1_CLKCTRL		Section 6.13.6.6
78h	PRCM_CM_PER_TPCC_CLKCTRL		Section 6.13.6.7
80h	PRCM_CM_PER_TPTC0_CLKCTRL		Section 6.13.6.8
88h	PRCM_CM_PER_TPTC1_CLKCTRL		Section 6.13.6.9
90h	PRCM_CM_PER_TPTC2_CLKCTRL		Section 6.13.6.10
98h	PRCM_CM_PER_DLL_AGING_CLKCTRL		Section 6.13.6.11
A0h	PRCM_CM_PER_L4HS_CLKCTRL		Section 6.13.6.12
200h	PRCM_CM_PER_L3S_CLKSTCTRL		Section 6.13.6.13
220h	PRCM_CM_PER_GPMC_CLKCTRL		Section 6.13.6.14
230h	PRCM_CM_PER_ADC1_CLKCTRL		Section 6.13.6.15
238h	PRCM_CM_PER_MCASP0_CLKCTRL		Section 6.13.6.16
240h	PRCM_CM_PER_MCASP1_CLKCTRL		Section 6.13.6.17
248h	PRCM_CM_PER_MMC2_CLKCTRL		Section 6.13.6.18
258h	PRCM_CM_PER_QSPI_CLKCTRL		Section 6.13.6.19
260h	PRCM_CM_PER_USB_OTG_SS0_CLKCTRL		Section 6.13.6.20
268h	PRCM_CM_PER_USB_OTG_SS1_CLKCTRL		Section 6.13.6.21
300h	PRCM_CM_PER_PRU_ICSS_CLKSTCTRL		Section 6.13.6.22
320h	PRCM_CM_PER_PRU_ICSS_CLKCTRL		Section 6.13.6.23
400h	PRCM_CM_PER_L4LS_CLKSTCTRL		Section 6.13.6.24
420h	PRCM_CM_PER_L4LS_CLKCTRL		Section 6.13.6.25
428h	PRCM_CM_PER_DCAN0_CLKCTRL		Section 6.13.6.26
430h	PRCM_CM_PER_DCAN1_CLKCTRL		Section 6.13.6.27
438h	PRCM_CM_PER_PWMSS0_CLKCTRL		Section 6.13.6.28
440h	PRCM_CM_PER_PWMSS1_CLKCTRL		Section 6.13.6.29
448h	PRCM_CM_PER_PWMSS2_CLKCTRL		Section 6.13.6.30
450h	PRCM_CM_PER_PWMSS3_CLKCTRL		Section 6.13.6.31
458h	PRCM_CM_PER_PWMSS4_CLKCTRL		Section 6.13.6.32
460h	PRCM_CM_PER_PWMSS5_CLKCTRL		Section 6.13.6.33
468h	PRCM_CM_PER_ELM_CLKCTRL		Section 6.13.6.34
478h	PRCM_CM_PER_GPIO1_CLKCTRL		Section 6.13.6.35
480h	PRCM_CM_PER_GPIO2_CLKCTRL		Section 6.13.6.36
488h	PRCM_CM_PER_GPIO3_CLKCTRL		Section 6.13.6.37
490h	PRCM_CM_PER_GPIO4_CLKCTRL		Section 6.13.6.38
498h	PRCM_CM_PER_GPIO5_CLKCTRL		Section 6.13.6.39
4A0h	PRCM_CM_PER_HDQ1W_CLKCTRL		Section 6.13.6.40
4A8h	PRCM_CM_PER_I2C1_CLKCTRL		Section 6.13.6.41
4B0h	PRCM_CM_PER_I2C2_CLKCTRL		Section 6.13.6.42
4B8h	PRCM_CM_PER_MAILBOX0_CLKCTRL		Section 6.13.6.43
4C0h	PRCM_CM_PER_MMC0_CLKCTRL		Section 6.13.6.44
4C8h	PRCM_CM_PER_MMC1_CLKCTRL		Section 6.13.6.45
500h	PRCM_CM_PER_SPI0_CLKCTRL		Section 6.13.6.46

Table 6-191. CM_PER Registers (continued)

Offset	Acronym	Register Name	Section
508h	PRCM_CM_PER_SPI1_CLKCTRL		Section 6.13.6.47
510h	PRCM_CM_PER_SPI2_CLKCTRL		Section 6.13.6.48
518h	PRCM_CM_PER_SPI3_CLKCTRL		Section 6.13.6.49
520h	PRCM_CM_PER_SPI4_CLKCTRL		Section 6.13.6.50
528h	PRCM_CM_PER_SPINLOCK_CLKCTRL		Section 6.13.6.51
530h	PRCM_CM_PER_TIMER2_CLKCTRL		Section 6.13.6.52
538h	PRCM_CM_PER_TIMER3_CLKCTRL		Section 6.13.6.53
540h	PRCM_CM_PER_TIMER4_CLKCTRL		Section 6.13.6.54
548h	PRCM_CM_PER_TIMER5_CLKCTRL		Section 6.13.6.55
550h	PRCM_CM_PER_TIMER6_CLKCTRL		Section 6.13.6.56
558h	PRCM_CM_PER_TIMER7_CLKCTRL		Section 6.13.6.57
560h	PRCM_CM_PER_TIMER8_CLKCTRL		Section 6.13.6.58
568h	PRCM_CM_PER_TIMER9_CLKCTRL		Section 6.13.6.59
570h	PRCM_CM_PER_TIMER10_CLKCTRL		Section 6.13.6.60
578h	PRCM_CM_PER_TIMER11_CLKCTRL		Section 6.13.6.61
580h	PRCM_CM_PER_UART1_CLKCTRL		Section 6.13.6.62
588h	PRCM_CM_PER_UART2_CLKCTRL		Section 6.13.6.63
590h	PRCM_CM_PER_UART3_CLKCTRL		Section 6.13.6.64
598h	PRCM_CM_PER_UART4_CLKCTRL		Section 6.13.6.65
5A0h	PRCM_CM_PER_UART5_CLKCTRL		Section 6.13.6.66
5B8h	PRCM_CM_PER_USBPHYOCP2SCP0_CLKCTRL		Section 6.13.6.67
5C0h	PRCM_CM_PER_USBPHYOCP2SCP1_CLKCTRL		Section 6.13.6.68
700h	PRCM_CM_PER_EMIF_CLKSTCTRL		Section 6.13.6.69
720h	PRCM_CM_PER_EMIF_CLKCTRL		Section 6.13.6.70
728h	PRCM_CM_PER_DLL_CLKCTRL		Section 6.13.6.71
800h	PRCM_CM_PER_LCDC_CLKSTCTRL		Section 6.13.6.72
A00h	PRCM_CM_PER_DSS_CLKSTCTRL		Section 6.13.6.73
A20h	PRCM_CM_PER_DSS_CLKCTRL		Section 6.13.6.74
B00h	PRCM_CM_PER_CPSW_CLKSTCTRL		Section 6.13.6.75
B20h	PRCM_CM_PER_CPGMAC0_CLKCTRL		Section 6.13.6.76
C00h	PRCM_CM_PER_OCPWP_L3_CLKSTCTRL		Section 6.13.6.77
C20h	PRCM_CM_PER_OCPWP_CLKCTRL		Section 6.13.6.78

6.13.6.1 PRCM_CM_PER_L3_CLKSTCTRL Register (Offset = 0h) [reset = 402h]

Register mask: FFFFFFFFh

PRCM_CM_PER_L3_CLKSTCTRL is shown in Figure 6-176 and described in Table 6-192.

Return to Summary Table.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

Figure 6-176. PRCM_CM_PER_L3_CLKSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							

15	14	13	12	11	10	9	8
RESERVED					CLKACTIVITY_L3_GCLK	RESERVED	CLKACTIVITY_DLL_AGING_GCLK
R-0-0h					R-1h	R-0-0h	R-0h

7	6	5	4	3	2	1	0
RESERVED						CLKTRCTRL	
R-0-0h						R/W-2h	

Table 6-192. PRCM_CM_PER_L3_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	
10	CLKACTIVITY_L3_GCLK	R	1h	This field indicates the state of the L3_GCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
9	RESERVED	R-0	0h	
8	CLKACTIVITY_DLL_AGING_GCLK	R	0h	This field indicates the state of the DLL_AGING_GCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
7-2	RESERVED	R-0	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the L3 clock domain. 0h (R/W) = NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1h (R/W) = SW_SLEEP: Start a software forced sleep transition on the domain. 2h (R/W) = SW_WKUP: Start a software forced wake-up transition on the domain. 3h (R/W) = Reserved

6.13.6.2 PRCM_CM_PER_L3_CLKCTRL Register (Offset = 20h) [reset = 2h]

Register mask: FFFFFFFFh

PRCM_CM_PER_L3_CLKCTRL is shown in Figure 6-177 and described in Table 6-193.

Return to Summary Table.

This register manages the L3 Interconnect clocks.

Figure 6-177. PRCM_CM_PER_L3_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED						IDLEST	
R-0-0h						R-0h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						MODULEMODE	
R-0-0h						R/W-2h	

Table 6-193. PRCM_CM_PER_L3_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	0h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.3 PRCM_CM_PER_L3_INSTR_CLKCTRL Register (Offset = 40h) [reset = 2h]

Register mask: FFFFFFFFh

PRCM_CM_PER_L3_INSTR_CLKCTRL is shown in Figure 6-178 and described in Table 6-194.

Return to Summary Table.

This register manages the L3 INSTR clocks.

Figure 6-178. PRCM_CM_PER_L3_INSTR_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED						IDLEST	
R-0-0h						R-0h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						MODULEMODE	
R-0-0h						R/W-2h	

Table 6-194. PRCM_CM_PER_L3_INSTR_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	0h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.4 PRCM_CM_PER_OCMCRAM_CLKCTRL Register (Offset = 50h) [reset = 00030002h]

Register mask: FFFFFFFFh

PRCM_CM_PER_OCMCRAM_CLKCTRL is shown in Figure 6-179 and described in Table 6-195.

Return to Summary Table.

This register manages the OCMC clocks.

Figure 6-179. PRCM_CM_PER_OCMCRAM_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-2h	

Table 6-195. PRCM_CM_PER_OCMCRAM_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.5 PRCM_CM_PER_VPFE0_CLKCTRL Register (Offset = 68h) [reset = 00070000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_VPFE0_CLKCTRL is shown in Figure 6-180 and described in Table 6-196.

Return to Summary Table.

This register manages the VPFE0 clocks.

Figure 6-180. PRCM_CM_PER_VPFE0_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED					STBYST	IDLEST	
R-0-0h					R-1h	R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-196. PRCM_CM_PER_VPFE0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	STBYST	R	1h	Module standby status. 0h (R) = Module is functional (not in standby) 1h (R) = Module is in standby
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.6 PRCM_CM_PER_VPFE1_CLKCTRL Register (Offset = 70h) [reset = 00070000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_VPFE1_CLKCTRL is shown in Figure 6-181 and described in Table 6-197.

Return to Summary Table.

This register manages the VPFE1 clocks.

Figure 6-181. PRCM_CM_PER_VPFE1_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED					STBYST	IDLEST	
R-0-0h					R-1h	R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-197. PRCM_CM_PER_VPFE1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	STBYST	R	1h	Module standby status. 0h (R) = Module is functional (not in standby) 1h (R) = Module is in standby
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.7 PRCM_CM_PER_TPCC_CLKCTRL Register (Offset = 78h) [reset = 00030000h]

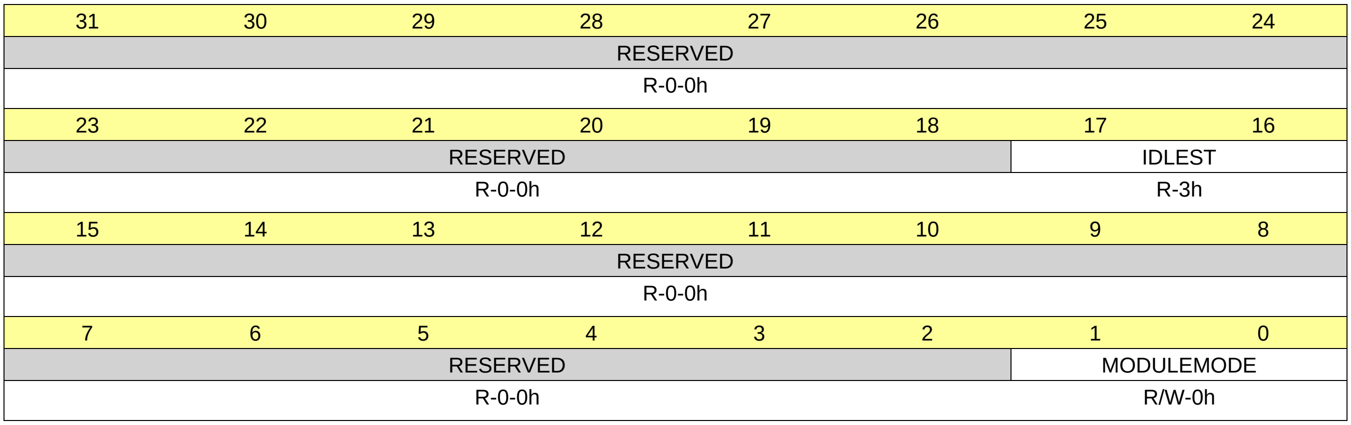
Register mask: FFFFFFFFh

PRCM_CM_PER_TPCC_CLKCTRL is shown in Figure 6-182 and described in Table 6-198.

Return to Summary Table.

This register manages the TPCC clocks.

Figure 6-182. PRCM_CM_PER_TPCC_CLKCTRL Register



31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED						IDLEST	
R-0-0h						R-3h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-198. PRCM_CM_PER_TPCC_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.8 PRCM_CM_PER_TPTC0_CLKCTRL Register (Offset = 80h) [reset = 00070000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_TPTC0_CLKCTRL is shown in Figure 6-183 and described in Table 6-199.

Return to Summary Table.

This register manages the TPTC clocks.

Figure 6-183. PRCM_CM_PER_TPTC0_CLKCTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="5">RESERVED</td><td>STBYST</td><td colspan="2">IDLEST</td></tr>
<tr><td colspan="5">R-0-0h</td><td>R-1h</td><td colspan="2">R-3h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">MODULEMODE</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 6-199. PRCM_CM_PER_TPTC0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	STBYST	R	1h	Module standby status. 0h (R) = Module is functional (not in standby) 1h (R) = Module is in standby
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.9 PRCM_CM_PER_TPTC1_CLKCTRL Register (Offset = 88h) [reset = 00070000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_TPTC1_CLKCTRL is shown in Figure 6-184 and described in Table 6-200.

Return to Summary Table.

This register manages the TPTC1 clocks.

Figure 6-184. PRCM_CM_PER_TPTC1_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED					STBYST	IDLEST	
R-0-0h					R-1h	R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-200. PRCM_CM_PER_TPTC1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	STBYST	R	1h	Module standby status. 0h (R) = Module is functional (not in standby) 1h (R) = Module is in standby
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.10 PRCM_CM_PER_TPTC2_CLKCTRL Register (Offset = 90h) [reset = 00070000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_TPTC2_CLKCTRL is shown in Figure 6-185 and described in Table 6-201.

Return to Summary Table.

This register manages the TPTC2 clocks.

Figure 6-185. PRCM_CM_PER_TPTC2_CLKCTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="5">RESERVED</td><td>STBYST</td><td colspan="2">IDLEST</td></tr>
<tr><td colspan="5">R-0-0h</td><td>R-1h</td><td colspan="2">R-3h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">MODULEMODE</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 6-201. PRCM_CM_PER_TPTC2_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	STBYST	R	1h	Module standby status. 0h (R) = Module is functional (not in standby) 1h (R) = Module is in standby
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.11 PRCM_CM_PER_DLL_AGING_CLKCTRL Register (Offset = 98h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_DLL_AGING_CLKCTRL is shown in Figure 6-186 and described in Table 6-202.

Return to Summary Table.

This register manages the DLL_AGING clocks.

Figure 6-186. PRCM_CM_PER_DLL_AGING_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-202. PRCM_CM_PER_DLL_AGING_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.12 PRCM_CM_PER_L4HS_CLKCTRL Register (Offset = A0h) [reset = 2h]

Register mask: FFFFFFFFh

PRCM_CM_PER_L4HS_CLKCTRL is shown in Figure 6-187 and described in Table 6-203.

Return to Summary Table.

This register manages the L4 Fast clocks.

Figure 6-187. PRCM_CM_PER_L4HS_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-0h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-2h	

Table 6-203. PRCM_CM_PER_L4HS_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	0h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.13 PRCM_CM_PER_L3S_CLKSTCTRL Register (Offset = 200h) [reset = 302h]

Register mask: FFFFFFFFh

PRCM_CM_PER_L3S_CLKSTCTRL is shown in Figure 6-188 and described in Table 6-204.

Return to Summary Table.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

Figure 6-188. PRCM_CM_PER_L3S_CLKSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED			CLKACTIVITY_USB_OTG_SS_REFCLK	CLKACTIVITY_MMC_FCLK	CLKACTIVITY_MCASP_GCLK	CLKACTIVITY_ADC1_FGCLK	CLKACTIVITY_L3S_GCLK
R-0-0h			R-0h	R-0h	R-0h	R-1h	R-1h
7	6	5	4	3	2	1	0
RESERVED						CLKTRCTRL	
R-0-0h						R/W-2h	

Table 6-204. PRCM_CM_PER_L3S_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R-0	0h	
12	CLKACTIVITY_USB_OTG_SS_REFCLK	R	0h	This field indicates the state of the USB_OTG_SS_REFCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
11	CLKACTIVITY_MMC_FCLK	R	0h	This field indicates the state of the MMC_GCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
10	CLKACTIVITY_MCASP_GCLK	R	0h	This field indicates the state of the MCASP_GCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
9	CLKACTIVITY_ADC1_FGCLK	R	1h	This register manages the ADC1 clocks. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
8	CLKACTIVITY_L3S_GCLK	R	1h	This field indicates the state of the L3S_GCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
7-2	RESERVED	R-0	0h	

Table 6-204. PRCM_CM_PER_L3S_CLKSTCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the L3 Slow clock domain. 0h (R/W) = NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1h (R/W) = SW_SLEEP: Start a software forced sleep transition on the domain. 2h (R/W) = SW_WKUP: Start a software forced wake-up transition on the domain. 3h (R/W) = Reserved

6.13.6.14 PRCM_CM_PER_GPMC_CLKCTRL Register (Offset = 220h) [reset = 00030002h]

Register mask: FFFFFFFFh

PRCM_CM_PER_GPMC_CLKCTRL is shown in Figure 6-189 and described in Table 6-205.

Return to Summary Table.

This register manages the GPMC clocks.

Figure 6-189. PRCM_CM_PER_GPMC_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED						IDLEST	
R-0-0h						R-3h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						MODULEMODE	
R-0-0h						R/W-2h	

Table 6-205. PRCM_CM_PER_GPMC_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.15 PRCM_CM_PER_ADC1_CLKCTRL Register (Offset = 230h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_ADC1_CLKCTRL is shown in Figure 6-190 and described in Table 6-206.

Return to Summary Table.

This register manages the ADC1 clocks.

Figure 6-190. PRCM_CM_PER_ADC1_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-206. PRCM_CM_PER_ADC1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.16 PRCM_CM_PER_MCASP0_CLKCTRL Register (Offset = 238h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_MCASP0_CLKCTRL is shown in Figure 6-191 and described in Table 6-207.

Return to Summary Table.

This register manages the MCASP0 clocks.

Figure 6-191. PRCM_CM_PER_MCASP0_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-207. PRCM_CM_PER_MCASP0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.17 PRCM_CM_PER_MCASP1_CLKCTRL Register (Offset = 240h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_MCASP1_CLKCTRL is shown in Figure 6-192 and described in Table 6-208.

Return to Summary Table.

This register manages the MCASP1 clocks.

Figure 6-192. PRCM_CM_PER_MCASP1_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-208. PRCM_CM_PER_MCASP1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.18 PRCM_CM_PER_MMC2_CLKCTRL Register (Offset = 248h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_MMC2_CLKCTRL is shown in Figure 6-193 and described in Table 6-209.

Return to Summary Table.

This register manages the MMC2 clocks.

Figure 6-193. PRCM_CM_PER_MMC2_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED					RESERVED	IDLEST	
R-0-0h					R-0-0h	R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-209. PRCM_CM_PER_MMC2_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.19 PRCM_CM_PER_QSPI_CLKCTRL Register (Offset = 258h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_QSPI_CLKCTRL is shown in Figure 6-194 and described in Table 6-210.

Return to Summary Table.

This register manages the QSPI clocks.

Figure 6-194. PRCM_CM_PER_QSPI_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED					RESERVED	IDLEST	
R-0-0h					R-0-0h	R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-210. PRCM_CM_PER_QSPI_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.20 PRCM_CM_PER_USB_OTG_SS0_CLKCTRL Register (Offset = 260h) [reset = 00070000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_USB_OTG_SS0_CLKCTRL is shown in Figure 6-195 and described in Table 6-211.

Return to Summary Table.

This register manages the USB_OTG_SS0 clocks.

Figure 6-195. PRCM_CM_PER_USB_OTG_SS0_CLKCTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="5">RESERVED</td><td>STBYST</td><td colspan="2">IDLEST</td></tr>
<tr><td colspan="5">R-0-0h</td><td>R-1h</td><td colspan="2">R-3h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>OPTFCLKEN_ REFCLK960M</td></tr>
<tr><td colspan="7">R-0-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">MODULEMODE</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 6-211. PRCM_CM_PER_USB_OTG_SS0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	STBYST	R	1h	Module standby status. [warm reset insensitive] 0h (R) = Module is functional (not in standby) 1h (R) = Module is in standby
17-16	IDLEST	R	3h	Module idle status. [warm reset insensitive] 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-9	RESERVED	R-0	0h	
8	OPTFCLKEN_REFCLK960M	R/W	0h	USB_OTG optional clock control: REFCLK960(960MHz) 0h (R/W) = Optional functional clock is disabled 1h (R/W) = Optional functional clock is enabled
7-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.21 PRCM_CM_PER_USB_OTG_SS1_CLKCTRL Register (Offset = 268h) [reset = 00070000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_USB_OTG_SS1_CLKCTRL is shown in Figure 6-196 and described in Table 6-212.

Return to Summary Table.

This register manages the USB_OTG_SS1 clocks.

Figure 6-196. PRCM_CM_PER_USB_OTG_SS1_CLKCTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="5">RESERVED</td><td>STBYST</td><td colspan="2">IDLEST</td></tr>
<tr><td colspan="5">R-0-0h</td><td>R-1h</td><td colspan="2">R-3h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>OPTFCLKEN_ REFCLK960M</td></tr>
<tr><td colspan="7">R-0-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">MODULEMODE</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 6-212. PRCM_CM_PER_USB_OTG_SS1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	STBYST	R	1h	Module standby status. [warm reset insensitive] 0h (R) = Module is functional (not in standby) 1h (R) = Module is in standby
17-16	IDLEST	R	3h	Module idle status. [warm reset insensitive] 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-9	RESERVED	R-0	0h	
8	OPTFCLKEN_REFCLK960M	R/W	0h	USB_OTG optional clock control: REFCLK960(960MHz) 0h (R/W) = Optional functional clock is disabled 1h (R/W) = Optional functional clock is enabled
7-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.22 PRCM_CM_PER_PRU_ICSS_CLKSTCTRL Register (Offset = 300h) [reset = 2h]

Register mask: FFFFFFFFh

PRCM_CM_PER_PRU_ICSS_CLKSTCTRL is shown in Figure 6-197 and described in Table 6-213.

Return to Summary Table.

This register enables the clock domain state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

Figure 6-197. PRCM_CM_PER_PRU_ICSS_CLKSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED					CLKACTIVITY_PRU_ICSS_UART_GCLK	CLKACTIVITY_PRU_ICSS_IEP_GCLK	CLKACTIVITY_PRU_ICSS_OCP_GCLK
R-0-0h					R-0h	R-0h	R-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						CLKTRCTRL	
R-0-0h						R/W-2h	

Table 6-213. PRCM_CM_PER_PRU_ICSS_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	
10	CLKACTIVITY_PRU_ICSS_UART_GCLK	R	0h	This field indicates the state of the PRU-ICSS UART clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
9	CLKACTIVITY_PRU_ICSS_IEP_GCLK	R	0h	This field indicates the state of the PRU-ICSS IEP clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
8	CLKACTIVITY_PRU_ICSS_OCP_GCLK	R	0h	This field indicates the state of the PRU-ICSS OCP clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
7-2	RESERVED	R-0	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the PRU-ICSS OCP clock domain. 0h (R/W) = NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1h (R/W) = SW_SLEEP: Start a software forced sleep transition on the domain. 2h (R/W) = SW_WKUP: Start a software forced wake-up transition on the domain. 3h (R/W) = Reserved

6.13.6.23 PRCM_CM_PER_PRU_ICSS_CLKCTRL Register (Offset = 320h) [reset = 00070000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_PRU_ICSS_CLKCTRL is shown in Figure 6-198 and described in Table 6-214.

Return to Summary Table.

This register manages the ICSS clocks.

Figure 6-198. PRCM_CM_PER_PRU_ICSS_CLKCTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="5">RESERVED</td><td>STBYST</td><td colspan="2">IDLEST</td></tr>
<tr><td colspan="5">R-0-0h</td><td>R-1h</td><td colspan="2">R-3h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">MODULEMODE</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 6-214. PRCM_CM_PER_PRU_ICSS_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	STBYST	R	1h	Module standby status. 0h (R) = Module is functional (not in standby) 1h (R) = Module is in standby
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.24 PRCM_CM_PER_L4LS_CLKSTCTRL Register (Offset = 400h) [reset = 00400102h]

Register mask: FFFFFFFFh

PRCM_CM_PER_L4LS_CLKSTCTRL is shown in Figure 6-199 and described in Table 6-215.

Return to Summary Table.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-PER and ON-INPER states. It also hold one status bit per clock input of the domain.

Figure 6-199. PRCM_CM_PER_L4LS_CLKSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED				CLKACTIVITY_I2C_FCLK	CLKACTIVITY_GPIO_5_GDBCLK	CLKACTIVITY_GPIO_4_GDBCLK	CLKACTIVITY_GPIO_3_GDBCLK
R/W-0h				R-0h	R-0h	R-0h	R-0h

23	22	21	20	19	18	17	16
CLKACTIVITY_GPIO_2_GDBCLK	CLKACTIVITY_GPIO_1_GDBCLK	CLKACTIVITY_TIMER11_GCLK	CLKACTIVITY_TIMER10_GCLK	CLKACTIVITY_TIMER9_GCLK	CLKACTIVITY_TIMER8_GCLK	CLKACTIVITY_TIMER7_GCLK	CLKACTIVITY_TIMER6_GCLK
R-0h	R-1h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

15	14	13	12	11	10	9	8
CLKACTIVITY_TIMER5_GCLK	CLKACTIVITY_TIMER4_GCLK	CLKACTIVITY_TIMER3_GCLK	CLKACTIVITY_TIMER2_GCLK	CLKACTIVITY_CAN_CLK	CLKACTIVITY_UART_GFCLK	CLKACTIVITY_SPI_GCLK	CLKACTIVITY_L4LS_GCLK
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-1h

7	6	5	4	3	2	1	0
RESERVED						CLKTRCTRL	
R-0-0h						R/W-2h	

Table 6-215. PRCM_CM_PER_L4LS_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27	CLKACTIVITY_I2C_FCLK	R	0h	This field indicates the state of the I2C _FCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
26	CLKACTIVITY_GPIO_5_GDBCLK	R	0h	This field indicates the state of the GPIO5_GDBCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
25	CLKACTIVITY_GPIO_4_GDBCLK	R	0h	This field indicates the state of the GPIO4_GDBCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
24	CLKACTIVITY_GPIO_3_GDBCLK	R	0h	This field indicates the state of the GPIO3_GDBCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
23	CLKACTIVITY_GPIO_2_GDBCLK	R	0h	This field indicates the state of the GPIO2_ GDBCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
22	CLKACTIVITY_GPIO_1_GDBCLK	R	1h	This field indicates the state of the GPIO1_GDBCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active

Table 6-215. PRCM_CM_PER_L4LS_CLKSTCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	CLKACTIVITY_TIMER11_GCLK	R	0h	This field indicates the state of the TIMER11 CLKTIMER clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
20	CLKACTIVITY_TIMER10_GCLK	R	0h	This field indicates the state of the TIMER10 CLKTIMER clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
19	CLKACTIVITY_TIMER9_GCLK	R	0h	This field indicates the state of the TIMER9 CLKTIMER clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
18	CLKACTIVITY_TIMER8_GCLK	R	0h	This field indicates the state of the TIMER8 CLKTIMER clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
17	CLKACTIVITY_TIMER7_GCLK	R	0h	This field indicates the state of the TIMER7 CLKTIMER clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
16	CLKACTIVITY_TIMER6_GCLK	R	0h	This field indicates the state of the TIMER6 CLKTIMER clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
15	CLKACTIVITY_TIMER5_GCLK	R	0h	This field indicates the state of the TIMER5 CLKTIMER clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
14	CLKACTIVITY_TIMER4_GCLK	R	0h	This field indicates the state of the TIMER4 CLKTIMER clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
13	CLKACTIVITY_TIMER3_GCLK	R	0h	This field indicates the state of the TIMER3 CLKTIMER clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
12	CLKACTIVITY_TIMER2_GCLK	R	0h	This field indicates the state of the TIMER2 CLKTIMER clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
11	CLKACTIVITY_CAN_CLK	R	0h	This field indicates the state of the CAN_CLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
10	CLKACTIVITY_UART_GFCLK	R	0h	This field indicates the state of the UART_GFCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
9	CLKACTIVITY_SPI_GCLK	R	0h	This field indicates the state of the SPI_GCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
8	CLKACTIVITY_L4LS_GCLK	R	1h	This field indicates the state of the L4LS_GCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active

Table 6-215. PRCM_CM_PER_L4LS_CLKSTCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-2	RESERVED	R-0	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the L4 SLOW clock domain in PER power domain. 0h (R/W) = NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1h (R/W) = SW_SLEEP: Start a software forced sleep transition on the domain. 2h (R/W) = SW_WKUP: Start a software forced wake-up transition on the domain. 3h (R/W) = Reserved

6.13.6.25 PRCM_CM_PER_L4LS_CLKCTRL Register (Offset = 420h) [reset = 2h]

Register mask: FFFFFFFFh

PRCM_CM_PER_L4LS_CLKCTRL is shown in Figure 6-200 and described in Table 6-216.

Return to Summary Table.

This register manages the L4LS clocks.

Figure 6-200. PRCM_CM_PER_L4LS_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-0h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-2h	

Table 6-216. PRCM_CM_PER_L4LS_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	0h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.26 PRCM_CM_PER_DCAN0_CLKCTRL Register (Offset = 428h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_DCAN0_CLKCTRL is shown in Figure 6-201 and described in Table 6-217.

Return to Summary Table.

This register manages the DCAN0 clocks.

Figure 6-201. PRCM_CM_PER_DCAN0_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED					RESERVED	IDLEST	
R-0-0h					R-0-0h	R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-217. PRCM_CM_PER_DCAN0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.27 PRCM_CM_PER_DCAN1_CLKCTRL Register (Offset = 430h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_DCAN1_CLKCTRL is shown in Figure 6-202 and described in Table 6-218.

Return to Summary Table.

This register manages the DCAN1 clocks.

Figure 6-202. PRCM_CM_PER_DCAN1_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED					RESERVED	IDLEST	
R-0-0h					R-0-0h	R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-218. PRCM_CM_PER_DCAN1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.28 PRCM_CM_PER_PWMSS0_CLKCTRL Register (Offset = 438h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_PWMSS0_CLKCTRL is shown in Figure 6-203 and described in Table 6-219.

Return to Summary Table.

This register manages the PWMSS0 clocks.

Figure 6-203. PRCM_CM_PER_PWMSS0_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-219. PRCM_CM_PER_PWMSS0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.29 PRCM_CM_PER_PWMSS1_CLKCTRL Register (Offset = 440h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_PWMSS1_CLKCTRL is shown in Figure 6-204 and described in Table 6-220.

Return to Summary Table.

This register manages the PWMSS1 clocks.

Figure 6-204. PRCM_CM_PER_PWMSS1_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							

23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							

7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-220. PRCM_CM_PER_PWMSS1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.30 PRCM_CM_PER_PWMSS2_CLKCTRL Register (Offset = 448h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_PWMSS2_CLKCTRL is shown in Figure 6-205 and described in Table 6-221.

Return to Summary Table.

This register manages the PWMSS2 clocks.

Figure 6-205. PRCM_CM_PER_PWMSS2_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-221. PRCM_CM_PER_PWMSS2_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.31 PRCM_CM_PER_PWMSS3_CLKCTRL Register (Offset = 450h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_PWMSS3_CLKCTRL is shown in Figure 6-206 and described in Table 6-222.

Return to Summary Table.

This register manages the PWMSS3 clocks.

Figure 6-206. PRCM_CM_PER_PWMSS3_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-222. PRCM_CM_PER_PWMSS3_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.32 PRCM_CM_PER_PWMSS4_CLKCTRL Register (Offset = 458h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_PWMSS4_CLKCTRL is shown in Figure 6-207 and described in Table 6-223.

Return to Summary Table.

This register manages the PWMSS4 clocks.

Figure 6-207. PRCM_CM_PER_PWMSS4_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED						IDLEST	
R-0-0h						R-3h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-223. PRCM_CM_PER_PWMSS4_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.33 PRCM_CM_PER_PWMSS5_CLKCTRL Register (Offset = 460h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_PWMSS5_CLKCTRL is shown in Figure 6-208 and described in Table 6-224.

Return to Summary Table.

This register manages the PWMSS5 clocks.

Figure 6-208. PRCM_CM_PER_PWMSS5_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-224. PRCM_CM_PER_PWMSS5_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.34 PRCM_CM_PER_ELM_CLKCTRL Register (Offset = 468h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_ELM_CLKCTRL is shown in Figure 6-209 and described in Table 6-225.

Return to Summary Table.

This register manages the ELM clocks.

Figure 6-209. PRCM_CM_PER_ELM_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-225. PRCM_CM_PER_ELM_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.35 PRCM_CM_PER_GPIO1_CLKCTRL Register (Offset = 478h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_GPIO1_CLKCTRL is shown in Figure 6-210 and described in Table 6-226.

Return to Summary Table.

This register manages the GPIO1 clocks.

Figure 6-210. PRCM_CM_PER_GPIO1_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED						IDLEST	
R-0-0h						R-3h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							OPTFCLKEN_GPIO_1_GDBCLK
R-0h							R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-226. PRCM_CM_PER_GPIO1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-9	RESERVED	R	0h	
8	OPTFCLKEN_GPIO_1_GDBCLK	R/W	0h	Optional functional clock control. 0h (R/W) = Optional functional clock is disabled 1h (R/W) = Optional functional clock is enabled
7-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.36 PRCM_CM_PER_GPIO2_CLKCTRL Register (Offset = 480h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_GPIO2_CLKCTRL is shown in Figure 6-211 and described in Table 6-227.

Return to Summary Table.

This register manages the GPIO2 clocks.

Figure 6-211. PRCM_CM_PER_GPIO2_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED						IDLEST	
R-0-0h						R-3h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							OPTFCLKEN_GPIO_2_GDBCLK
R-0h							R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-227. PRCM_CM_PER_GPIO2_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-9	RESERVED	R	0h	
8	OPTFCLKEN_GPIO_2_GDBCLK	R/W	0h	Optional functional clock control. 0h (R/W) = Optional functional clock is disabled 1h (R/W) = Optional functional clock is enabled
7-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.37 PRCM_CM_PER_GPIO3_CLKCTRL Register (Offset = 488h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_GPIO3_CLKCTRL is shown in Figure 6-212 and described in Table 6-228.

Return to Summary Table.

This register manages the GPIO3 clocks.

Figure 6-212. PRCM_CM_PER_GPIO3_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							OPTFCLKEN_GPIO_3_GDBCLK
R-0h							R/W-0h
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-228. PRCM_CM_PER_GPIO3_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-9	RESERVED	R	0h	
8	OPTFCLKEN_GPIO_3_GDBCLK	R/W	0h	Optional functional clock control. 0h (R/W) = Optional functional clock is disabled 1h (R/W) = Optional functional clock is enabled
7-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.38 PRCM_CM_PER_GPIO4_CLKCTRL Register (Offset = 490h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_GPIO4_CLKCTRL is shown in Figure 6-213 and described in Table 6-229.

Return to Summary Table.

This register manages the GPIO4 clocks.

Figure 6-213. PRCM_CM_PER_GPIO4_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED						IDLEST	
R-0-0h						R-3h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							OPTFCLKEN_GPIO_4_GDBCLK
R-0h							R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-229. PRCM_CM_PER_GPIO4_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-9	RESERVED	R	0h	
8	OPTFCLKEN_GPIO_4_GDBCLK	R/W	0h	Optional functional clock control. 0h (R/W) = Optional functional clock is disabled 1h (R/W) = Optional functional clock is enabled
7-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.39 PRCM_CM_PER_GPIO5_CLKCTRL Register (Offset = 498h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_GPIO5_CLKCTRL is shown in Figure 6-214 and described in Table 6-230.

Return to Summary Table.

This register manages the GPIO5 clocks.

Figure 6-214. PRCM_CM_PER_GPIO5_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED						IDLEST	
R-0-0h						R-3h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							OPTFCLKEN_GPIO_5_GDBCLK
R-0h							R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-230. PRCM_CM_PER_GPIO5_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-9	RESERVED	R	0h	
8	OPTFCLKEN_GPIO_5_GDBCLK	R/W	0h	Optional functional clock control. 0h (R/W) = Optional functional clock is disabled 1h (R/W) = Optional functional clock is enabled
7-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.40 PRCM_CM_PER_HDQ1W_CLKCTRL Register (Offset = 4A0h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_HDQ1W_CLKCTRL is shown in Figure 6-215 and described in Table 6-231.

Return to Summary Table.

This register manages the HDQ1W clocks.

Figure 6-215. PRCM_CM_PER_HDQ1W_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-231. PRCM_CM_PER_HDQ1W_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.41 PRCM_CM_PER_I2C1_CLKCTRL Register (Offset = 4A8h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_I2C1_CLKCTRL is shown in Figure 6-216 and described in Table 6-232.

Return to Summary Table.

This register manages the I2C1 clocks.

Figure 6-216. PRCM_CM_PER_I2C1_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-232. PRCM_CM_PER_I2C1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.42 PRCM_CM_PER_I2C2_CLKCTRL Register (Offset = 4B0h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_I2C2_CLKCTRL is shown in Figure 6-217 and described in Table 6-233.

Return to Summary Table.

This register manages the I2C2 clocks.

Figure 6-217. PRCM_CM_PER_I2C2_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-233. PRCM_CM_PER_I2C2_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.43 PRCM_CM_PER_MAILBOX0_CLKCTRL Register (Offset = 4B8h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_MAILBOX0_CLKCTRL is shown in Figure 6-218 and described in Table 6-234.

Return to Summary Table.

This register manages the MAILBOX0 clocks.

Figure 6-218. PRCM_CM_PER_MAILBOX0_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-234. PRCM_CM_PER_MAILBOX0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.44 PRCM_CM_PER_MMC0_CLKCTRL Register (Offset = 4C0h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_MMC0_CLKCTRL is shown in Figure 6-219 and described in Table 6-235.

Return to Summary Table.

This register manages the MMC0 clocks.

Figure 6-219. PRCM_CM_PER_MMC0_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED					RESERVED	IDLEST	
R-0-0h					R-0-0h	R-3h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-235. PRCM_CM_PER_MMC0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.45 PRCM_CM_PER_MMC1_CLKCTRL Register (Offset = 4C8h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_MMC1_CLKCTRL is shown in Figure 6-220 and described in Table 6-236.

Return to Summary Table.

This register manages the MMC1 clocks.

Figure 6-220. PRCM_CM_PER_MMC1_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED					RESERVED	IDLEST	
R-0-0h					R-0-0h	R-3h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-236. PRCM_CM_PER_MMC1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.46 PRCM_CM_PER_SPI0_CLKCTRL Register (Offset = 500h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_SPI0_CLKCTRL is shown in Figure 6-221 and described in Table 6-237.

Return to Summary Table.

This register manages the SPI0 clocks.

Figure 6-221. PRCM_CM_PER_SPI0_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-237. PRCM_CM_PER_SPI0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.47 PRCM_CM_PER_SPI1_CLKCTRL Register (Offset = 508h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_SPI1_CLKCTRL is shown in Figure 6-222 and described in Table 6-238.

Return to Summary Table.

This register manages the SPI1 clocks.

Figure 6-222. PRCM_CM_PER_SPI1_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-238. PRCM_CM_PER_SPI1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.48 PRCM_CM_PER_SPI2_CLKCTRL Register (Offset = 510h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_SPI2_CLKCTRL is shown in Figure 6-223 and described in Table 6-239.

Return to Summary Table.

This register manages the SPI2 clocks.

Figure 6-223. PRCM_CM_PER_SPI2_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-239. PRCM_CM_PER_SPI2_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.49 PRCM_CM_PER_SPI3_CLKCTRL Register (Offset = 518h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_SPI3_CLKCTRL is shown in Figure 6-224 and described in Table 6-240.

Return to Summary Table.

This register manages the SPI3 clocks.

Figure 6-224. PRCM_CM_PER_SPI3_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-240. PRCM_CM_PER_SPI3_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.50 PRCM_CM_PER_SPI4_CLKCTRL Register (Offset = 520h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_SPI4_CLKCTRL is shown in Figure 6-225 and described in Table 6-241.

Return to Summary Table.

This register manages the SPI4 clocks.

Figure 6-225. PRCM_CM_PER_SPI4_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-241. PRCM_CM_PER_SPI4_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.51 PRCM_CM_PER_SPINLOCK_CLKCTRL Register (Offset = 528h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_SPINLOCK_CLKCTRL is shown in Figure 6-226 and described in Table 6-242.

Return to Summary Table.

This register manages the SPINLOCK clocks.

Figure 6-226. PRCM_CM_PER_SPINLOCK_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-242. PRCM_CM_PER_SPINLOCK_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.52 PRCM_CM_PER_TIMER2_CLKCTRL Register (Offset = 530h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_TIMER2_CLKCTRL is shown in Figure 6-227 and described in Table 6-243.

Return to Summary Table.

This register manages the TIMER2 clocks.

Figure 6-227. PRCM_CM_PER_TIMER2_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED						IDLEST	
R-0-0h						R-3h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-243. PRCM_CM_PER_TIMER2_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.53 PRCM_CM_PER_TIMER3_CLKCTRL Register (Offset = 538h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_TIMER3_CLKCTRL is shown in Figure 6-228 and described in Table 6-244.

Return to Summary Table.

This register manages the TIMER3 clocks.

Figure 6-228. PRCM_CM_PER_TIMER3_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-244. PRCM_CM_PER_TIMER3_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.54 PRCM_CM_PER_TIMER4_CLKCTRL Register (Offset = 540h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_TIMER4_CLKCTRL is shown in Figure 6-229 and described in Table 6-245.

Return to Summary Table.

This register manages the TIMER4 clocks.

Figure 6-229. PRCM_CM_PER_TIMER4_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-245. PRCM_CM_PER_TIMER4_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.55 PRCM_CM_PER_TIMER5_CLKCTRL Register (Offset = 548h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_TIMER5_CLKCTRL is shown in Figure 6-230 and described in Table 6-246.

Return to Summary Table.

This register manages the TIMER5 clocks.

Figure 6-230. PRCM_CM_PER_TIMER5_CLKCTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">IDLEST</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R-3h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">MODULEMODE</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 6-246. PRCM_CM_PER_TIMER5_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.56 PRCM_CM_PER_TIMER6_CLKCTRL Register (Offset = 550h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_TIMER6_CLKCTRL is shown in Figure 6-231 and described in Table 6-247.

Return to Summary Table.

This register manages the TIMER6 clocks.

Figure 6-231. PRCM_CM_PER_TIMER6_CLKCTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">IDLEST</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R-3h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">MODULEMODE</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 6-247. PRCM_CM_PER_TIMER6_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.57 PRCM_CM_PER_TIMER7_CLKCTRL Register (Offset = 558h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_TIMER7_CLKCTRL is shown in Figure 6-232 and described in Table 6-248.

Return to Summary Table.

This register manages the TIMER7 clocks.

Figure 6-232. PRCM_CM_PER_TIMER7_CLKCTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">IDLEST</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R-3h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">MODULEMODE</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 6-248. PRCM_CM_PER_TIMER7_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.58 PRCM_CM_PER_TIMER8_CLKCTRL Register (Offset = 560h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_TIMER8_CLKCTRL is shown in Figure 6-233 and described in Table 6-249.

Return to Summary Table.

This register manages the TIMER8 clocks.

Figure 6-233. PRCM_CM_PER_TIMER8_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-249. PRCM_CM_PER_TIMER8_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.59 PRCM_CM_PER_TIMER9_CLKCTRL Register (Offset = 568h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_TIMER9_CLKCTRL is shown in Figure 6-234 and described in Table 6-250.

Return to Summary Table.

This register manages the TIMER9 clocks.

Figure 6-234. PRCM_CM_PER_TIMER9_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED						IDLEST	
R-0-0h						R-3h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-250. PRCM_CM_PER_TIMER9_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.60 PRCM_CM_PER_TIMER10_CLKCTRL Register (Offset = 570h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_TIMER10_CLKCTRL is shown in Figure 6-235 and described in Table 6-251.

Return to Summary Table.

This register manages the TIMER10 clocks.

Figure 6-235. PRCM_CM_PER_TIMER10_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-251. PRCM_CM_PER_TIMER10_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.61 PRCM_CM_PER_TIMER11_CLKCTRL Register (Offset = 578h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_TIMER11_CLKCTRL is shown in Figure 6-236 and described in Table 6-252.

Return to Summary Table.

This register manages the TIMER11 clocks.

Figure 6-236. PRCM_CM_PER_TIMER11_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-252. PRCM_CM_PER_TIMER11_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.62 PRCM_CM_PER_UART1_CLKCTRL Register (Offset = 580h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_UART1_CLKCTRL is shown in Figure 6-237 and described in Table 6-253.

Return to Summary Table.

This register manages the IART1 clocks.

Figure 6-237. PRCM_CM_PER_UART1_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-253. PRCM_CM_PER_UART1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.63 PRCM_CM_PER_UART2_CLKCTRL Register (Offset = 588h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_UART2_CLKCTRL is shown in Figure 6-238 and described in Table 6-254.

Return to Summary Table.

This register manages the UART2 clocks.

Figure 6-238. PRCM_CM_PER_UART2_CLKCTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">IDLEST</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R-3h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">MODULEMODE</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 6-254. PRCM_CM_PER_UART2_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.64 PRCM_CM_PER_UART3_CLKCTRL Register (Offset = 590h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_UART3_CLKCTRL is shown in Figure 6-239 and described in Table 6-255.

Return to Summary Table.

This register manages the UART3 clocks.

Figure 6-239. PRCM_CM_PER_UART3_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-255. PRCM_CM_PER_UART3_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.65 PRCM_CM_PER_UART4_CLKCTRL Register (Offset = 598h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_UART4_CLKCTRL is shown in Figure 6-240 and described in Table 6-256.

Return to Summary Table.

This register manages the UART4 clocks.

Figure 6-240. PRCM_CM_PER_UART4_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-256. PRCM_CM_PER_UART4_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.66 PRCM_CM_PER_UART5_CLKCTRL Register (Offset = 5A0h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_UART5_CLKCTRL is shown in Figure 6-241 and described in Table 6-257.

Return to Summary Table.

This register manages the UART5 clocks.

Figure 6-241. PRCM_CM_PER_UART5_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-257. PRCM_CM_PER_UART5_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.67 PRCM_CM_PER_USBPHYOCP2SCP0_CLKCTRL Register (Offset = 5B8h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_USBPHYOCP2SCP0_CLKCTRL is shown in Figure 6-242 and described in Table 6-258.

Return to Summary Table.

This register manages the USBPHYOCP2SCP0 clocks and the optional clock of USB PHY.

Figure 6-242. PRCM_CM_PER_USBPHYOCP2SCP0_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-258. PRCM_CM_PER_USBPHYOCP2SCP0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. [warm reset insensitive] 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.68 PRCM_CM_PER_USBPHYOCP2SCP1_CLKCTRL Register (Offset = 5C0h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_USBPHYOCP2SCP1_CLKCTRL is shown in Figure 6-243 and described in Table 6-259.

Return to Summary Table.

This register manages the USBPHYOCP2SCP1 clocks and the optional clock of USB PHY.

Figure 6-243. PRCM_CM_PER_USBPHYOCP2SCP1_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-259. PRCM_CM_PER_USBPHYOCP2SCP1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. [warm reset insensitive] 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.69 PRCM_CM_PER_EMIF_CLKSTCTRL Register (Offset = 700h) [reset = 2h]

Register mask: FFFFFFFFh

PRCM_CM_PER_EMIF_CLKSTCTRL is shown in Figure 6-244 and described in Table 6-260.

Return to Summary Table.

This register enables the clock domain state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

Figure 6-244. PRCM_CM_PER_EMIF_CLKSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED					CLKACTIVITY_EMIF_PHY_GCLK	CLKACTIVITY_DLL_GCLK	CLKACTIVITY_EMIF_L3_GCLK
R-0-0h					R-0h	R-0h	R-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						CLKTRCTRL	
R-0-0h						R/W-2h	

Table 6-260. PRCM_CM_PER_EMIF_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	
10	CLKACTIVITY_EMIF_PHY_GCLK	R	0h	This field indicates the state of the EMIF PHY clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
9	CLKACTIVITY_DLL_GCLK	R	0h	This field indicates the state of the DLL clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
8	CLKACTIVITY_EMIF_L3_GCLK	R	0h	This field indicates the state of the EMIF L3 clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
7-2	RESERVED	R-0	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the ICSS OCP clock domain. 0h (R/W) = NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1h (R/W) = SW_SLEEP: Start a software forced sleep transition on the domain. 2h (R/W) = SW_WKUP: Start a software forced wake-up transition on the domain. 3h (R/W) = Reserved

6.13.6.70 PRCM_CM_PER_EMIF_CLKCTRL Register (Offset = 720h) [reset = 00030000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_EMIF_CLKCTRL is shown in Figure 6-245 and described in Table 6-261.

Return to Summary Table.

This register manages the EMIF clocks.

Figure 6-245. PRCM_CM_PER_EMIF_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-261. PRCM_CM_PER_EMIF_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R/W) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.71 PRCM_CM_PER_DLL_CLKCTRL Register (Offset = 728h) [reset = 0h]

Register mask: FFFFFFFFh

PRCM_CM_PER_DLL_CLKCTRL is shown in Figure 6-246 and described in Table 6-262.

Return to Summary Table.

This register manages the DLL clock.

Figure 6-246. PRCM_CM_PER_DLL_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							OPTFCLKEN_DLL_CLK
R-0-0h							R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							
R-0-0h							

Table 6-262. PRCM_CM_PER_DLL_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R-0	0h	
8	OPTFCLKEN_DLL_CLK	R/W	0h	Optional functional clock control. 0h (R/W) = Optional functional clock is disabled. DLL_CLK can be gated when EMIF domain performs sleep transition 1h (R/W) = Optional functional clock is enabled. DLL_CLK is garantied to not be gated if already running.
7-0	RESERVED	R-0	0h	

6.13.6.72 PRCM_CM_PER_LCDC_CLKSTCTRL Register (Offset = 800h) [reset = 2h]

Register mask: FFFFFFFFh

PRCM_CM_PER_LCDC_CLKSTCTRL is shown in Figure 6-247 and described in Table 6-263.

Return to Summary Table.

This register enables the clock domain state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

Figure 6-247. PRCM_CM_PER_LCDC_CLKSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED					CLKACTIVITY_LCDC_GCLK	RESERVED	CLKACTIVITY_LCDC_L3_OCP_GCLK
R-0-0h					R-0h	R-0-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED						CLKTRCTRL	
R-0-0h						R/W-2h	

Table 6-263. PRCM_CM_PER_LCDC_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	
10	CLKACTIVITY_LCDC_GCLK	R	0h	This field indicates the state of the LCD clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
9	RESERVED	R-0	0h	
8	CLKACTIVITY_LCDC_L3_OCP_GCLK	R	0h	This field indicates the state of the LCDC L3 OCP clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
7-2	RESERVED	R-0	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the LCDC OCP clock domain. 0h (R/W) = NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1h (R/W) = SW_SLEEP: Start a software forced sleep transition on the domain. 2h (R/W) = SW_WKUP: Start a software forced wake-up transition on the domain. 3h (R/W) = Reserved

6.13.6.73 PRCM_CM_PER_DSS_CLKSTCTRL Register (Offset = A00h) [reset = 2h]

Register mask: FFFFFFFFh

PRCM_CM_PER_DSS_CLKSTCTRL is shown in Figure 6-248 and described in Table 6-264.

Return to Summary Table.

This register enables the clock domain state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

Figure 6-248. PRCM_CM_PER_DSS_CLKSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED				CLKACTIVITY_DSS_L3_OCP_GCLK	CLKACTIVITY_DSS_SYSCLK	CLKACTIVITY_DSS_CLK	RESERVED
R-0-0h				R-0h	R-0h	R-0h	R-0-0h
7	6	5	4	3	2	1	0
RESERVED						CLKTRCTRL	
R-0-0h						R/W-2h	

Table 6-264. PRCM_CM_PER_DSS_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	
11	CLKACTIVITY_DSS_L3_OCP_GCLK	R	0h	This field indicates the state of the DSS L3 OCP clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
10	CLKACTIVITY_DSS_SYSCLK	R	0h	This field indicates the state of the DSS SYSCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
9	CLKACTIVITY_DSS_CLK	R	0h	This field indicates the state of the DSS CLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
8-2	RESERVED	R-0	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the DSS OCP clock domain. 0h (R/W) = NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1h (R/W) = SW_SLEEP: Start a software forced sleep transition on the domain. 2h (R/W) = SW_WKUP: Start a software forced wake-up transition on the domain. 3h (R/W) = Reserved

6.13.6.74 PRCM_CM_PER_DSS_CLKCTRL Register (Offset = A20h) [reset = 00070000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_DSS_CLKCTRL is shown in Figure 6-249 and described in Table 6-265.

Return to Summary Table.

This register manages the DSS clocks.

Figure 6-249. PRCM_CM_PER_DSS_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED					STBYST	IDLEST	
R-0-0h					R-1h	R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-265. PRCM_CM_PER_DSS_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	STBYST	R	1h	Module standby status. 0h (R) = Module is functional (not in standby) 1h (R) = Module is in standby
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.75 PRCM_CM_PER_CPSW_CLKSTCTRL Register (Offset = B00h) [reset = 1C02h]

Register mask: FFFFFFFFh

PRCM_CM_PER_CPSW_CLKSTCTRL is shown in Figure 6-250 and described in Table 6-266.

Return to Summary Table.

This register enables the clock domain state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

Figure 6-250. PRCM_CM_PER_CPSW_CLKSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED			CLKACTIVITY_CPSW_5MHZ_GCLK	CLKACTIVITY_CPSW_50MHZ_GCLK	CLKACTIVITY_CPSW_250MHZ_GCLK	CLKACTIVITY_CPTS_RFT_GCLK	CLKACTIVITY_CPSW_125MHZ_GCLK
R-0-0h			R-1h	R-1h	R-1h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED						CLKTRCTRL	
R-0-0h						R/W-2h	

Table 6-266. PRCM_CM_PER_CPSW_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R-0	0h	
12	CLKACTIVITY_CPSW_5MHZ_GCLK	R	1h	This field indicates the state of the CPSW_5MHZ_GCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
11	CLKACTIVITY_CPSW_50MHZ_GCLK	R	1h	This field indicates the state of the CPSW_50MHZ_GCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
10	CLKACTIVITY_CPSW_250MHZ_GCLK	R	1h	This field indicates the state of the CPSW_250MHZ_GCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
9	CLKACTIVITY_CPTS_RFT_GCLK	R	0h	This field indicates the state of the CLKACTIVITY_CPTS_RFT_GCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
8	CLKACTIVITY_CPSW_125MHZ_GCLK	R	0h	This field indicates the state of the CPSW 125 MHz OCP clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
7-2	RESERVED	R-0	0h	

Table 6-266. PRCM_CM_PER_CPSW_CLKSTCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the CPSW OCP clock domain. 0h (R/W) = NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1h (R/W) = SW_SLEEP: Start a software forced sleep transition on the domain. 2h (R/W) = SW_WKUP: Start a software forced wake-up transition on the domain. 3h (R/W) = Reserved

6.13.6.76 PRCM_CM_PER_CPGMAC0_CLKCTRL Register (Offset = B20h) [reset = 00070000h]

Register mask: FFFFFFFFh

PRCM_CM_PER_CPGMAC0_CLKCTRL is shown in Figure 6-251 and described in Table 6-267.

Return to Summary Table.

This register manages the CPSW clocks.

Figure 6-251. PRCM_CM_PER_CPGMAC0_CLKCTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="5">RESERVED</td><td>STBYST</td><td colspan="2">IDLEST</td></tr>
<tr><td colspan="5">R-0-0h</td><td>R-1h</td><td colspan="2">R-3h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">MODULEMODE</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 6-267. PRCM_CM_PER_CPGMAC0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	STBYST	R	1h	Module standby status. This bit is warm reset insensitive when CPSW RESET_ISO is enabled 0h (R) = Module is functional (not in standby) 1h (R) = Module is in standby
17-16	IDLEST	R	3h	Module idle status. This bit is warm reset insensitive when CPSW RESET_ISO is enabled 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. This bit is warm reset insensitive when CPSW RESET_ISO is enabled 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.6.77 PRCM_CM_PER_OCPWP_L3_CLKSTCTRL Register (Offset = C00h) [reset = 2h]

Register mask: FFFFFFFFh

PRCM_CM_PER_OCPWP_L3_CLKSTCTRL is shown in Figure 6-252 and described in Table 6-268.

Return to Summary Table.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

Figure 6-252. PRCM_CM_PER_OCPWP_L3_CLKSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							CLKACTIVITY_OCPWP_L3_GCLK
R-0-0h							R-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						CLKTRCTRL	
R-0-0h						R/W-2h	

Table 6-268. PRCM_CM_PER_OCPWP_L3_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R-0	0h	
8	CLKACTIVITY_OCPWP_L3_GCLK	R	0h	This field indicates the state of the OCPWP L3 clock in the domain. 0h (R) = 0 1h (R) = 1
7-2	RESERVED	R-0	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the OCPWP clock domain. 0h (R/W) = NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1h (R/W) = SW_SLEEP: Start a software forced sleep transition on the domain. 2h (R/W) = SW_WKUP: Start a software forced wake-up transition on the domain. 3h (R/W) = Reserved

6.13.6.78 PRCM_CM_PER_OCPWP_CLKCTRL Register (Offset = C20h) [reset = 00030002h]

Register mask: FFFFFFFFh

PRCM_CM_PER_OCPWP_CLKCTRL is shown in Figure 6-253 and described in Table 6-269.

Return to Summary Table.

This register manages the OCPWP clocks.

Figure 6-253. PRCM_CM_PER_OCPWP_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-2h	

Table 6-269. PRCM_CM_PER_OCPWP_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.7 PRCM_CM_RTC Registers

Table 6-270 lists the memory-mapped registers for the PRCM_CM_RTC. All register offset addresses not listed in Table 6-270 should be considered as reserved locations and the register contents should not be modified.

Table 6-270. PRCM_CM_RTC REGISTERS

Offset	Acronym	Register Name	Section
0h	PRCM_CM_RTC_CLKSTCTRL		Section 6.13.7.1
20h	PRCM_CM_RTC_CLKCTRL		Section 6.13.7.2

6.13.7.1 PRCM_CM_RTC_CLKSTCTRL Register (offset = 0h) [reset = 102h]

Register mask: FFFFFFFFh

PRCM_CM_RTC_CLKSTCTRL is shown in Figure 6-254 and described in Table 6-271.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

Figure 6-254. PRCM_CM_RTC_CLKSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							

15	14	13	12	11	10	9	8
RESERVED						CLKACTIVITY_RTC_32KCLK	CLKACTIVITY_L4_RTC_GCLK
Rreturns0s-0h						R-0h	R-1h

7	6	5	4	3	2	1	0
RESERVED						CLKTRCTRL	
Rreturns0s-0h						R/W-2h	

Table 6-271. PRCM_CM_RTC_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	Rreturns0s	0h	
9	CLKACTIVITY_RTC_32KCLK	R	0h	This field indicates the state of the 32K RTC clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
8	CLKACTIVITY_L4_RTC_GCLK	R	1h	This field indicates the state of the L4 RTC clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
7-2	RESERVED	Rreturns0s	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the RTC clock domains. 0h (R/W) = NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1h (R/W) = SW_SLEEP: Start a software forced sleep transition on the domain. 2h (R/W) = SW_WKUP: Start a software forced wake-up transition on the domain. 3h (R/W) = Reserved

6.13.7.2 PRCM_CM_RTC_CLKCTRL Register (offset = 20h) [reset = 30002h]

Register mask: FFFFFFFFh

PRCM_CM_RTC_CLKCTRL is shown in Figure 6-255 and described in Table 6-272.

This register manages the RTC clocks.

Figure 6-255. PRCM_CM_RTC_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							

23	22	21	20	19	18	17	16
RESERVED					RESERVED	IDLEST	
Rreturns0s-0h					Rreturns0s-0h	R-3h	

15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							

7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
Rreturns0s-0h						R/W-2h	

Table 6-272. PRCM_CM_RTC_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	Rreturns0s	0h	
18	RESERVED	Rreturns0s	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	Rreturns0s	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.8 CM_WKUP Registers

Table 6-273 lists the memory-mapped registers for the CM_WKUP. All register offset addresses not listed in Table 6-273 should be considered as reserved locations and the register contents should not be modified.

Table 6-273. CM_WKUP Registers

Offset	Acronym	Register Name	Section
0h	PRCM_CM_L3_AON_CLKSTCTRL		Section 6.13.8.1
20h	PRCM_CM_WKUP_DBGSS_CLKCTRL		Section 6.13.8.2

Table 6-273. CM_WKUP Registers (continued)

Offset	Acronym	Register Name	Section
100h	PRCM_CM_L3S_TSC_CLKSTCTRL		Section 6.13.8.3
100h	PRCM_CM_L3S_ADC0_CLKSTCTRL		Section 6.13.8.4
120h	PRCM_CM_WKUP_ADC_TSC_CLKCTRL		Section 6.13.8.5
120h	PRCM_CM_WKUP_ADC0_CLKCTRL		Section 6.13.8.6
200h	PRCM_CM_L4_WKUP_AON_CLKSTCTRL		Section 6.13.8.7
220h	PRCM_CM_WKUP_L4WKUP_CLKCTRL		Section 6.13.8.8
228h	PRCM_CM_WKUP_M3_CLKCTRL		Section 6.13.8.9
228h	PRCM_CM_WKUP_PROC_CLKCTRL		Section 6.13.8.10
230h	PRCM_CM_WKUP_SYNCTIMER_CLKCTRL		Section 6.13.8.11
238h	PRCM_CM_WKUP_CLKDIV32K_CLKCTRL		Section 6.13.8.12
240h	PRCM_CM_WKUP_USBPHY0_CLKCTRL		Section 6.13.8.13
248h	PRCM_CM_WKUP_USBPHY1_CLKCTRL		Section 6.13.8.14
300h	PRCM_CM_WKUP_CLKSTCTRL		Section 6.13.8.15
320h	PRCM_CM_WKUP_TIMER0_CLKCTRL		Section 6.13.8.16
328h	PRCM_CM_WKUP_TIMER1_CLKCTRL		Section 6.13.8.17
330h	PRCM_CM_WKUP_WDT0_CLKCTRL		Section 6.13.8.18
338h	PRCM_CM_WKUP_WDT1_CLKCTRL		Section 6.13.8.19
340h	PRCM_CM_WKUP_I2C0_CLKCTRL		Section 6.13.8.20
348h	PRCM_CM_WKUP_UART0_CLKCTRL		Section 6.13.8.21
360h	PRCM_CM_WKUP_CTRL_CLKCTRL		Section 6.13.8.22
368h	PRCM_CM_WKUP_GPIO0_CLKCTRL		Section 6.13.8.23
520h	PRCM_CM_CLKMODE_DPLL_CORE		Section 6.13.8.24
524h	PRCM_CM_IDLEST_DPLL_CORE		Section 6.13.8.25
52Ch	PRCM_CM_CLKSEL_DPLL_CORE		Section 6.13.8.26
538h	PRCM_CM_DIV_M4_DPLL_CORE		Section 6.13.8.27
53Ch	PRCM_CM_DIV_M5_DPLL_CORE		Section 6.13.8.28
540h	PRCM_CM_DIV_M6_DPLL_CORE		Section 6.13.8.29
548h	PRCM_CM_SSC_DELTAMSTEP_DPLL_CORE		Section 6.13.8.30
54Ch	PRCM_CM_SSC_MODFREQDIV_DPLL_CORE		Section 6.13.8.31
560h	PRCM_CM_CLKMODE_DPLL_MPU		Section 6.13.8.32
564h	PRCM_CM_IDLEST_DPLL_MPU		Section 6.13.8.33
56Ch	PRCM_CM_CLKSEL_DPLL_MPU		Section 6.13.8.34
570h	PRCM_CM_DIV_M2_DPLL_MPU		Section 6.13.8.35
588h	PRCM_CM_SSC_DELTAMSTEP_DPLL_MPU		Section 6.13.8.36
58Ch	PRCM_CM_SSC_MODFREQDIV_DPLL_MPU		Section 6.13.8.37
5A0h	PRCM_CM_CLKMODE_DPLL_DDR		Section 6.13.8.38
5A4h	PRCM_CM_IDLEST_DPLL_DDR		Section 6.13.8.39
5ACh	PRCM_CM_CLKSEL_DPLL_DDR		Section 6.13.8.40
5B0h	PRCM_CM_DIV_M2_DPLL_DDR		Section 6.13.8.41
5B8h	PRCM_CM_DIV_M4_DPLL_DDR		Section 6.13.8.42
5C8h	PRCM_CM_SSC_DELTAMSTEP_DPLL_DDR		Section 6.13.8.43

Table 6-273. CM_WKUP Registers (continued)

Offset	Acronym	Register Name	Section
5CCh	PRCM_CM_SSC_MODFREQDIV_DPLL_DDR		Section 6.13.8.44
5E0h	PRCM_CM_CLKMODE_DPLL_PER		Section 6.13.8.45
5E4h	PRCM_CM_IDLEST_DPLL_PER		Section 6.13.8.46
5ECh	PRCM_CM_CLKSEL_DPLL_PER		Section 6.13.8.47
5F0h	PRCM_CM_DIV_M2_DPLL_PER		Section 6.13.8.48
604h	PRCM_CM_CLKSEL2_DPLL_PER		Section 6.13.8.49
608h	PRCM_CM_SSC_DELTAMSTEP_DPLL_PER		Section 6.13.8.50
60Ch	PRCM_CM_SSC_MODFREQDIV_DPLL_PER		Section 6.13.8.51
614h	PRCM_CM_CLKDCOLDO_DPLL_PER		Section 6.13.8.52
620h	PRCM_CM_CLKMODE_DPLL_DISP		Section 6.13.8.53
624h	PRCM_CM_IDLEST_DPLL_DISP		Section 6.13.8.54
62Ch	PRCM_CM_CLKSEL_DPLL_DISP		Section 6.13.8.55
630h	PRCM_CM_DIV_M2_DPLL_DISP		Section 6.13.8.56
648h	PRCM_CM_SSC_DELTAMSTEP_DPLL_DISP		Section 6.13.8.57
64Ch	PRCM_CM_SSC_MODFREQDIV_DPLL_DISP		Section 6.13.8.58
660h	PRCM_CM_CLKMODE_DPLL_EXTDEV		Section 6.13.8.59
664h	PRCM_CM_IDLEST_DPLL_EXTDEV		Section 6.13.8.60
66Ch	PRCM_CM_CLKSEL_DPLL_EXTDEV		Section 6.13.8.61
670h	PRCM_CM_DIV_M2_DPLL_EXTDEV		Section 6.13.8.62
684h	PRCM_CM_CLKSEL2_DPLL_EXTDEV		Section 6.13.8.63
688h	PRCM_CM_SSC_DELTAMSTEP_DPLL_EXTDEV		Section 6.13.8.64
68Ch	PRCM_CM_SSC_MODFREQDIV_DPLL_EXTDEV		Section 6.13.8.65
7A0h	PRCM_CM_SHADOW_FREQ_CONFIG1		Section 6.13.8.66
7A4h	PRCM_CM_SHADOW_FREQ_CONFIG2		Section 6.13.8.67

6.13.8.1 PRCM_CM_L3_AON_CLKSTCTRL Register (Offset = 0h) [reset = 1E02h]

PRCM_CM_L3_AON_CLKSTCTRL is shown in Figure 6-256 and described in Table 6-274.

Return to Summary Table.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

Figure 6-256. PRCM_CM_L3_AON_CLKSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED			CLKACTIVITY_DBG_CLKC	CLKACTIVITY_DBG_CLKB	CLKACTIVITY_DBG_CLKA	CLKACTIVITY_L3_AON_GCLK	CLKACTIVITY_DBGSYSCLK
R-0-0h			R-1h	R-1h	R-1h	R-1h	R-0h
7	6	5	4	3	2	1	0
RESERVED						CLKTRCTRL	
R-0h						R/W-2h	

Table 6-274. PRCM_CM_L3_AON_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R-0	0h	
12	CLKACTIVITY_DBG_CLKC	R	1h	This field indicates the state of the Debugss CLKC clock in the domain.
11	CLKACTIVITY_DBG_CLKB	R	1h	This field indicates the state of the Debugss CLKB clock in the domain.
10	CLKACTIVITY_DBG_CLKA	R	1h	This field indicates the state of the Debugss CLKA clock in the domain.
9	CLKACTIVITY_L3_AON_GCLK	R	1h	This field indicates the state of the L3_AON clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
8	CLKACTIVITY_DBGSYSCLK	R	0h	This field indicates the state of the Debugss sysclk clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
7-2	RESERVED	R	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the l3 AON clock domain. 0h (R/W) = NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1h (R/W) = SW_SLEEP: Start a software forced sleep transition on the domain. 2h (R/W) = SW_WKUP: Start a software forced wake-up transition on the domain. 3h (R/W) = Reserved

6.13.8.2 PRCM_CM_WKUP_DBGSS_CLKCTRL Register (Offset = 20h) [reset = 12540F02h]

PRCM_CM_WKUP_DBGSS_CLKCTRL is shown in Figure 6-257 and described in Table 6-275.

Return to Summary Table.

This register manages the DEBUGSS clocks. [warm reset insensitive]

Figure 6-257. PRCM_CM_WKUP_DBGSS_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED		STM_PMD_CLKDIVSEL			TRC_PMD_CLKDIVSEL		
R-0h		R/W-2h			R/W-2h		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TRC_PMD_CLKSEL		STM_PMD_CLKSEL		RESERVED	STBYST	IDLEST	
R/W-1h		R/W-1h		R-0h	R-1h	R-0h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED				OPTCLK_DBG _CLKC	OPTCLK_DBG _CLKB	OPTCLK_DBG _CLKA	OPTFCLKEN_ DBGSYSCLK
R-0h				R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						MODULEMODE	
R-0-0h						R/W-2h	

Table 6-275. PRCM_CM_WKUP_DBGSS_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	Reserved
29-27	STM_PMD_CLKDIVSEL	R/W	2h	STM Trace clock divider control 0h (R/W) = Divide by 1 2h (R/W) = Divide by 2 4h (R/W) = Divide by 4
26-24	TRC_PMD_CLKDIVSEL	R/W	2h	TPIU trace clock divider control 0h (R/W) = Divide by 1 2h (R/W) = Divide by 2 4h (R/W) = Divide by 4
23-22	TRC_PMD_CLKSEL	R/W	1h	TPIU Trace clock select 0h (R/W) = Selects DGBSYSCLK as TPIU trace clock 1h (R/W) = Selects CLKA as TPIU trace clock 2h (R/W) = Selects CLKB as TPIU trace clock 3h (R/W) = Selects CLKC as TPIU trace clock
21-20	STM_PMD_CLKSEL	R/W	1h	STM trace clock select 0h (R/W) = Selects DGBSYSCLK as STM trace clock 1h (R/W) = Selects CLKA as STM trace clock 2h (R/W) = Selects CLKB as STM trace clock 3h (R/W) = Selects CLKC as STM trace clock
19	RESERVED	R	0h	
18	STBYST	R	1h	Module standby status. 0h (R) = Module is functional (not in standby) 1h (R) = Module is in standby
17-16	IDLEST	R	0h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed

Table 6-275. PRCM_CM_WKUP_DBGSS_CLKCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	
11	OPTCLK_DBG_CLKC	R/W	1h	Controls Optional Functional Clock CLKC 0h (R/W) = Disable optional clock DEBUG_CLKC 1h (R/W) = Enable optional clock DEBUG_CLKC
10	OPTCLK_DBG_CLKB	R/W	1h	Controls Optional Functional Clock CLKB 0h (R/W) = Disable optional clock DEBUG_CLKB 1h (R/W) = Enable optional clock DEBUG_CLKB
9	OPTCLK_DBG_CLKA	R/W	1h	Optional functional clock control 0h (R/W) = Disable optional clock DEBUG_CLKA 1h (R/W) = Enable optional clock DEBUG_CLKA
8	OPTFCLKEN_DBGSYSCLK	R/W	1h	Optional functional clock control. 0h (R/W) = Optional functional clock is disabled 1h (R/W) = Optional functional clock is enabled
7-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.8.3 PRCM_CM_L3S_TSC_CLKSTCTRL Register (Offset = 100h) [reset = 2h]

PRCM_CM_L3S_TSC_CLKSTCTRL is shown in Figure 6-258 and described in Table 6-276.

Return to Summary Table.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

Figure 6-258. PRCM_CM_L3S_TSC_CLKSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED						CLKACTIVITY_ADC_FCLK	CLKACTIVITY_L3S_TSC_GCLK
R-0-0h						R-0h	R-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						CLKTRCTRL	
R-0-0h						R/W-2h	

Table 6-276. PRCM_CM_L3S_TSC_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R-0	0h	
9	CLKACTIVITY_ADC_FCLK	R	0h	This field indicates the state of the TSC ADC FCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
8	CLKACTIVITY_L3S_TSC_GCLK	R	0h	This field indicates the state of the L3S_TSC clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
7-2	RESERVED	R-0	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the always on clock domain. 0h (R/W) = NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1h (R/W) = SW_SLEEP: Start a software forced sleep transition on the domain. 2h (R/W) = SW_WKUP: Start a software forced wake-up transition on the domain. 3h (R/W) = Reserved

6.13.8.4 PRCM_CM_L3S_ADC0_CLKSTCTRL Register (Offset = 100h) [reset = 2h]

PRCM_CM_L3S_ADC0_CLKSTCTRL is shown in Figure 6-259 and described in Table 6-277.

Return to Summary Table.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

Figure 6-259. PRCM_CM_L3S_ADC0_CLKSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED						CLKACTIVITY_ADC0_FCLK	CLKACTIVITY_L3S_ADC0_GCLK
R-0-0h						R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED						CLKTRCTRL	
R-0-0h						R/W-2h	

Table 6-277. PRCM_CM_L3S_ADC0_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R-0	0h	
9	CLKACTIVITY_ADC0_FCLK	R	0h	This field indicates the state of the ADC0 FCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
8	CLKACTIVITY_L3S_ADC0_GCLK	R	0h	This field indicates the state of the L3S_ADC0 clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
7-2	RESERVED	R-0	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the always on clock domain. 0h (R/W) = NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1h (R/W) = SW_SLEEP: Start a software forced sleep transition on the domain. 2h (R/W) = SW_WKUP: Start a software forced wake-up transition on the domain. 3h (R/W) = Reserved

6.13.8.5 PRCM_CM_WKUP_ADC_TSC_CLKCTRL Register (Offset = 120h) [reset = 00030000h]

PRCM_CM_WKUP_ADC_TSC_CLKCTRL is shown in Figure 6-260 and described in Table 6-278.

Return to Summary Table.

This register manages the ADC clocks.

Figure 6-260. PRCM_CM_WKUP_ADC_TSC_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-278. PRCM_CM_WKUP_ADC_TSC_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.8.6 PRCM_CM_WKUP_ADC0_CLKCTRL Register (Offset = 120h) [reset = 00030000h]

PRCM_CM_WKUP_ADC0_CLKCTRL is shown in Figure 6-261 and described in Table 6-279.

Return to Summary Table.

This register manages the ADC0 clocks.

Figure 6-261. PRCM_CM_WKUP_ADC0_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-279. PRCM_CM_WKUP_ADC0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.8.7 PRCM_CM_L4_WKUP_AON_CLKSTCTRL Register (Offset = 200h) [reset = 702h]

PRCM_CM_L4_WKUP_AON_CLKSTCTRL is shown in Figure 6-262 and described in Table 6-280.

Return to Summary Table.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

Figure 6-262. PRCM_CM_L4_WKUP_AON_CLKSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED					CLKACTIVITY_USBPHY_32KHZ_GCLK	CLKACTIVITY_SYNCTIMER32K_GFCLK	CLKACTIVITY_L4_WKUP_AON_GCLK
R-0-0h					R-1h	R-1h	R-1h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						CLKTRCTRL	
R-0-0h						Rreturns-2h	

Table 6-280. PRCM_CM_L4_WKUP_AON_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	
10	CLKACTIVITY_USBPHY_32KHZ_GCLK	R	1h	This field indicates the state of the USBPHY 32KHZ clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
9	CLKACTIVITY_SYNCTIMER32K_GFCLK	R	1h	This field indicates the state of the SYNCTIMER clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
8	CLKACTIVITY_L4_WKUP_AON_GCLK	R	1h	This field indicates the state of the L4_WKUP clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
7-2	RESERVED	R-0	0h	
1-0	CLKTRCTRL	Rreturns	2h	Controls the clock state transition of the always on L4 clock domain.

6.13.8.8 PRCM_CM_WKUP_L4WKUP_CLKCTRL Register (Offset = 220h) [reset = 2h]

PRCM_CM_WKUP_L4WKUP_CLKCTRL is shown in Figure 6-263 and described in Table 6-281.

Return to Summary Table.

This register manages the L4WKUP clocks.

Figure 6-263. PRCM_CM_WKUP_L4WKUP_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED					RESERVED	IDLEST	
R-0-0h					R-0-0h	R-0h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						MODULEMODE	
R-0-0h						Rreturns-2h	

Table 6-281. PRCM_CM_WKUP_L4WKUP_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	RESERVED	R-0	0h	
17-16	IDLEST	R	0h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	Rreturns	2h	Control the way mandatory clocks are managed.

6.13.8.9 PRCM_CM_WKUP_M3_CLKCTRL Register (Offset = 228h) [reset = 00040002h]

PRCM_CM_WKUP_M3_CLKCTRL is shown in Figure 6-264 and described in Table 6-282.

Return to Summary Table.

This register manages the WKUP M3 clocks.

Figure 6-264. PRCM_CM_WKUP_M3_CLKCTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="5">RESERVED</td><td>STBYST</td><td colspan="2">RESERVED</td></tr>
<tr><td colspan="5">R-0-0h</td><td>R-1h</td><td colspan="2">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">MODULEMODE</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">Rreturns-2h</td></tr>
</table>

Table 6-282. PRCM_CM_WKUP_M3_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	STBYST	R	1h	Module standby status. 0h (R) = Module is functional (not in standby) 1h (R) = Module is in standby
17-2	RESERVED	R-0	0h	
1-0	MODULEMODE	Rreturns	2h	Control the way mandatory clocks are managed.

6.13.8.10 PRCM_CM_WKUP_PROC_CLKCTRL Register (Offset = 228h) [reset = 00040002h]

PRCM_CM_WKUP_PROC_CLKCTRL is shown in Figure 6-265 and described in Table 6-283.

Return to Summary Table.

This register manages the WKUP M3 clocks.

Figure 6-265. PRCM_CM_WKUP_PROC_CLKCTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="5">RESERVED</td><td>STBYST</td><td colspan="2">RESERVED</td></tr>
<tr><td colspan="5">R-0-0h</td><td>R-1h</td><td colspan="2">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">MODULEMODE</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">Rreturns-2h</td></tr>
</table>

Table 6-283. PRCM_CM_WKUP_PROC_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	STBYST	R	1h	Module standby status. 0h (R) = Module is functional (not in standby) 1h (R) = Module is in standby
17-2	RESERVED	R-0	0h	
1-0	MODULEMODE	Rreturns	2h	Control the way mandatory clocks are managed.

6.13.8.11 PRCM_CM_WKUP_SYNCTIMER_CLKCTRL Register (Offset = 230h) [reset = 00030002h]

PRCM_CM_WKUP_SYNCTIMER_CLKCTRL is shown in Figure 6-266 and described in Table 6-284.

Return to Summary Table.

This register manages the SYNCTIMER clocks.

Figure 6-266. PRCM_CM_WKUP_SYNCTIMER_CLKCTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">IDLEST</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R-3h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>OPTFCLKEN_
FCLK32</td></tr>
<tr><td colspan="7">R-0-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">MODULEMODE</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R/W-2h</td></tr>
</table>

Table 6-284. PRCM_CM_WKUP_SYNCTIMER_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-9	RESERVED	R-0	0h	
8	OPTFCLKEN_FCLK32	R/W	0h	Optional functional clock control. 0h (R/W) = Optional functional clock is disabled 1h (R/W) = Optional functional clock is enabled
7-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.8.12 PRCM_CM_WKUP_CLKDIV32K_CLKCTRL Register (Offset = 238h) [reset = 0h]

PRCM_CM_WKUP_CLKDIV32K_CLKCTRL is shown in Figure 6-267 and described in Table 6-285.

Return to Summary Table.

This register manages the CLKDIV32K clocks.

Figure 6-267. PRCM_CM_WKUP_CLKDIV32K_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							OPTFCLKEN_FCLK
R-0-0h							R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							
R-0-0h							

Table 6-285. PRCM_CM_WKUP_CLKDIV32K_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R-0	0h	
8	OPTFCLKEN_FCLK	R/W	0h	Optional functional clock control. 0h (R/W) = Optional functional clock is disabled 1h (R/W) = Optional functional clock is enabled
7-0	RESERVED	R-0	0h	

6.13.8.13 PRCM_CM_WKUP_USBPHY0_CLKCTRL Register (Offset = 240h) [reset = 0h]

PRCM_CM_WKUP_USBPHY0_CLKCTRL is shown in Figure 6-268 and described in Table 6-286.

Return to Summary Table.

This register manages the USBPHY0 32KHz clocks.

Figure 6-268. PRCM_CM_WKUP_USBPHY0_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							OPTFCLKEN_CLK32K
R-0-0h							R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							
R-0-0h							

Table 6-286. PRCM_CM_WKUP_USBPHY0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R-0	0h	
8	OPTFCLKEN_CLK32K	R/W	0h	Optional functional clock control. 0h (R/W) = Optional functional clock is disabled 1h (R/W) = Optional functional clock is enabled
7-0	RESERVED	R-0	0h	

6.13.8.14 PRCM_CM_WKUP_USBPHY1_CLKCTRL Register (Offset = 248h) [reset = 0h]

PRCM_CM_WKUP_USBPHY1_CLKCTRL is shown in Figure 6-269 and described in Table 6-287.

Return to Summary Table.

This register manages the USBPHY1 32KHz clocks.

Figure 6-269. PRCM_CM_WKUP_USBPHY1_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							OPTFCLKEN_CLK32K
R-0-0h							R/W-0h
7	6	5	4	3	2	1	0
RESERVED							
R-0-0h							

Table 6-287. PRCM_CM_WKUP_USBPHY1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R-0	0h	
8	OPTFCLKEN_CLK32K	R/W	0h	Optional functional clock control. 0h (R/W) = Optional functional clock is disabled 1h (R/W) = Optional functional clock is enabled
7-0	RESERVED	R-0	0h	

6.13.8.15 PRCM_CM_WKUP_CLKSTCTRL Register (Offset = 300h) [reset = 102h]

PRCM_CM_WKUP_CLKSTCTRL is shown in Figure 6-270 and described in Table 6-288.

Return to Summary Table.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

Figure 6-270. PRCM_CM_WKUP_CLKSTCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							

23	22	21	20	19	18	17	16
RESERVED							CLKACTIVITY_TIMER1_GCLK
R-0-0h							R-0h

15	14	13	12	11	10	9	8
CLKACTIVITY_UART0_GFCLK	CLKACTIVITY_I2C0_GFCLK	RESERVED		CLKACTIVITY_GPIO0_GDBCLK	CLKACTIVITY_WDT1_GCLK	RESERVED	CLKACTIVITY_L4_WKUP_GCLK
R-0h	R-0h	R-0-0h		R-0h	R-0h	R-0h	R-1h

7	6	5	4	3	2	1	0
RESERVED						CLKTRCTRL	
R-0-0h						R/W-2h	

Table 6-288. PRCM_CM_WKUP_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R-0	0h	
16	CLKACTIVITY_TIMER1_GCLK	R	0h	This field indicates the state of the TIMER1 clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
15	CLKACTIVITY_UART0_GFCLK	R	0h	This field indicates the state of the UART0 clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
14	CLKACTIVITY_I2C0_GFCLK	R	0h	This field indicates the state of the I2C0 clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
13-12	RESERVED	R-0	0h	
11	CLKACTIVITY_GPIO0_GDBCLK	R	0h	This field indicates the state of the WKUPGPIO_DBGICLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
10	CLKACTIVITY_WDT1_GCLK	R	0h	This field indicates the state of the WDT1_GCLK clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
9	RESERVED	R	0h	Reserved
8	CLKACTIVITY_L4_WKUP_GCLK	R	1h	This field indicates the state of the L4_WKUP clock in the domain. 0h (R) = Corresponding clock is gated 1h (R) = Corresponding clock is active
7-2	RESERVED	R-0	0h	

Table 6-288. PRCM_CM_WKUP_CLKSTCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the always on clock domain. 0h (R/W) = NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1h (R/W) = SW_SLEEP: Start a software forced sleep transition on the domain. 2h (R/W) = SW_WKUP: Start a software forced wake-up transition on the domain. 3h (R/W) = Reserved

6.13.8.16 PRCM_CM_WKUP_TIMER0_CLKCTRL Register (Offset = 320h) [reset = 00030002h]

PRCM_CM_WKUP_TIMER0_CLKCTRL is shown in Figure 6-271 and described in Table 6-289.

Return to Summary Table.

This register manages the TIMER0 clocks.

Figure 6-271. PRCM_CM_WKUP_TIMER0_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED					RESERVED	IDLEST	
R-0-0h					R-0-0h	R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-2h	

Table 6-289. PRCM_CM_WKUP_TIMER0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.8.17 PRCM_CM_WKUP_TIMER1_CLKCTRL Register (Offset = 328h) [reset = 00030000h]

PRCM_CM_WKUP_TIMER1_CLKCTRL is shown in Figure 6-272 and described in Table 6-290.

Return to Summary Table.

This register manages the TIMER1 clocks.

Figure 6-272. PRCM_CM_WKUP_TIMER1_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED					RESERVED	IDLEST	
R-0-0h					R-0-0h	R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-290. PRCM_CM_WKUP_TIMER1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.8.18 PRCM_CM_WKUP_WDT0_CLKCTRL Register (Offset = 330h) [reset = 00030002h]

PRCM_CM_WKUP_WDT0_CLKCTRL is shown in Figure 6-273 and described in Table 6-291.

Return to Summary Table.

This register manages the WDT0 clocks.

Figure 6-273. PRCM_CM_WKUP_WDT0_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED					RESERVED	IDLEST	
R-0-0h					R-0-0h	R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-2h	

Table 6-291. PRCM_CM_WKUP_WDT0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.8.19 PRCM_CM_WKUP_WDT1_CLKCTRL Register (Offset = 338h) [reset = 00030002h]

PRCM_CM_WKUP_WDT1_CLKCTRL is shown in Figure 6-274 and described in Table 6-292.

Return to Summary Table.

This register manages the WDT1 clocks.

Figure 6-274. PRCM_CM_WKUP_WDT1_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							

23	22	21	20	19	18	17	16
RESERVED					RESERVED	IDLEST	
R-0-0h					R-0-0h	R-3h	

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							

7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-2h	

Table 6-292. PRCM_CM_WKUP_WDT1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.8.20 PRCM_CM_WKUP_I2C0_CLKCTRL Register (Offset = 340h) [reset = 00030000h]

PRCM_CM_WKUP_I2C0_CLKCTRL is shown in Figure 6-275 and described in Table 6-293.

Return to Summary Table.

This register manages the I2C0 clocks.

Figure 6-275. PRCM_CM_WKUP_I2C0_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						IDLEST	
R-0-0h						R-3h	
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-293. PRCM_CM_WKUP_I2C0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.8.21 PRCM_CM_WKUP_UART0_CLKCTRL Register (Offset = 348h) [reset = 00030000h]

PRCM_CM_WKUP_UART0_CLKCTRL is shown in Figure 6-276 and described in Table 6-294.

Return to Summary Table.

This register manages the UART0 clocks.

Figure 6-276. PRCM_CM_WKUP_UART0_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED						IDLEST	
R-0-0h						R-3h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-294. PRCM_CM_WKUP_UART0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.8.22 PRCM_CM_WKUP_CTRL_CLKCTRL Register (Offset = 360h) [reset = 00030002h]

PRCM_CM_WKUP_CTRL_CLKCTRL is shown in Figure 6-277 and described in Table 6-295.

Return to Summary Table.

This register manages the Control Module clocks.

Figure 6-277. PRCM_CM_WKUP_CTRL_CLKCTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="5">RESERVED</td><td>RESERVED</td><td colspan="2">IDLEST</td></tr>
<tr><td colspan="5">R-0-0h</td><td>R-0-0h</td><td colspan="2">R-3h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">MODULEMODE</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R/W-2h</td></tr>
</table>

Table 6-295. PRCM_CM_WKUP_CTRL_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R-0	0h	
18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.8.23 PRCM_CM_WKUP_GPIO0_CLKCTRL Register (Offset = 368h) [reset = 00030000h]

PRCM_CM_WKUP_GPIO0_CLKCTRL is shown in Figure 6-278 and described in Table 6-296.

Return to Summary Table.

This register manages the GPIO0 clocks.

Figure 6-278. PRCM_CM_WKUP_GPIO0_CLKCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED						IDLEST	
R-0-0h						R-3h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							OPTFCLKEN_GPIO0_GDBCLK
R-0h							R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						MODULEMODE	
R-0-0h						R/W-0h	

Table 6-296. PRCM_CM_WKUP_GPIO0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17-16	IDLEST	R	3h	Module idle status. 0h (R) = Module is fully functional, including OCP 1h (R) = Module is performing transition: wakeup, or sleep, or sleep abortion 2h (R) = Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3h (R) = Module is disabled and cannot be accessed
15-9	RESERVED	R	0h	
8	OPTFCLKEN_GPIO0_GDBCLK	R/W	0h	Optional functional clock control. 0h (R/W) = Optional functional clock is disabled 1h (R/W) = Optional functional clock is enabled
7-2	RESERVED	R-0	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0h (R/W) = Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1h (R) = Reserved 2h (R/W) = Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3h (R) = Reserved

6.13.8.24 PRCM_CM_CLKMODE_DPLL_CORE Register (Offset = 520h) [reset = 4h]

PRCM_CM_CLKMODE_DPLL_CORE is shown in Figure 6-279 and described in Table 6-297.

Return to Summary Table.

This register allows controlling the DPLL modes.

Figure 6-279. PRCM_CM_CLKMODE_DPLL_CORE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
DPLL_SSC_TYPE	DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	DPLL_REGM4XEN	DPLL_LPMODE_EN	DPLL_RELOCK_RAMP_EN	DPLL_DRIFTGUARD_EN
R/W-0h	R/W-0h	R-0h	R/W-0h	R-0-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
DPLL_RAMP_RATE			DPLL_RAMP_LEVEL		DPLL_EN		
R/W-0h			R/W-0h		R/W-4h		

Table 6-297. PRCM_CM_CLKMODE_DPLL_CORE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	
15	DPLL_SSC_TYPE	R/W	0h	Select between Triangular and SquareWave Spread Spectrum Clocking 0h (R/W) = Triangular Spread Spectrum Clocking is selected 1h (R/W) = Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)
14	DPLL_SSC_DOWNSPREAD	R/W	0h	Control if only low frequency spread is required 0h (R/W) = When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 1h (R/W) = When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency
13	DPLL_SSC_ACK	R	0h	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0h (R) = SSC has been turned off on PLL o/ps 1h (R) = SSC has been turned on on PLL o/ps
12	DPLL_SSC_EN	R/W	0h	Enable or disable Spread Spectrum Clocking 0h (R/W) = SSC disabled 1h (R/W) = SSC enabled
11	DPLL_REGM4XEN	R-0	0h	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0h (R) = REGM4XEN mode of the DPLL is disabled
10	DPLL_LPMODE_EN	R/W	0h	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0h (R/W) = Low power mode of the DPLL is disabled 1h (R/W) = Low power mode of the DPLL is enabled
9	DPLL_RELOCK_RAMP_EN	R/W	0h	If enabled, the clock ramping feature is used applied during the lock process, as well as the relock process. If disabled, the clock ramping feature is used only during the first lock.

Table 6-297. PRCM_CM_CLKMODE_DPLL_CORE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	DPLL_DRIFTGUARD_EN	R/W	0h	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0h (R/W) = DRIFTGUARD feature is disabled 1h (R/W) = DRIFTGUARD feature is enabled
7-5	DPLL_RAMP_RATE	R/W	0h	Selects the time in terms of DPLL REFCLKs spent at each stage of the clock ramping process 0h (R/W) = 2 REFCLKs 1h (R/W) = 4 REFCLKs 2h (R/W) = 8 REFCLKs 3h (R/W) = 16 REFCLKs 4h (R/W) = 32 REFCLKs 5h (R/W) = 64 REFCLKs 6h (R/W) = 128 REFCLKs 7h (R/W) = 512 REFCLKs
4-3	DPLL_RAMP_LEVEL	R/W	0h	The DPLL provides an output clock frequency ramping feature when switching from bypass clock to normal clock during lock and re-lock. The frequency ramping will happen in a maximum of 4 steps in frequency before the DPLL's frequency lock indicator is asserted. This register is used to enable/disable the DPLL ramping feature. If enabled, it is also used to select the algorithm used for clock ramping 0h (R/W) = CLKOUT => No ramping CLKOUTX2 => No ramping 1h (R/W) = CLKOUT => Bypass clk -> Fout/8 -> Fout/4 -> Fout/2 -> Fout CLKOUTX2 => Bypass clk -> Foutx2/8 -> Foutx2/4 -> Foutx2/2 -> Foutx2 2h (R/W) = CLKOUT => Bypass clk -> Fout/4 -> Fout/2 -> Fout/1.5 -> Fout CLKOUTX2 => Bypass clk -> Foutx2/4 -> Foutx2/2 -> Foutx2/1.5 -> Foutx2 3h (R/W) = Reserved
2-0	DPLL_EN	R/W	4h	DPLL control. Upon Warm Reset, the PRCM DPLL control state machine updates this register to reflect MN Bypass mode. 0h (R/W) = Reserved 1h (R/W) = Reserved 2h (R/W) = Reserved 3h (R/W) = Reserved 4h (R/W) = Put the DPLL in MN Bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 5h (R/W) = Put the DPLL in Idle Bypass Low Power mode. 6h (R/W) = Put the DPLL in Idle Bypass Fast Relock mode. 7h (R/W) = Enables the DPLL in Lock mode

6.13.8.25 PRCM_CM_IDLEST_DPLL_CORE Register (Offset = 524h) [reset = 0h]

PRCM_CM_IDLEST_DPLL_CORE is shown in Figure 6-280 and described in Table 6-298.

Return to Summary Table.

This register allows monitoring the master clock activity. This register is read only and automatically updated. [warm reset insensitive]

Figure 6-280. PRCM_CM_IDLEST_DPLL_CORE Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>ST_MN_BYPASS</td></tr>
<tr><td colspan="7">R-0-0h</td><td>R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="7">RESERVED</td><td>ST_DPLL_CLK</td></tr>
<tr><td colspan="7">R-0-0h</td><td>R-0h</td></tr>
</table>

Table 6-298. PRCM_CM_IDLEST_DPLL_CORE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R-0	0h	
8	ST_MN_BYPASS	R	0h	DPLL MN_BYPASS status 0h (R) = DPLL is not in MN_Bypass 1h (R) = DPLL is in MN_Bypass
7-1	RESERVED	R-0	0h	
0	ST_DPLL_CLK	R	0h	DPLL clock activity 0h (R) = DPLL is either in bypass mode or in stop mode. 1h (R) = DPLL is LOCKED

6.13.8.26 PRCM_CM_CLKSEL_DPLL_CORE Register (Offset = 52Ch) [reset = 0h]

PRCM_CM_CLKSEL_DPLL_CORE is shown in Figure 6-281 and described in Table 6-299.

Return to Summary Table.

This register provides controls over the DPLL.

Figure 6-281. PRCM_CM_CLKSEL_DPLL_CORE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED	RESERVED				DPLL_MULT		
R-0-0h	R-0-0h				R/W-0h		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DPLL_MULT							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	DPLL_DIV						
R-0-0h	R/W-0h						

Table 6-299. PRCM_CM_CLKSEL_DPLL_CORE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R-0	0h	
22-19	RESERVED	R-0	0h	
18-8	DPLL_MULT	R/W	0h	DPLL multiplier factor (2 to 2047). This register is automatically cleared to 0 when the DPLL_EN field in the *CLKMODE_DPLL* register is set to select MN Bypass mode. (equal to input M of DPLL M=2 to 2047 => DPLL multiplies by M) 0h (R/W) = Reserved 1h (R/W) = Reserved
7	RESERVED	R-0	0h	
6-0	DPLL_DIV	R/W	0h	DPLL divider factor (0 to 127) (equal to input N of DPLL actual division factor is N+1).

6.13.8.27 PRCM_CM_DIV_M4_DPLL_CORE Register (Offset = 538h) [reset = 4h]

PRCM_CM_DIV_M4_DPLL_CORE is shown in Figure 6-282 and described in Table 6-300.

Return to Summary Table.

This register provides controls over the CLKOUT1 o/p of the HSDIVIDER.

Figure 6-282. PRCM_CM_DIV_M4_DPLL_CORE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED			HSDIVIDER_CLKOUT1_PWDN	RESERVED		ST_HSDIVIDER_CLKOUT1	HSDIVIDER_CLKOUT1_GATE_CTRL
R-0-0h			R/W-0h	R-0-0h		R-0h	R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED		HSDIVIDER_CLKOUT1_DIVCHACK	HSDIVIDER_CLKOUT1_DIV				
R-0-0h		R-0h	R/W-4h				

Table 6-300. PRCM_CM_DIV_M4_DPLL_CORE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R-0	0h	
12	HSDIVIDER_CLKOUT1_PWDN	R/W	0h	Power down for HSDIVIDER M4 divider and hence CLKOUT1 output 0h (R/W) = M4 divider active 1h (R/W) = M4 divider is powered down
11-10	RESERVED	R-0	0h	
9	ST_HSDIVIDER_CLKOUT1	R	0h	HSDIVIDER CLKOUT1 status 0h (R) = The clock output is gated 1h (R) = The clock output is enabled
8	HSDIVIDER_CLKOUT1_GATE_CTRL	R/W	0h	Control gating of HSDIVIDER CLKOUT1 0h (R/W) = Automatically gate this clock when there is no dependency for it 1h (R/W) = Force this clock to stay enabled even if there is no request
7-6	RESERVED	R-0	0h	
5	HSDIVIDER_CLKOUT1_DIVCHACK	R	0h	Toggle on this status bit after changing HSDIVIDER_CLKOUT1_DIV indicates that the change in divider value has taken effect
4-0	HSDIVIDER_CLKOUT1_DIV	R/W	4h	DPLL post-divider factor, M4, for internal clock generation. Divide values from 1 to 31. 0h (R/W) = Reserved

6.13.8.28 PRCM_CM_DIV_M5_DPLL_CORE Register (Offset = 53Ch) [reset = 4h]

PRCM_CM_DIV_M5_DPLL_CORE is shown in Figure 6-283 and described in Table 6-301.

Return to Summary Table.

This register provides controls over the CLKOUT2 o/p of the HSDIVIDER.

Figure 6-283. PRCM_CM_DIV_M5_DPLL_CORE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED			HSDIVIDER_CLKOUT2_PWDN	RESERVED		ST_HSDIVIDER_CLKOUT2	HSDIVIDER_CLKOUT2_GATE_CTRL
R-0-0h			R/W-0h	R-0-0h		R-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED		HSDIVIDER_CLKOUT2_DIVCHACK	HSDIVIDER_CLKOUT2_DIV				
R-0-0h		R-0h	R/W-4h				

Table 6-301. PRCM_CM_DIV_M5_DPLL_CORE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R-0	0h	
12	HSDIVIDER_CLKOUT2_PWDN	R/W	0h	PPower down for HSDIVIDER M5 divider and hence CLKOUT2 output 0h (R/W) = M5 divider is ACTIVE 1h (R/W) = M5 divider is powered down
11-10	RESERVED	R-0	0h	
9	ST_HSDIVIDER_CLKOUT2	R	0h	HSDIVIDER CLKOUT2 status 0h (R) = The clock output is gated 1h (R) = The clock output is enabled
8	HSDIVIDER_CLKOUT2_GATE_CTRL	R/W	0h	Control gating of HSDIVIDER CLKOUT2 0h (R/W) = Automatically gate this clock when there is no dependency for it 1h (R/W) = Force this clock to stay enabled even if there is no request
7-6	RESERVED	R-0	0h	
5	HSDIVIDER_CLKOUT2_DIVCHACK	R	0h	Toggle on this status bit after changing HSDIVIDER_CLKOUT2_DIV indicates that the change in divider value has taken effect
4-0	HSDIVIDER_CLKOUT2_DIV	R/W	4h	DPLL post-divider factor, M5, for internal clock generation. Divide values from 1 to 31. 0h (R/W) = Reserved

6.13.8.29 PRCM_CM_DIV_M6_DPLL_CORE Register (Offset = 540h) [reset = 4h]

PRCM_CM_DIV_M6_DPLL_CORE is shown in Figure 6-284 and described in Table 6-302.

Return to Summary Table.

This register provides controls over the CLKOUT3 o/p of the HSDIVIDER.

Figure 6-284. PRCM_CM_DIV_M6_DPLL_CORE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED			HSDIVIDER_CLKOUT3_PWDN	RESERVED		ST_HSDIVIDER_CLKOUT3	HSDIVIDER_CLKOUT3_GATE_CTRL
R-0-0h			R/W-0h	R-0-0h		R-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED		HSDIVIDER_CLKOUT3_DIVCHACK	HSDIVIDER_CLKOUT3_DIV				
R-0-0h		R-0h	R/W-4h				

Table 6-302. PRCM_CM_DIV_M6_DPLL_CORE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R-0	0h	
12	HSDIVIDER_CLKOUT3_PWDN	R/W	0h	Power down for HSDIVIDER M6 divider and hence CLKOUT3 output 0h (R/W) = M6 divider is ACTIVE 1h (R/W) = M6 divider is powered down
11-10	RESERVED	R-0	0h	
9	ST_HSDIVIDER_CLKOUT3	R	0h	HSDIVIDER CLKOUT3 status 0h (R) = The clock output is gated 1h (R) = The clock output is enabled
8	HSDIVIDER_CLKOUT3_GATE_CTRL	R/W	0h	Control gating of HSDIVIDER CLKOUT3 0h (R/W) = Automatically gate this clock when there is no dependency for it 1h (R/W) = Force this clock to stay enabled even if there is no request
7-6	RESERVED	R-0	0h	
5	HSDIVIDER_CLKOUT3_DIVCHACK	R	0h	Toggle on this status bit after changing HSDIVIDER_CLKOUT3_DIV indicates that the change in divider value has taken effect
4-0	HSDIVIDER_CLKOUT3_DIV	R/W	4h	DPLL post-divider factor, M6, for internal clock generation. Divide values from 1 to 31. 0h (R/W) = Reserved

6.13.8.30 PRCM_CM_SSC_DELTAMSTEP_DPLL_CORE Register (Offset = 548h) [reset = 0h]

PRCM_CM_SSC_DELTAMSTEP_DPLL_CORE is shown in Figure 6-285 and described in Table 6-303.

Return to Summary Table.

Control the DeltaMStep parameter for Spread Spectrum Clocking technique
DeltaMStep is split into fractional and integer part.
[warm reset insensitive]

Figure 6-285. PRCM_CM_SSC_DELTAMSTEP_DPLL_CORE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED				DELTAMSTEP_INTEGER		DELTAMSTEP_FRACTION	
R-0-0h				R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
DELTAMSTEP_FRACTION							
R/W-0h							
7	6	5	4	3	2	1	0
DELTAMSTEP_FRACTION							
R/W-0h							

Table 6-303. PRCM_CM_SSC_DELTAMSTEP_DPLL_CORE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	
19-18	DELTAMSTEP_INTEGER	R/W	0h	Integer part for DeltaM coefficient
17-0	DELTAMSTEP_FRACTION	R/W	0h	Fractional setting for DeltaMStep parameter

6.13.8.31 PRCM_CM_SSC_MODFREQDIV_DPLL_CORE Register (Offset = 54Ch) [reset = 0h]

PRCM_CM_SSC_MODFREQDIV_DPLL_CORE is shown in Figure 6-286 and described in Table 6-304.

Return to Summary Table.

Control the Modulation Frequency (Fm) for Spread Spectrum Clocking technique by defining it as a ratio of DPLL_REFCLK/4
Fm = [DPLL_REFCLK/4]/MODFREQDIV
MODFREQDIV = MODFREQDIV_MANTISSA * $2^{\text{MODFREQDIV\_EXPONENT}}$
[warm reset insensitive]

Figure 6-286. PRCM_CM_SSC_MODFREQDIV_DPLL_CORE Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="5">RESERVED</td><td colspan="3">MODFREQDIV_EXPONENT</td></tr>
<tr><td colspan="5">R-0-0h</td><td colspan="3">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>RESERVED</td><td colspan="7">MODFREQDIV_MANTISSA</td></tr>
<tr><td>R-0-0h</td><td colspan="7">R/W-0h</td></tr>
</table>

Table 6-304. PRCM_CM_SSC_MODFREQDIV_DPLL_CORE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	
10-8	MODFREQDIV_EXPONENT	R/W	0h	Set the Exponent component of MODFREQDIV factor
7	RESERVED	R-0	0h	
6-0	MODFREQDIV_MANTISSA	R/W	0h	Set the Mantissa component of MODFREQDIV factor

6.13.8.32 PRCM_CM_CLKMODE_DPLL_MPU Register (Offset = 560h) [reset = 4h]

PRCM_CM_CLKMODE_DPLL_MPU is shown in Figure 6-287 and described in Table 6-305.

Return to Summary Table.

This register allows controlling the DPLL modes.

Figure 6-287. PRCM_CM_CLKMODE_DPLL_MPU Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
DPLL_SSC_TYPE	DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	DPLL_REGM4XEN	DPLL_LPMODE_EN	DPLL_RELOCK_RAMP_EN	DPLL_DRIFTGUARD_EN
R/W-0h	R/W-0h	R-0h	R/W-0h	R-0-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
DPLL_RAMP_RATE			DPLL_RAMP_LEVEL		DPLL_EN		
R/W-0h			R/W-0h		R/W-4h		

Table 6-305. PRCM_CM_CLKMODE_DPLL_MPU Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	
15	DPLL_SSC_TYPE	R/W	0h	Select between Triangular and SquareWave Spread Spectrum Clocking 0h (R/W) = Triangular Spread Spectrum Clocking is selected 1h (R/W) = Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)
14	DPLL_SSC_DOWNSPREAD	R/W	0h	Control if only low frequency spread is required 0h (R/W) = When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 1h (R/W) = When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency
13	DPLL_SSC_ACK	R	0h	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0h (R) = SSC has been turned off on PLL o/ps 1h (R) = SSC has been turned on on PLL o/ps
12	DPLL_SSC_EN	R/W	0h	Enable or disable Spread Spectrum Clocking 0h (R/W) = SSC disabled 1h (R/W) = SSC enabled
11	DPLL_REGM4XEN	R-0	0h	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0h (R) = REGM4XEN mode of the DPLL is disabled
10	DPLL_LPMODE_EN	R/W	0h	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0h (R/W) = Low power mode of the DPLL is disabled 1h (R/W) = Low power mode of the DPLL is enabled
9	DPLL_RELOCK_RAMP_EN	R/W	0h	If enabled, the clock ramping feature is used applied during the lock process, as well as the relock process. If disabled, the clock ramping feature is used only during the first lock.

Table 6-305. PRCM_CM_CLKMODE_DPLL_MPU Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	DPLL_DRIFTGUARD_EN	R/W	0h	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0h (R/W) = DRIFTGUARD feature is disabled 1h (R/W) = DRIFTGUARD feature is enabled
7-5	DPLL_RAMP_RATE	R/W	0h	Selects the time in terms of DPLL REFCLKs spent at each stage of the clock ramping process 0h (R/W) = 2 REFCLKs 1h (R/W) = 4 REFCLKs 2h (R/W) = 8 REFCLKs 3h (R/W) = 16 REFCLKs 4h (R/W) = 32 REFCLKs 5h (R/W) = 64 REFCLKs 6h (R/W) = 128 REFCLKs 7h (R/W) = 512 REFCLKs
4-3	DPLL_RAMP_LEVEL	R/W	0h	The DPLL provides an output clock frequency ramping feature when switching from bypass clock to normal clock during lock and re-lock. The frequency ramping will happen in a maximum of 4 steps in frequency before the DPLL's frequency lock indicator is asserted. This register is used to enable/disable the DPLL ramping feature. If enabled, it is also used to select the algorithm used for clock ramping 0h (R/W) = CLKOUT => No ramping CLKOUTX2 => No ramping 1h (R/W) = CLKOUT => Bypass clk -> Fout/8 -> Fout/4 -> Fout/2 -> Fout CLKOUTX2 => Bypass clk -> Foutx2/8 -> Foutx2/4 -> Foutx2/2 -> Foutx2 2h (R/W) = CLKOUT => Bypass clk -> Fout/4 -> Fout/2 -> Fout/1.5 -> Fout CLKOUTX2 => Bypass clk -> Foutx2/4 -> Foutx2/2 -> Foutx2/1.5 -> Foutx2 3h (R/W) = Reserved
2-0	DPLL_EN	R/W	4h	DPLL control. Upon Warm Reset, the PRCM DPLL control state machine updates this register to reflect MN Bypass mode. 0h (R/W) = Reserved 1h (R/W) = Reserved 2h (R/W) = Reserved 3h (R/W) = Reserved 4h (R/W) = Put the DPLL in MN Bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 5h (R/W) = Put the DPLL in Idle Bypass Low Power mode. 6h (R/W) = Put the DPLL in Idle Bypass Fast Relock mode. 7h (R/W) = Enables the DPLL in Lock mode

6.13.8.33 PRCM_CM_IDLEST_DPLL_MPU Register (Offset = 564h) [reset = 0h]

PRCM_CM_IDLEST_DPLL_MPU is shown in Figure 6-288 and described in Table 6-306.

Return to Summary Table.

This register allows monitoring the master clock activity. This register is read only and automatically updated.[warm reset insensitive]

Figure 6-288. PRCM_CM_IDLEST_DPLL_MPU Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							

15	14	13	12	11	10	9	8
RESERVED							ST_MN_BYPASS
R-0-0h							R-0h

7	6	5	4	3	2	1	0
RESERVED							ST_DPLL_CLK
R-0-0h							R-0h

Table 6-306. PRCM_CM_IDLEST_DPLL_MPU Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R-0	0h	
8	ST_MN_BYPASS	R	0h	DPLL MN_BYPASS status 0h (R) = DPLL is not in MN_Bypass 1h (R) = DPLL is in MN_Bypass
7-1	RESERVED	R-0	0h	
0	ST_DPLL_CLK	R	0h	DPLL clock activity 0h (R) = DPLL is either in bypass mode or in stop mode. 1h (R) = DPLL is LOCKED

6.13.8.34 PRCM_CM_CLKSEL_DPLL_MPU Register (Offset = 56Ch) [reset = 05000000h]

PRCM_CM_CLKSEL_DPLL_MPU is shown in Figure 6-289 and described in Table 6-307.

Return to Summary Table.

This register provides controls over the DPLL.

Figure 6-289. PRCM_CM_CLKSEL_DPLL_MPU Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">DCC_COUNT_MAX</td></tr>
<tr><td colspan="8">R/W-5h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>DPLL_BYP_CLKSEL</td><td>DCC_EN</td><td colspan="3">RESERVED</td><td colspan="3">DPLL_MULT</td></tr>
<tr><td>R/W-0h</td><td>R/W-0h</td><td colspan="3">R-0-0h</td><td colspan="3">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">DPLL_MULT</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>RESERVED</td><td colspan="7">DPLL_DIV</td></tr>
<tr><td>R-0-0h</td><td colspan="7">R/W-0h</td></tr>
</table>

Table 6-307. PRCM_CM_CLKSEL_DPLL_MPU Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	DCC_COUNT_MAX	R/W	5h	The value "NbCycles" set in this field determines the duration of the clock ramp step during which output frequency is Fdpll/(2*M2). The duration is computed as 32 x NbCycles of WKUP-L4 clock cycles (SYSCLK/~26MHz). Duration should be > 1.5us to allow enough time for DCC to lock. This bit-field is only relevant when DCC_EN=1.
23	DPLL_BYP_CLKSEL	R/W	0h	Selects CLKINP or CLKINPULOW as Bypass Clock 0h (R/W) = Selects CLKINP Clock as BYPASS Clock 1h (R/W) = Selects CLKINPULOW as Bypass Clock
22	DCC_EN	R/W	0h	Enable or disable Duty Cycle Correction. Must be enabled only for frequency > 1GHz. When enabled, the CLKOUTHIF output of the DPLL is used after duty cycle correction instead of CLKOUT. M3 divider is hard-wired to 1 so the lock frequency Fdpll is directly provided to MPU. 0h (R/W) = DCC disabled 1h (R/W) = DCC enabled
21-19	RESERVED	R-0	0h	
18-8	DPLL_MULT	R/W	0h	DPLL multiplier factor (2 to 2047). This register is automatically cleared to 0 when the DPLL_EN field in the *CLKMODE_DPLL* register is set to select MN Bypass mode. (equal to input M of DPLL M=2 to 2047 => DPLL multiplies by M). 0h (R/W) = 0 : Reserved 1h (R/W) = 1 : Reserved
7	RESERVED	R-0	0h	
6-0	DPLL_DIV	R/W	0h	DPLL divider factor (0 to 127) (equal to input N of DPLL actual division factor is N+1).

6.13.8.35 PRCM_CM_DIV_M2_DPLL_MPU Register (Offset = 570h) [reset = 1h]

PRCM_CM_DIV_M2_DPLL_MPU is shown in Figure 6-290 and described in Table 6-308.

Return to Summary Table.

This register provides controls over the M2 divider of the DPLL.

Figure 6-290. PRCM_CM_DIV_M2_DPLL_MPU Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="6">RESERVED</td><td>ST_DPLL_CLKOUT</td><td>DPLL_CLKOUT_GATE_CTRL</td></tr>
<tr><td colspan="6">R-0-0h</td><td>R-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">RESERVED</td><td>DPLL_CLKOUT_DIVCHACK</td><td colspan="5">DPLL_CLKOUT_DIV</td></tr>
<tr><td colspan="2">R-0-0h</td><td>R-0h</td><td colspan="5">R/W-1h</td></tr>
</table>

Table 6-308. PRCM_CM_DIV_M2_DPLL_MPU Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R-0	0h	
9	ST_DPLL_CLKOUT	R	0h	DPLL CLKOUT status 0h (R) = The clock output is gated 1h (R) = The clock output is enabled
8	DPLL_CLKOUT_GATE_CTRL	R/W	0h	Control gating of DPLL CLKOUT 0h (R/W) = Automatically gate this clock when there is no dependency for it 1h (R/W) = Force this clock to stay enabled even if there is no request
7-6	RESERVED	R-0	0h	
5	DPLL_CLKOUT_DIVCHACK	R	0h	Toggle on this status bit after changing DPLL_CLKOUT_DIV indicates that the change in divider value has taken effect
4-0	DPLL_CLKOUT_DIV	R/W	1h	DPLL M2 post-divider factor (1 to 31). 0h (R/W) = Reserved

6.13.8.36 PRCM_CM_SSC_DELTAMSTEP_DPLL_MPU Register (Offset = 588h) [reset = 0h]

PRCM_CM_SSC_DELTAMSTEP_DPLL_MPU is shown in Figure 6-291 and described in Table 6-309.

Return to Summary Table.

Control the DeltaMStep parameter for Spread Spectrum Clocking technique
DeltaMStep is split into fractional and integer part.
[warm reset insensitive]

Figure 6-291. PRCM_CM_SSC_DELTAMSTEP_DPLL_MPU Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="2">DELTAMSTEP_INTEGER</td><td colspan="2">DELTAMSTEP_FRACTION</td></tr>
<tr><td colspan="4">R-0-0h</td><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">DELTAMSTEP_FRACTION</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">DELTAMSTEP_FRACTION</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 6-309. PRCM_CM_SSC_DELTAMSTEP_DPLL_MPU Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	
19-18	DELTAMSTEP_INTEGER	R/W	0h	Integer part for DeltaM coefficient
17-0	DELTAMSTEP_FRACTION	R/W	0h	Fractional setting for DeltaMStep parameter

6.13.8.37 PRCM_CM_SSC_MODFREQDIV_DPLL_MPU Register (Offset = 58Ch) [reset = 0h]

PRCM_CM_SSC_MODFREQDIV_DPLL_MPU is shown in Figure 6-292 and described in Table 6-310.

Return to Summary Table.

Control the Modulation Frequency (Fm) for Spread Spectrum Clocking technique by defining it as a ratio of DPLL_REFCLK/4
Fm = [DPLL_REFCLK/4]/MODFREQDIV
MODFREQDIV = MODFREQDIV_MANTISSA * $2^{\text{MODFREQDIV\_EXPONENT}}$
[warm reset insensitive]

Figure 6-292. PRCM_CM_SSC_MODFREQDIV_DPLL_MPU Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="5">RESERVED</td><td colspan="3">MODFREQDIV_EXPONENT</td></tr>
<tr><td colspan="5">R-0-0h</td><td colspan="3">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>RESERVED</td><td colspan="7">MODFREQDIV_MANTISSA</td></tr>
<tr><td>R-0-0h</td><td colspan="7">R/W-0h</td></tr>
</table>

Table 6-310. PRCM_CM_SSC_MODFREQDIV_DPLL_MPU Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	
10-8	MODFREQDIV_EXPONENT	R/W	0h	Set the Exponent component of MODFREQDIV factor
7	RESERVED	R-0	0h	
6-0	MODFREQDIV_MANTISSA	R/W	0h	Set the Mantissa component of MODFREQDIV factor

6.13.8.38 PRCM_CM_CLKMODE_DPLL_DDR Register (Offset = 5A0h) [reset = 4h]

PRCM_CM_CLKMODE_DPLL_DDR is shown in Figure 6-293 and described in Table 6-311.

Return to Summary Table.

This register allows controlling the DPLL modes.

Figure 6-293. PRCM_CM_CLKMODE_DPLL_DDR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>DPLL_SSC_TYPE</td><td>DPLL_SSC_DOWNSPREAD</td><td>DPLL_SSC_ACK</td><td>DPLL_SSC_EN</td><td>DPLL_REGM4XEN</td><td>DPLL_LPMODE_EN</td><td>DPLL_RELOCK_RAMP_EN</td><td>DPLL_DRIFTGUARD_EN</td></tr>
<tr><td>R/W-0h</td><td>R/W-0h</td><td>R-0h</td><td>R/W-0h</td><td>R-0-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">DPLL_RAMP_RATE</td><td colspan="2">DPLL_RAMP_LEVEL</td><td colspan="3">DPLL_EN</td></tr>
<tr><td colspan="3">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="3">R/W-4h</td></tr>
</table>

Table 6-311. PRCM_CM_CLKMODE_DPLL_DDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	
15	DPLL_SSC_TYPE	R/W	0h	Select between Triangular and SquareWave Spread Spectrum Clocking 0h (R/W) = Triangular Spread Spectrum Clocking is selected 1h (R/W) = Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)
14	DPLL_SSC_DOWNSPREAD	R/W	0h	Control if only low frequency spread is required 0h (R/W) = When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 1h (R/W) = When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency
13	DPLL_SSC_ACK	R	0h	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0h (R) = SSC has been turned off on PLL o/ps 1h (R) = SSC has been turned on on PLL o/ps
12	DPLL_SSC_EN	R/W	0h	Enable or disable Spread Spectrum Clocking 0h (R/W) = SSC disabled 1h (R/W) = SSC enabled
11	DPLL_REGM4XEN	R-0	0h	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0h (R) = REGM4XEN mode of the DPLL is disabled
10	DPLL_LPMODE_EN	R/W	0h	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0h (R/W) = Low power mode of the DPLL is disabled 1h (R/W) = Low power mode of the DPLL is enabled
9	DPLL_RELOCK_RAMP_EN	R/W	0h	If enabled, the clock ramping feature is used applied during the lock process, as well as the relock process. If disabled, the clock ramping feature is used only during the first lock.

Table 6-311. PRCM_CM_CLKMODE_DPLL_DDR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	DPLL_DRIFTGUARD_EN	R/W	0h	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0h (R/W) = DRIFTGUARD feature is disabled 1h (R/W) = DRIFTGUARD feature is enabled
7-5	DPLL_RAMP_RATE	R/W	0h	Selects the time in terms of DPLL REFCLKs spent at each stage of the clock ramping process 0h (R/W) = 2 REFCLKs 1h (R/W) = 4 REFCLKs 2h (R/W) = 8 REFCLKs 3h (R/W) = 16 REFCLKs 4h (R/W) = 32 REFCLKs 5h (R/W) = 64 REFCLKs 6h (R/W) = 128 REFCLKs 7h (R/W) = 512 REFCLKs
4-3	DPLL_RAMP_LEVEL	R/W	0h	The DPLL provides an output clock frequency ramping feature when switching from bypass clock to normal clock during lock and re-lock. The frequency ramping will happen in a maximum of 4 steps in frequency before the DPLL's frequency lock indicator is asserted. This register is used to enable/disable the DPLL ramping feature. If enabled, it is also used to select the algorithm used for clock ramping 0h (R/W) = CLKOUT => No ramping CLKOUTX2 => No ramping 1h (R/W) = CLKOUT => Bypass clk -> Fout/8 -> Fout/4 -> Fout/2 -> Fout CLKOUTX2 => Bypass clk -> Foutx2/8 -> Foutx2/4 -> Foutx2/2 -> Foutx2 2h (R/W) = CLKOUT => Bypass clk -> Fout/4 -> Fout/2 -> Fout/1.5 -> Fout CLKOUTX2 => Bypass clk -> Foutx2/4 -> Foutx2/2 -> Foutx2/1.5 -> Foutx2 3h (R/W) = Reserved
2-0	DPLL_EN	R/W	4h	DPLL control. Upon Warm Reset, the PRCM DPLL control state machine updates this register to reflect MN Bypass mode. 0h (R/W) = Reserved 1h (R/W) = Reserved 2h (R/W) = Reserved 3h (R/W) = Reserved 4h (R/W) = Put the DPLL in MN Bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 5h (R/W) = Put the DPLL in Idle Bypass Low Power mode. 6h (R/W) = Put the DPLL in Idle Bypass Fast Relock mode. 7h (R/W) = Enables the DPLL in Lock mode

6.13.8.39 PRCM_CM_IDLEST_DPLL_DDR Register (Offset = 5A4h) [reset = 0h]

PRCM_CM_IDLEST_DPLL_DDR is shown in Figure 6-294 and described in Table 6-312.

Return to Summary Table.

This register allows monitoring the master clock activity. This register is read only and automatically updated. [warm reset insensitive]

Figure 6-294. PRCM_CM_IDLEST_DPLL_DDR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							ST_MN_BYPASS
R-0-0h							R-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							ST_DPLL_CLK
R-0-0h							R-0h

Table 6-312. PRCM_CM_IDLEST_DPLL_DDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R-0	0h	
8	ST_MN_BYPASS	R	0h	DPLL MN_BYPASS status 0h (R) = DPLL is not in MN_Bypass 1h (R) = DPLL is in MN_Bypass
7-1	RESERVED	R-0	0h	
0	ST_DPLL_CLK	R	0h	DPLL clock activity 0h (R) = DPLL is either in bypass mode or in stop mode. 1h (R) = DPLL is LOCKED

6.13.8.40 PRCM_CM_CLKSEL_DPLL_DDR Register (Offset = 5ACh) [reset = 0h]

PRCM_CM_CLKSEL_DPLL_DDR is shown in Figure 6-295 and described in Table 6-313.

Return to Summary Table.

This register provides controls over the DPLL.

Figure 6-295. PRCM_CM_CLKSEL_DPLL_DDR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
DPLL_BYP_CLKSEL	RESERVED				DPLL_MULT		
R/W-0h	R-0-0h				R/W-0h		
15	14	13	12	11	10	9	8
DPLL_MULT							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED	DPLL_DIV						
R-0-0h	R/W-0h						

Table 6-313. PRCM_CM_CLKSEL_DPLL_DDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R-0	0h	
23	DPLL_BYP_CLKSEL	R/W	0h	Select CLKINP orr CLKINPULOW as bypass clock 0h (R/W) = Selects CLKINP Clock as BYPASS Clock 1h (R/W) = Selects CLKINPULOW as Bypass Clock
22-19	RESERVED	R-0	0h	
18-8	DPLL_MULT	R/W	0h	DPLL multiplier factor (2 to 2047). This register is automatically cleared to 0 when the DPLL_EN field in the *CLKMODE_DPLL* register is set to select MN Bypass mode. (equal to input M of DPLL M=2 to 2047 => DPLL multiplies by M). 0h (R/W) = 0 : Reserved 1h (R/W) = 1 : Reserved
7	RESERVED	R-0	0h	
6-0	DPLL_DIV	R/W	0h	DPLL divider factor (0 to 127) (equal to input N of DPLL actual division factor is N+1).

6.13.8.41 PRCM_CM_DIV_M2_DPLL_DDR Register (Offset = 5B0h) [reset = 1h]

PRCM_CM_DIV_M2_DPLL_DDR is shown in Figure 6-296 and described in Table 6-314.

Return to Summary Table.

This register provides controls over the M2 divider of the DPLL.

Figure 6-296. PRCM_CM_DIV_M2_DPLL_DDR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="6">RESERVED</td><td>ST_DPLL_CLKOUT</td><td>DPLL_CLKOUT_GATE_CTRL</td></tr>
<tr><td colspan="6">R-0-0h</td><td>R-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">RESERVED</td><td>DPLL_CLKOUT_DIVCHACK</td><td colspan="5">DPLL_CLKOUT_DIV</td></tr>
<tr><td colspan="2">R-0-0h</td><td>R-0h</td><td colspan="5">R/W-1h</td></tr>
</table>

Table 6-314. PRCM_CM_DIV_M2_DPLL_DDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R-0	0h	
9	ST_DPLL_CLKOUT	R	0h	DPLL CLKOUT status 0h (R) = The clock output is gated 1h (R) = The clock output is enabled
8	DPLL_CLKOUT_GATE_CTRL	R/W	0h	Control gating of DPLL CLKOUT 0h (R/W) = Automatically gate this clock when there is no dependency for it 1h (R/W) = Force this clock to stay enabled even if there is no request
7-6	RESERVED	R-0	0h	
5	DPLL_CLKOUT_DIVCHACK	R	0h	Toggle on this status bit after changing DPLL_CLKOUT_DIV indicates that the change in divider value has taken effect
4-0	DPLL_CLKOUT_DIV	R/W	1h	DPLL M2 post-divider factor (1 to 31). 0h (R/W) = Reserved

6.13.8.42 PRCM_CM_DIV_M4_DPLL_DDR Register (Offset = 5B8h) [reset = 4h]

PRCM_CM_DIV_M4_DPLL_DDR is shown in Figure 6-297 and described in Table 6-315.

Return to Summary Table.

This register provides controls over the CLKOUT1 o/p of the DDR PLL HSDIVIDER.

Figure 6-297. PRCM_CM_DIV_M4_DPLL_DDR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							

15	14	13	12	11	10	9	8
RESERVED			HSDIVIDER_CLKOUT1_PWDN	RESERVED		ST_HSDIVIDER_CLKOUT1	HSDIVIDER_CLKOUT1_GATE_CTRL
R-0-0h			R/W-0h	R-0-0h		R-0h	R/W-0h

7	6	5	4	3	2	1	0
RESERVED		HSDIVIDER_CLKOUT1_DIVCHACK	HSDIVIDER_CLKOUT1_DIV				
R-0-0h		R-0h	R/W-4h				

Table 6-315. PRCM_CM_DIV_M4_DPLL_DDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R-0	0h	
12	HSDIVIDER_CLKOUT1_PWDN	R/W	0h	Power down for HSDIVIDER M4 divider and hence CLKOUT1 output 0h (R/W) = M4 divider active 1h (R/W) = M4 divider is powered down
11-10	RESERVED	R-0	0h	
9	ST_HSDIVIDER_CLKOUT1	R	0h	HSDIVIDER CLKOUT1 status 0h (R) = The clock output is gated 1h (R) = The clock output is enabled
8	HSDIVIDER_CLKOUT1_GATE_CTRL	R/W	0h	Control gating of HSDIVIDER CLKOUT1 0h (R/W) = Automatically gate this clock when there is no dependency for it 1h (R/W) = Force this clock to stay enabled even if there is no request
7-6	RESERVED	R-0	0h	
5	HSDIVIDER_CLKOUT1_DIVCHACK	R	0h	Toggle on this status bit after changing HSDIVIDER_CLKOUT1_DIV indicates that the change in divider value has taken effect
4-0	HSDIVIDER_CLKOUT1_DIV	R/W	4h	DPLL post-divider factor, M4, for internal clock generation. Divide values from 1 to 31. 0h (R/W) = Reserved

6.13.8.43 PRCM_CM_SSC_DELTAMSTEP_DPLL_DDR Register (Offset = 5C8h) [reset = 0h]

PRCM_CM_SSC_DELTAMSTEP_DPLL_DDR is shown in Figure 6-298 and described in Table 6-316.

Return to Summary Table.

Control the DeltaMStep parameter for Spread Spectrum Clocking technique
DeltaMStep is split into fractional and integer part.
[warm reset insensitive]

Figure 6-298. PRCM_CM_SSC_DELTAMSTEP_DPLL_DDR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				DELTAMSTEP_INTEGER		DELTAMSTEP_FRACTION	
R-0-0h				R/W-0h		R/W-0h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DELTAMSTEP_FRACTION							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DELTAMSTEP_FRACTION							
R/W-0h							

Table 6-316. PRCM_CM_SSC_DELTAMSTEP_DPLL_DDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	
19-18	DELTAMSTEP_INTEGER	R/W	0h	Integer part for DeltaM coefficient
17-0	DELTAMSTEP_FRACTION	R/W	0h	Fractional setting for DeltaMStep parameter

6.13.8.44 PRCM_CM_SSC_MODFREQDIV_DPLL_DDR Register (Offset = 5CCh) [reset = 0h]

PRCM_CM_SSC_MODFREQDIV_DPLL_DDR is shown in Figure 6-299 and described in Table 6-317.

Return to Summary Table.

Control the Modulation Frequency (Fm) for Spread Spectrum Clocking technique by defining it as a ratio of DPLL_REFCLK/4
Fm = [DPLL_REFCLK/4]/MODFREQDIV
$\text{MODFREQDIV} = \text{MODFREQDIV\_MANTISSA} * 2^{\text{MODFREQDIV\_EXPONENT}}$
[warm reset insensitive]

Figure 6-299. PRCM_CM_SSC_MODFREQDIV_DPLL_DDR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="5">RESERVED</td><td colspan="3">MODFREQDIV_EXPONENT</td></tr>
<tr><td colspan="5">R-0-0h</td><td colspan="3">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>RESERVED</td><td colspan="7">MODFREQDIV_MANTISSA</td></tr>
<tr><td>R-0-0h</td><td colspan="7">R/W-0h</td></tr>
</table>

Table 6-317. PRCM_CM_SSC_MODFREQDIV_DPLL_DDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	
10-8	MODFREQDIV_EXPONENT	R/W	0h	Set the Exponent component of MODFREQDIV factor
7	RESERVED	R-0	0h	
6-0	MODFREQDIV_MANTISSA	R/W	0h	Set the Mantissa component of MODFREQDIV factor

6.13.8.45 PRCM_CM_CLKMODE_DPLL_PER Register (Offset = 5E0h) [reset = 4h]

PRCM_CM_CLKMODE_DPLL_PER is shown in Figure 6-300 and described in Table 6-318.

Return to Summary Table.

This register allows controlling the DPLL modes.

Figure 6-300. PRCM_CM_CLKMODE_DPLL_PER Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
DPLL_SSC_TYPE	DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	RESERVED			
R/W-0h	R/W-0h	R-0h	R/W-0h	R-0-0h			
7	6	5	4	3	2	1	0
RESERVED					DPLL_EN		
R-0-0h					R/W-4h		

Table 6-318. PRCM_CM_CLKMODE_DPLL_PER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	
15	DPLL_SSC_TYPE	R/W	0h	Select between Triangular and SquareWave Spread Spectrum Clocking 0h (R/W) = Triangular Spread Spectrum Clocking is selected 1h (R/W) = Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)
14	DPLL_SSC_DOWNSPREAD	R/W	0h	Control if only low frequency spread is required 0h (R/W) = When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 1h (R/W) = When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency
13	DPLL_SSC_ACK	R	0h	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0h (R) = SSC has been turned off on PLL o/ps 1h (R) = SSC has been turned on on PLL o/ps
12	DPLL_SSC_EN	R/W	0h	Enable or disable Spread Spectrum Clocking 0h (R/W) = SSC disabled 1h (R/W) = SSC enabled
11-3	RESERVED	R-0	0h	
2-0	DPLL_EN	R/W	4h	DPLL control. Upon Warm Reset, the PRCM DPLL control state machine updates this register to reflect DPLL Low Power Stop mode. 0h (R/W) = Reserved 1h (R/W) = Reserved 2h (R/W) = Reserved2 3h (R/W) = Reserved 4h (R/W) = Put the DPLL in MN Bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 5h (R/W) = Put the DPLL in Idle Bypass Low Power mode. 6h (R/W) = Reserved 7h (R/W) = Enables the DPLL in Lock mode

6.13.8.46 PRCM_CM_IDLEST_DPLL_PER Register (Offset = 5E4h) [reset = 0h]

PRCM_CM_IDLEST_DPLL_PER is shown in Figure 6-301 and described in Table 6-319.

Return to Summary Table.

This register allows monitoring the master clock activity. This register is read only and automatically updated. [warm reset insensitive]

Figure 6-301. PRCM_CM_IDLEST_DPLL_PER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>ST_MN_BYPASS</td></tr>
<tr><td colspan="7">R-0-0h</td><td>R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="7">RESERVED</td><td>ST_DPLL_CLK</td></tr>
<tr><td colspan="7">R-0-0h</td><td>R-0h</td></tr>
</table>

Table 6-319. PRCM_CM_IDLEST_DPLL_PER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R-0	0h	
8	ST_MN_BYPASS	R	0h	DPLL MN_BYPASS status 0h (R) = DPLL is not in MN_Bypass 1h (R) = DPLL is in MN_Bypass
7-1	RESERVED	R-0	0h	
0	ST_DPLL_CLK	R	0h	DPLL clock activity 0h (R) = DPLL is either in bypass mode or in stop mode. 1h (R) = DPLL is LOCKED

6.13.8.47 PRCM_CM_CLKSEL_DPLL_PER Register (Offset = 5ECh) [reset = 0h]

PRCM_CM_CLKSEL_DPLL_PER is shown in Figure 6-302 and described in Table 6-320.

Return to Summary Table.

This register provides controls over the DPLL.

Figure 6-302. PRCM_CM_CLKSEL_DPLL_PER Register

31	30	29	28	27	26	25	24
DPLL_SD_DIV							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED	RESERVED			DPLL_MULT			
R-0-0h	R-0-0h			R/W-0h			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DPLL_MULT							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DPLL_DIV							
R/W-0h							

Table 6-320. PRCM_CM_CLKSEL_DPLL_PER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	DPLL_SD_DIV	R/W	0h	Sigma-Delta divider select (2-255). This factor must be set by s/w to ensure optimum jitter performance. DPLL_SD_DIV = CEILING ([DPLL_MULT/(DPLL_DIV+1)] * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. 0h (R/W) = Reserved 1h (R/W) = Reserved
23	RESERVED	R-0	0h	
22-20	RESERVED	R-0	0h	
19-8	DPLL_MULT	R/W	0h	DPLL multiplier factor (2 to 4095). This register is automatically cleared to 0 when the DPLL_EN field in the *CLKMODE_DPLL* register is set to select MN Bypass mode. (equal to input M of DPLL M=2 to 4095 => DPLL multiplies by M). 0h (R/W) = 0 : Reserved 1h (R/W) = 1 : Reserved
7-0	DPLL_DIV	R/W	0h	DPLL divider factor (0 to 255) (equal to input N of DPLL actual division factor is N+1).

6.13.8.48 PRCM_CM_DIV_M2_DPLL_PER Register (Offset = 5F0h) [reset = 1h]

PRCM_CM_DIV_M2_DPLL_PER is shown in Figure 6-303 and described in Table 6-321.

Return to Summary Table.

This register provides controls over the M2 divider of the DPLL.

Figure 6-303. PRCM_CM_DIV_M2_DPLL_PER Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED						ST_DPLL_CLKOUT	DPLL_CLKOUT_GATE_CTRL
R-0-0h						R-0h	R/W-0h
7	6	5	4	3	2	1	0
DPLL_CLKOUT_DIVCHACK	DPLL_CLKOUT_DIV						
R-0h	R/W-1h						

Table 6-321. PRCM_CM_DIV_M2_DPLL_PER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R-0	0h	
9	ST_DPLL_CLKOUT	R	0h	DPLL CLKOUT status 0h (R) = The clock output is gated 1h (R) = The clock output is enabled
8	DPLL_CLKOUT_GATE_CTRL	R/W	0h	Control gating of DPLL CLKOUT 0h (R/W) = Automatically gate this clock when there is no dependency for it 1h (R/W) = Force this clock to stay enabled even if there is no request
7	DPLL_CLKOUT_DIVCHACK	R	0h	Toggle on this status bit after changing DPLL_CLKOUT_DIV indicates that the change in divider value has taken effect
6-0	DPLL_CLKOUT_DIV	R/W	1h	DPLL M2 post-divider factor (1 to 31). 0h (R/W) = Reserved

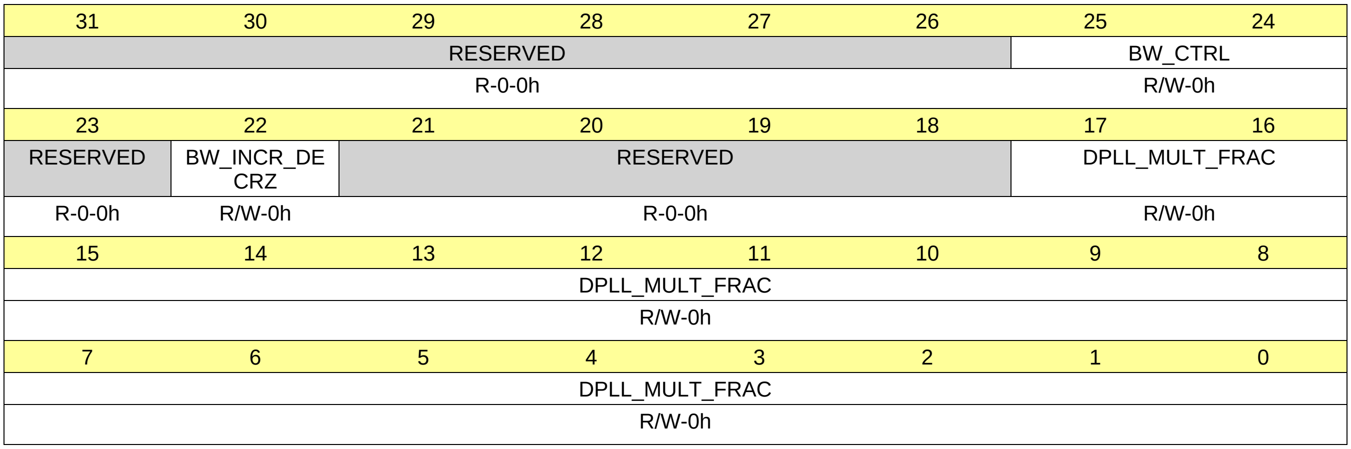
6.13.8.49 PRCM_CM_CLKSEL2_DPLL_PER Register (Offset = 604h) [reset = 0h]

PRCM_CM_CLKSEL2_DPLL_PER is shown in Figure 6-304 and described in Table 6-322.

Return to Summary Table.

This register provides DPLL fractional multiplier factor control and BandWidth Control for PER DPLL.

Figure 6-304. PRCM_CM_CLKSEL2_DPLL_PER Register



<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">BW_CTRL</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>RESERVED</td><td>BW_INCR_DECRZ</td><td colspan="4">RESERVED</td><td colspan="2">DPLL_MULT_FRAC</td></tr>
<tr><td>R-0-0h</td><td>R/W-0h</td><td colspan="4">R-0-0h</td><td colspan="2">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">DPLL_MULT_FRAC</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">DPLL_MULT_FRAC</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 6-322. PRCM_CM_CLKSEL2_DPLL_PER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	RESERVED
25-24	BW_CTRL	R/W	0h	Register to change the BandWidth of the PLL. This field should be used only for DEBUG purpose. 0h (R/W) = BW mode is 1x for BW_INCR_DECRZ=0/1 1h (R/W) = BW mode is 0.5x for BW_INCR_DECRZ=0 and BW mode is 2x for BW_INCR_DECRZ=1 2h (R/W) = BW mode is 0.25x for BW_INCR_DECRZ=0 and is not supported for BW_INCR_DECRZ=1 3h (R/W) = BW mode is 0.125x for BW_INCR_DECRZ=0 and is not supported for BW_INCR_DECRZ=1
23	RESERVED	R-0	0h	RESERVED
22	BW_INCR_DECRZ	R/W	0h	Register for controlling BandWidth increase/decrease. This field should be used only for DEBUG purpose. 0h (R/W) = Decreases the BandWidth 1h (R/W) = Increases the BandWidth
21-18	RESERVED	R-0	0h	RESERVED
17-0	DPLL_MULT_FRAC	R/W	0h	DPLL fractional multiplier factor. Setting to 0 keeps DPLL in integer mode. This field should be used only for DEBUG purpose.

6.13.8.50 PRCM_CM_SSC_DELTAMSTEP_DPLL_PER Register (Offset = 608h) [reset = 0h]

PRCM_CM_SSC_DELTAMSTEP_DPLL_PER is shown in Figure 6-305 and described in Table 6-323.

Return to Summary Table.

Control the DeltaMStep parameter for Spread Spectrum Clocking technique
DeltaMStep is split into fractional and integer part.
[warm reset insensitive]

Figure 6-305. PRCM_CM_SSC_DELTAMSTEP_DPLL_PER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="2">DELTAMSTEP_INTEGER</td><td colspan="2">DELTAMSTEP_FRACTION</td></tr>
<tr><td colspan="4">R-0-0h</td><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">DELTAMSTEP_FRACTION</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">DELTAMSTEP_FRACTION</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 6-323. PRCM_CM_SSC_DELTAMSTEP_DPLL_PER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	
19-18	DELTAMSTEP_INTEGER	R/W	0h	Integer part for DeltaM coefficient
17-0	DELTAMSTEP_FRACTION	R/W	0h	Fractional setting for DeltaMStep parameter

6.13.8.51 PRCM_CM_SSC_MODFREQDIV_DPLL_PER Register (Offset = 60Ch) [reset = 0h]

PRCM_CM_SSC_MODFREQDIV_DPLL_PER is shown in Figure 6-306 and described in Table 6-324.

Return to Summary Table.

Control the Modulation Frequency (Fm) for Spread Spectrum Clocking technique by defining it as a ratio of DPLL_REFCLK/4
Fm = [DPLL_REFCLK/4]/MODFREQDIV
MODFREQDIV = MODFREQDIV_MANTISSA * $2^{\text{MODFREQDIV\_EXPONENT}}$
[warm reset insensitive]

Figure 6-306. PRCM_CM_SSC_MODFREQDIV_DPLL_PER Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED					MODFREQDIV_EXPONENT		
R-0-0h					R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	MODFREQDIV_MANTISSA						
R-0-0h	R/W-0h						

Table 6-324. PRCM_CM_SSC_MODFREQDIV_DPLL_PER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	
10-8	MODFREQDIV_EXPONENT	R/W	0h	Set the Exponent component of MODFREQDIV factor
7	RESERVED	R-0	0h	
6-0	MODFREQDIV_MANTISSA	R/W	0h	Set the Mantissa component of MODFREQDIV factor

6.13.8.52 PRCM_CM_CLKDCOLDO_DPLL_PER Register (Offset = 614h) [reset = 0h]

PRCM_CM_CLKDCOLDO_DPLL_PER is shown in Figure 6-307 and described in Table 6-325.

Return to Summary Table.

This register provides controls over the CLKDCOLDO output of the DPLL.

Figure 6-307. PRCM_CM_CLKDCOLDO_DPLL_PER Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED			DPLL_CLKDCOLDO_PWDN	RESERVED		ST_DPLL_CLKDCOLDO	DPLL_CLKDCOLDO_GATE_CTRL
R-0-0h			R/W-0h	R-0-0h		R-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED							
R-0-0h							

Table 6-325. PRCM_CM_CLKDCOLDO_DPLL_PER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R-0	0h	
12	DPLL_CLKDCOLDO_PWDN	R/W	0h	Software control for PWRDN on DCOLDO O/P 0h (R/W) = DCOLDO O/P is ACTIVE 1h (R/W) = DCOLDO O/P is PWRDN
11-10	RESERVED	R-0	0h	
9	ST_DPLL_CLKDCOLDO	R	0h	DPLL CLKDCOLDO status 0h (R) = The clock output is gated 1h (R) = The clock output is enabled
8	DPLL_CLKDCOLDO_GATE_CTRL	R/W	0h	Control gating of DPLL CLKDCOLDO 0h (R/W) = Automatically gate this clock when there is no dependency for it 1h (R/W) = Force this clock to stay enabled even if there is no request
7-0	RESERVED	R-0	0h	

6.13.8.53 PRCM_CM_CLKMODE_DPLL_DISP Register (Offset = 620h) [reset = 4h]

PRCM_CM_CLKMODE_DPLL_DISP is shown in Figure 6-308 and described in Table 6-326.

Return to Summary Table.

This register allows controlling the DPLL modes.

Figure 6-308. PRCM_CM_CLKMODE_DPLL_DISP Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DPLL_SSC_TYPE	DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	DPLL_REGM4XEN	DPLL_LPMODE_EN	DPLL_RELOCK_RAMP_EN	DPLL_DRIFTGUARD_EN
R/W-0h	R/W-0h	R-0h	R/W-0h	R-0-0h	R/W-0h	R/W-0h	R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DPLL_RAMP_RATE			DPLL_RAMP_LEVEL		DPLL_EN		
R/W-0h			R/W-0h		R/W-4h		

Table 6-326. PRCM_CM_CLKMODE_DPLL_DISP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	
15	DPLL_SSC_TYPE	R/W	0h	Select between Triangular and SquareWave Spread Spectrum Clocking 0h (R/W) = Triangular Spread Spectrum Clocking is selected 1h (R/W) = Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)
14	DPLL_SSC_DOWNSPREAD	R/W	0h	Control if only low frequency spread is required 0h (R/W) = When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 1h (R/W) = When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency
13	DPLL_SSC_ACK	R	0h	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0h (R) = SSC has been turned off on PLL o/ps 1h (R) = SSC has been turned on on PLL o/ps
12	DPLL_SSC_EN	R/W	0h	Enable or disable Spread Spectrum Clocking 0h (R/W) = SSC disabled 1h (R/W) = SSC enabled
11	DPLL_REGM4XEN	R-0	0h	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0h (R) = REGM4XEN mode of the DPLL is disabled
10	DPLL_LPMODE_EN	R/W	0h	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0h (R/W) = Low power mode of the DPLL is disabled 1h (R/W) = Low power mode of the DPLL is enabled
9	DPLL_RELOCK_RAMP_EN	R/W	0h	If enabled, the clock ramping feature is used applied during the lock process, as well as the relock process. If disabled, the clock ramping feature is used only during the first lock.

Table 6-326. PRCM_CM_CLKMODE_DPLL_DISP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	DPLL_DRIFTGUARD_EN	R/W	0h	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0h (R/W) = DRIFTGUARD feature is disabled 1h (R/W) = DRIFTGUARD feature is enabled
7-5	DPLL_RAMP_RATE	R/W	0h	Selects the time in terms of DPLL REFCLKs spent at each stage of the clock ramping process 0h (R/W) = 2 REFCLKs 1h (R/W) = 4 REFCLKs 2h (R/W) = 8 REFCLKs 3h (R/W) = 16 REFCLKs 4h (R/W) = 32 REFCLKs 5h (R/W) = 64 REFCLKs 6h (R/W) = 128 REFCLKs 7h (R/W) = 512 REFCLKs
4-3	DPLL_RAMP_LEVEL	R/W	0h	The DPLL provides an output clock frequency ramping feature when switching from bypass clock to normal clock during lock and re-lock. The frequency ramping will happen in a maximum of 4 steps in frequency before the DPLL's frequency lock indicator is asserted. This register is used to enable/disable the DPLL ramping feature. If enabled, it is also used to select the algorithm used for clock ramping 0h (R/W) = CLKOUT => No ramping CLKOUTX2 => No ramping 1h (R/W) = CLKOUT => Bypass clk -> Fout/8 -> Fout/4 -> Fout/2 -> Fout CLKOUTX2 => Bypass clk -> Foutx2/8 -> Foutx2/4 -> Foutx2/2 -> Foutx2 2h (R/W) = CLKOUT => Bypass clk -> Fout/4 -> Fout/2 -> Fout/1.5 -> Fout CLKOUTX2 => Bypass clk -> Foutx2/4 -> Foutx2/2 -> Foutx2/1.5 -> Foutx2 3h (R/W) = Reserved
2-0	DPLL_EN	R/W	4h	DPLL control. Upon Warm Reset, the PRCM DPLL control state machine updates this register to reflect MN Bypass mode. 0h (R/W) = Reserved 1h (R/W) = Reserved 2h (R/W) = Reserved 3h (R/W) = Reserved 4h (R/W) = Put the DPLL in MN Bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 5h (R/W) = Put the DPLL in Idle Bypass Low Power mode. 6h (R/W) = Put the DPLL in Idle Bypass Fast Relock mode. 7h (R/W) = Enables the DPLL in Lock mode

6.13.8.54 PRCM_CM_IDLEST_DPLL_DISP Register (Offset = 624h) [reset = 0h]

PRCM_CM_IDLEST_DPLL_DISP is shown in Figure 6-309 and described in Table 6-327.

Return to Summary Table.

This register allows monitoring the master clock activity. This register is read only and automatically updated. [warm reset insensitive]

Figure 6-309. PRCM_CM_IDLEST_DPLL_DISP Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							ST_MN_BYPASS
R-0-0h							R-0h
7	6	5	4	3	2	1	0
RESERVED							ST_DPLL_CLK
R-0-0h							R-0h

Table 6-327. PRCM_CM_IDLEST_DPLL_DISP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R-0	0h	
8	ST_MN_BYPASS	R	0h	DPLL MN_BYPASS status 0h (R) = DPLL is not in MN_Bypass 1h (R) = DPLL is in MN_Bypass
7-1	RESERVED	R-0	0h	
0	ST_DPLL_CLK	R	0h	DPLL clock activity 0h (R) = DPLL is either in bypass mode or in stop mode. 1h (R) = DPLL is LOCKED

6.13.8.55 PRCM_CM_CLKSEL_DPLL_DISP Register (Offset = 62Ch) [reset = 0h]

PRCM_CM_CLKSEL_DPLL_DISP is shown in Figure 6-310 and described in Table 6-328.

Return to Summary Table.

This register provides controls over the DPLL.

Figure 6-310. PRCM_CM_CLKSEL_DPLL_DISP Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DPLL_BYP_CLKSEL	RESERVED				DPLL_MULT		
R/W-0h	R-0-0h				R/W-0h		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DPLL_MULT							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	DPLL_DIV						
R-0-0h	R/W-0h						

Table 6-328. PRCM_CM_CLKSEL_DPLL_DISP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R-0	0h	
23	DPLL_BYP_CLKSEL	R/W	0h	Select CLKINP or CLKINPULOW as bypass clock 0h (R/W) = Selects CLKINP Clock as BYPASS Clock 1h (R/W) = Selects CLKINPULOW as Bypass Clock
22-19	RESERVED	R-0	0h	
18-8	DPLL_MULT	R/W	0h	DPLL multiplier factor (2 to 2047). This register is automatically cleared to 0 when the DPLL_EN field in the *CLKMODE_DPLL* register is set to select MN Bypass mode. (equal to input M of DPLL M=2 to 2047 => DPLL multiplies by M). 0h (R/W) = 0 : Reserved 1h (R/W) = 1 : Reserved
7	RESERVED	R-0	0h	
6-0	DPLL_DIV	R/W	0h	DPLL divider factor (0 to 127) (equal to input N of DPLL actual division factor is N+1).

6.13.8.56 PRCM_CM_DIV_M2_DPLL_DISP Register (Offset = 630h) [reset = 1h]

PRCM_CM_DIV_M2_DPLL_DISP is shown in Figure 6-311 and described in Table 6-329.

Return to Summary Table.

This register provides controls over the M2 divider of the DPLL.

Figure 6-311. PRCM_CM_DIV_M2_DPLL_DISP Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED						ST_DPLL_CLKOUT	DPLL_CLKOUT_GATE_CTRL
R-0-0h						R-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED		DPLL_CLKOUT_DIVCHACK	DPLL_CLKOUT_DIV				
R-0-0h		R-0h	R/W-1h				

Table 6-329. PRCM_CM_DIV_M2_DPLL_DISP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R-0	0h	
9	ST_DPLL_CLKOUT	R	0h	DPLL CLKOUT status 0h (R) = The clock output is gated 1h (R) = The clock output is enabled
8	DPLL_CLKOUT_GATE_CTRL	R/W	0h	Control gating of DPLL CLKOUT 0h (R/W) = Automatically gate this clock when there is no dependency for it 1h (R/W) = Force this clock to stay enabled even if there is no request
7-6	RESERVED	R-0	0h	
5	DPLL_CLKOUT_DIVCHACK	R	0h	Toggle on this status bit after changing DPLL_CLKOUT_DIV indicates that the change in divider value has taken effect
4-0	DPLL_CLKOUT_DIV	R/W	1h	DPLL M2 post-divider factor (1 to 31). 0h (R/W) = Reserved

6.13.8.57 PRCM_CM_SSC_DELTAMSTEP_DPLL_DISP Register (Offset = 648h) [reset = 0h]

PRCM_CM_SSC_DELTAMSTEP_DPLL_DISP is shown in Figure 6-312 and described in Table 6-330.

Return to Summary Table.

Control the DeltaMStep parameter for Spread Spectrum Clocking technique
DeltaMStep is split into fractional and integer part.
[warm reset insensitive]

Figure 6-312. PRCM_CM_SSC_DELTAMSTEP_DPLL_DISP Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED				DELTAMSTEP_INTEGER		DELTAMSTEP_FRACTION	
R-0-0h				R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
DELTAMSTEP_FRACTION							
R/W-0h							
7	6	5	4	3	2	1	0
DELTAMSTEP_FRACTION							
R/W-0h							

Table 6-330. PRCM_CM_SSC_DELTAMSTEP_DPLL_DISP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	
19-18	DELTAMSTEP_INTEGER	R/W	0h	Integer part of coefficient
17-0	DELTAMSTEP_FRACTION	R/W	0h	Fractional setting for DeltaMStep parameter

6.13.8.58 PRCM_CM_SSC_MODFREQDIV_DPLL_DISP Register (Offset = 64Ch) [reset = 0h]

PRCM_CM_SSC_MODFREQDIV_DPLL_DISP is shown in Figure 6-313 and described in Table 6-331.

Return to Summary Table.

Control the Modulation Frequency (Fm) for Spread Spectrum Clocking technique by defining it as a ratio of DPLL_REFCLK/4
Fm = [DPLL_REFCLK/4]/MODFREQDIV
MODFREQDIV = MODFREQDIV_MANTISSA * $2^{\text{MODFREQDIV\_EXPONENT}}$
[warm reset insensitive]

Figure 6-313. PRCM_CM_SSC_MODFREQDIV_DPLL_DISP Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="5">RESERVED</td><td colspan="3">MODFREQDIV_EXPONENT</td></tr>
<tr><td colspan="5">R-0-0h</td><td colspan="3">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>RESERVED</td><td colspan="7">MODFREQDIV_MANTISSA</td></tr>
<tr><td>R-0-0h</td><td colspan="7">R/W-0h</td></tr>
</table>

Table 6-331. PRCM_CM_SSC_MODFREQDIV_DPLL_DISP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	
10-8	MODFREQDIV_EXPONENT	R/W	0h	Set the Exponent component of MODFREQDIV factor
7	RESERVED	R-0	0h	
6-0	MODFREQDIV_MANTISSA	R/W	0h	Set the Mantissa component of MODFREQDIV factor

6.13.8.59 PRCM_CM_CLKMODE_DPLL_EXTDEV Register (Offset = 660h) [reset = 4h]

PRCM_CM_CLKMODE_DPLL_EXTDEV is shown in Figure 6-314 and described in Table 6-332.

Return to Summary Table.

This register allows controlling the DPLL modes.

Figure 6-314. PRCM_CM_CLKMODE_DPLL_EXTDEV Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DPLL_SSC_TYPE	DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	RESERVED			
R/W-0h	R/W-0h	R-0h	R/W-0h	R-0-0h			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					DPLL_EN		
R-0-0h					R/W-4h		

Table 6-332. PRCM_CM_CLKMODE_DPLL_EXTDEV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	
15	DPLL_SSC_TYPE	R/W	0h	Select between Triangular and SquareWave Spread Spectrum Clocking 0h (R/W) = Triangular Spread Spectrum Clocking is selected 1h (R/W) = Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)
14	DPLL_SSC_DOWNSPREAD	R/W	0h	Control if only low frequency spread is required 0h (R/W) = When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 1h (R/W) = When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency
13	DPLL_SSC_ACK	R	0h	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0h (R) = SSC has been turned off on PLL o/ps 1h (R) = SSC has been turned on on PLL o/ps
12	DPLL_SSC_EN	R/W	0h	Enable or disable Spread Spectrum Clocking 0h (R/W) = SSC disabled 1h (R/W) = SSC enabled
11-3	RESERVED	R-0	0h	
2-0	DPLL_EN	R/W	4h	DPLL control. Upon Warm Reset, the PRCM DPLL control state machine updates this register to reflect DPLL Low Power Stop mode. 0h (R/W) = Reserved 1h (R/W) = Reserved 2h (R/W) = Reserved2 3h (R/W) = Reserved 4h (R/W) = Put the DPLL in MN Bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 5h (R/W) = Put the DPLL in Idle Bypass Low Power mode. 6h (R/W) = Reserved 7h (R/W) = Enables the DPLL in Lock mode

6.13.8.60 PRCM_CM_IDLEST_DPLL_EXTDEV Register (Offset = 664h) [reset = 0h]

PRCM_CM_IDLEST_DPLL_EXTDEV is shown in Figure 6-315 and described in Table 6-333.

Return to Summary Table.

This register allows monitoring the master clock activity. This register is read only and automatically updated. [warm reset insensitive]

Figure 6-315. PRCM_CM_IDLEST_DPLL_EXTDEV Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							ST_MN_BYPASS
R-0-0h							R-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							ST_DPLL_CLK
R-0-0h							R-0h

Table 6-333. PRCM_CM_IDLEST_DPLL_EXTDEV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R-0	0h	
8	ST_MN_BYPASS	R	0h	DPLL MN_BYPASS status 0h (R) = DPLL is not in MN_Bypass 1h (R) = DPLL is in MN_Bypass
7-1	RESERVED	R-0	0h	
0	ST_DPLL_CLK	R	0h	DPLL clock activity 0h (R) = DPLL is either in bypass mode or in stop mode. 1h (R) = DPLL is LOCKED

6.13.8.61 PRCM_CM_CLKSEL_DPLL_EXTDEV Register (Offset = 66Ch) [reset = 0h]

PRCM_CM_CLKSEL_DPLL_EXTDEV is shown in Figure 6-316 and described in Table 6-334.

Return to Summary Table.

This register provides controls over the DPLL.

Figure 6-316. PRCM_CM_CLKSEL_DPLL_EXTDEV Register

31	30	29	28	27	26	25	24
DPLL_SD_DIV							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED	RESERVED			DPLL_MULT			
R-0-0h	R-0-0h			R/W-0h			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DPLL_MULT							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DPLL_DIV							
R/W-0h							

Table 6-334. PRCM_CM_CLKSEL_DPLL_EXTDEV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	DPLL_SD_DIV	R/W	0h	Sigma-Delta divider select (2-255). This factor must be set by s/w to ensure optimum jitter performance. DPLL_SD_DIV = CEILING ([DPLL_MULT/(DPLL_DIV+1)] * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. 0h (R/W) = Reserved 1h (R/W) = Reserved
23	RESERVED	R-0	0h	
22-20	RESERVED	R-0	0h	
19-8	DPLL_MULT	R/W	0h	DPLL multiplier factor (2 to 4095). This register is automatically cleared to 0 when the DPLL_EN field in the *CLKMODE_DPLL* register is set to select MN Bypass mode. (equal to input M of DPLL M=2 to 4095 => DPLL multiplies by M). 0h (R/W) = 0 : Reserved 1h (R/W) = 1 : Reserved
7-0	DPLL_DIV	R/W	0h	DPLL divider factor (0 to 255) (equal to input N of DPLL actual division factor is N+1).

6.13.8.62 PRCM_CM_DIV_M2_DPLL_EXTDEV Register (Offset = 670h) [reset = 1h]

PRCM_CM_DIV_M2_DPLL_EXTDEV is shown in Figure 6-317 and described in Table 6-335.

Return to Summary Table.

This register provides controls over the M2 divider of the DPLL.

Figure 6-317. PRCM_CM_DIV_M2_DPLL_EXTDEV Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED						ST_DPLL_CLKOUT	DPLL_CLKOUT_GATE_CTRL
R-0-0h						R-0h	R/W-0h
7	6	5	4	3	2	1	0
DPLL_CLKOUT_DIVCHACK	DPLL_CLKOUT_DIV						
R-0h	R/W-1h						

Table 6-335. PRCM_CM_DIV_M2_DPLL_EXTDEV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R-0	0h	
9	ST_DPLL_CLKOUT	R	0h	DPLL CLKOUT status 0h (R) = The clock output is gated 1h (R) = The clock output is enabled
8	DPLL_CLKOUT_GATE_CTRL	R/W	0h	Control gating of DPLL CLKOUT 0h (R/W) = Automatically gate this clock when there is no dependency for it 1h (R/W) = Force this clock to stay enabled even if there is no request
7	DPLL_CLKOUT_DIVCHACK	R	0h	Toggle on this status bit after changing DPLL_CLKOUT_DIV indicates that the change in divider value has taken effect
6-0	DPLL_CLKOUT_DIV	R/W	1h	DPLL M2 post-divider factor (1 to 31). 0h (R/W) = Reserved

6.13.8.63 PRCM_CM_CLKSEL2_DPLL_EXTDEV Register (Offset = 684h) [reset = 00080000h]

PRCM_CM_CLKSEL2_DPLL_EXTDEV is shown in Figure 6-318 and described in Table 6-336.

Return to Summary Table.

This register provides DPLL fractional multiplier factor control, SELFREQDCO control and BandWidth Control for EXTDEV DPLL.

Figure 6-318. PRCM_CM_CLKSEL2_DPLL_EXTDEV Register

31	30	29	28	27	26	25	24
RESERVED						BW_CTRL	
R-0-0h						R/W-0h	

23	22	21	20	19	18	17	16
RESERVED	BW_INCR_DECRZ	RESERVED	FREQSELDCO			DPLL_MULT_FRAC	
R-0-0h	R/W-0h	R-0-0h	R/W-2h			R/W-0h	

15	14	13	12	11	10	9	8
DPLL_MULT_FRAC							
R/W-0h							

7	6	5	4	3	2	1	0
DPLL_MULT_FRAC							
R/W-0h							

Table 6-336. PRCM_CM_CLKSEL2_DPLL_EXTDEV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	RESERVED
25-24	BW_CTRL	R/W	0h	Register to change the BandWidth of the PLL. This field should be used only for DEBUG purpose. 0h (R/W) = BW mode is 1x for BW_INCR_DECRZ=0/1 1h (R/W) = BW mode is 0.5x for BW_INCR_DECRZ=0 and BW mode is 2x for BW_INCR_DECRZ=1 2h (R/W) = BW mode is 0.25x for BW_INCR_DECRZ=0 and is not supported for BW_INCR_DECRZ=1 3h (R/W) = BW mode is 0.125x for BW_INCR_DECRZ=0 and is not supported for BW_INCR_DECRZ=1
23	RESERVED	R-0	0h	RESERVED
22	BW_INCR_DECRZ	R/W	0h	Register for controlling BandWidth increase/decrease. This field should be used only for DEBUG purpose. 0h (R/W) = Decreases the BandWidth 1h (R/W) = Increases the BandWidth
21	RESERVED	R-0	0h	RESERVED
20-18	FREQSELDCO	R/W	2h	Register for controlling the input 'freqseldco' for the DPLL_EXTDEV. Legal values are '010'[if DCO clk is in the range 500-1000MHz] and '100'[if DCO clk is in the range 1000-2000MHz] 0h (R) = Reserved 1h (R) = Reserved 2h (R/W) = This values needs to be selected in the DCO clk freq is in the range 500-1000 MHz. 3h (R) = Reserved 4h (R/W) = This values needs to be selected in the DCO clk freq is in the range 1000-2000 MHz. 5h (R/W) = RESERVED 6h (R) = RESERVED 7h (R) = RESERVED
17-0	DPLL_MULT_FRAC	R/W	0h	DPLL fractional multiplier factor. Setting to 0 keeps DPLL in integer mode.

6.13.8.64 PRCM_CM_SSC_DELTAMSTEP_DPLL_EXTDEV Register (Offset = 688h) [reset = 0h]

PRCM_CM_SSC_DELTAMSTEP_DPLL_EXTDEV is shown in Figure 6-319 and described in Table 6-337.

Return to Summary Table.

Control the DeltaMStep parameter for Spread Spectrum Clocking technique
DeltaMStep is split into fractional and integer part.
[warm reset insensitive]

Figure 6-319. PRCM_CM_SSC_DELTAMSTEP_DPLL_EXTDEV Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="2">DELTAMSTEP_INTEGER</td><td colspan="2">DELTAMSTEP_FRACTION</td></tr>
<tr><td colspan="4">R-0-0h</td><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">DELTAMSTEP_FRACTION</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">DELTAMSTEP_FRACTION</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 6-337. PRCM_CM_SSC_DELTAMSTEP_DPLL_EXTDEV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	
19-18	DELTAMSTEP_INTEGER	R/W	0h	Integer part for DeltaM coefficient
17-0	DELTAMSTEP_FRACTION	R/W	0h	Fractional setting for DeltaMStep parameter

6.13.8.65 PRCM_CM_SSC_MODFREQDIV_DPLL_EXTDEV Register (Offset = 68Ch) [reset = 0h]

PRCM_CM_SSC_MODFREQDIV_DPLL_EXTDEV is shown in Figure 6-320 and described in Table 6-338.

Return to Summary Table.

Control the Modulation Frequency (Fm) for Spread Spectrum Clocking technique by defining it as a ratio of DPLL_REFCLK/4
Fm = [DPLL_REFCLK/4]/MODFREQDIV
MODFREQDIV = MODFREQDIV_MANTISSA * $2^{\text{MODFREQDIV\_EXPONENT}}$
[warm reset insensitive]

Figure 6-320. PRCM_CM_SSC_MODFREQDIV_DPLL_EXTDEV Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED					MODFREQDIV_EXPONENT		
R-0-0h					R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	MODFREQDIV_MANTISSA						
R-0-0h	R/W-0h						

Table 6-338. PRCM_CM_SSC_MODFREQDIV_DPLL_EXTDEV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	
10-8	MODFREQDIV_EXPONENT	R/W	0h	Set the Exponent component of MODFREQDIV factor
7	RESERVED	R-0	0h	
6-0	MODFREQDIV_MANTISSA	R/W	0h	Set the Mantissa component of MODFREQDIV factor

6.13.8.66 PRCM_CM_SHADOW_FREQ_CONFIG1 Register (Offset = 7A0h) [reset = D0Ch]

PRCM_CM_SHADOW_FREQ_CONFIG1 is shown in Figure 6-321 and described in Table 6-339.

Return to Summary Table.

Shadow register to program new DPLL configuration affecting EMIF and GPMC (L3 clock) functional frequency during DVFS. The PRCM h/w automatically applies the new configuration after EMIF/GPMC have been put in idle state.

Figure 6-321. PRCM_CM_SHADOW_FREQ_CONFIG1 Register

31	30	29	28	27	26	25	24
RSVD5							
R-0h							
23	22	21	20	19	18	17	16
RSVD5							
R-0h							
15	14	13	12	11	10	9	8
DPLL_DDR_M2_DIV					DPLL_DDR_EN		
R/W-1h					R/W-5h		
7	6	5	4	3	2	1	0
RSVD2				DLL_RESET	DLL_OVERRIDE	RSVD1	FREQ_UPDATE
R-0h				R/W-1h	R/W-1h	R-0h	R/WSpecial-0h

Table 6-339. PRCM_CM_SHADOW_FREQ_CONFIG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RSVD5	R	0h	
15-11	DPLL_DDR_M2_DIV	R/W	1h	Shadow register for CM_DIV_M2_DPLL_DDR.DPLL_CLKOUT_DIV. The main register is loaded by WKUP-M3 with the shadow register value after EMIF IDLE if the FREQ_UPDATE field is set to '1'. Divide value from 1 to 31. 0h (R/W) = Reserved
10-8	DPLL_DDR_EN	R/W	5h	Shadow register for CM_CLKMODE_DPLL_DDR.DPLL_EN. The main register is loaded by WKUP-M3 with the shadow register value after EMIF IDLE if the FREQ_UPDATE field is set to '1'. 0h (R/W) = Reserved 1h (R/W) = Reserved 2h (R/W) = Reserved 3h (R/W) = Reserved 4h (R/W) = Put the DPLL in MN Bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 5h (R/W) = Put the DPLL in Idle Bypass Low Power mode. 6h (R/W) = Put the DPLL in Idle Bypass Fast Relock mode. 7h (R/W) = Enables the DPLL in Lock mode
7-4	RSVD2	R	0h	
3	DLL_RESET	R/W	1h	Specify if DLL should be reset or not during the frequency change hardware sequence. 0h (R/W) = DLL is not reset during the frequency change hardware sequence 1h (R/W) = DLL is reset automatically during the frequency change hardware sequence

Table 6-339. PRCM_CM_SHADOW_FREQ_CONFIG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DLL_OVERRIDE	R/W	1h	Shadow register for CM_DLL_CTRL.DLL_OVERRIDE.The main register is automatically loaded with the shadow register value after EMIF IDLE if the FREQ_UPDATE field is set to '1'. 0h (R/W) = Lock and code outputs are not overriden 1h (R/W) = Lock output is overriden to '1' and code output is overriden with a value coming from control module.
1	RSVD1	R	0h	Reserved
0	FREQ_UPDATE	R/WSpecial	0h	Writing '1' indicates that a new configuration is available. It is automatically cleared by h/w after the configuration has been applied.

6.13.8.67 PRCM_CM_SHADOW_FREQ_CONFIG2 Register (Offset = 7A4h) [reset = 410h]

PRCM_CM_SHADOW_FREQ_CONFIG2 is shown in Figure 6-322 and described in Table 6-340.

Return to Summary Table.

Shadow register to program new DPLL configuration affecting GPMC (L3 clock) functional frequency during DVFS. The PRCM h/w automatically applies the new configuration after EMIF/GPMC have been put in idle state.

Figure 6-322. PRCM_CM_SHADOW_FREQ_CONFIG2 Register

31	30	29	28	27	26	25	24
RSVD6							
R-0h							
23	22	21	20	19	18	17	16
RSVD6							
R-0h							
15	14	13	12	11	10	9	8
RSVD6					DPLL_CORE_EN		
R-0h					R/W-4h		
7	6	5	4	3	2	1	0
RSVD5	DPLL_CORE_M4_DIV					RSVD1	GPMC_FREQ_UPDATE
R-0h	R/W-4h					R-0h	R/W-0h

Table 6-340. PRCM_CM_SHADOW_FREQ_CONFIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RSVD6	R	0h	
10-8	DPLL_CORE_EN	R/W	4h	Shadow register for CM_CLKMODE_DPLL_CORE.DPLL_EN. 0h (R/W) = Reserved 1h (R/W) = Reserved 2h (R/W) = Reserved 3h (R/W) = Reserved 4h (R/W) = Put the DPLL in MN Bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 5h (R/W) = Put the DPLL in Idle Bypass Low Power mode. 6h (R/W) = Put the DPLL in Idle Bypass Fast Relock mode. 7h (R/W) = Enables the DPLL in Lock mode
7	RSVD5	R	0h	
6-2	DPLL_CORE_M4_DIV	R/W	4h	Shadow register for CM_DIV_M4_DPLL_CORE.DIV. The main register is loaded by WKUP-M3 with the shadow register value after GPMC IDLE if the CM_SHADOW_FREQ_CONFIG1.FREQ_UPDATE field is set to '1' and GPMC_FREQ_UPDATE is set to '1'. Divide value from 1 to 31. 0h (R/W) = Reserved
1	RSVD1	R	0h	Reserved
0	GPMC_FREQ_UPDATE	R/W	0h	Controls whether or not GPMC has to be put automatically into idle during the frequency change operation. 0h (R/W) = GPMC is not put automatically into idle during frequency change operation. 1h (R/W) = GPMC is put automatically into idle during frequency change operation.

Chapter 7

SPRUHL7H–April 2014–Revised January 2018

Control Module

This chapter describes the control module of the device.

Topic **Page**

7.1 Introduction

The control module includes status and control logic not addressed within the peripherals or the rest of the device infrastructure. This module provides interface to control the following areas of the device:

- Functional I/O multiplexing
- Emulation controls
- Device control and status
- DDR PHY control and IO control registers
- EDMA event multiplexing control registers

Note: For writing to the control module registers, the MPU will need to be in privileged mode of operation and writes will not work from user mode.

7.2 Functional Description

7.2.1 Pad Control Registers

The Pad Control Registers are 32-bit registers to control the signal muxing and other aspects of each I/O pad. After POR, software must set the pad functional multiplexing and configuration registers to the desired values according to the requested device configuration. The configuration is controlled by pads or by a group of pads. Each configurable pin has its own configuration register for pullup/down control and for the assignment to a given module.

CAUTION

The multiplexer controlling the signal mux mode selection is not a glitch free structure. Therefore, it is possible to see the signal glitch for a few nanoseconds during the MUXMODE change. The user should ensure a glitch will not cause contention or negatively impact an external device connected to the pad

Table 7-1 shows the generic Pad Control Register Description.

Table 7-1. Pad Control Register Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved.
30	WUEVT		Wakeup event.
		0	No event.
		1	Event occurred.
29	WUEN		Wakeup enable.
		0	Disable.
		1	Enable.
28	DSPULLTYPSELECT		DS0 mode pullup/down selection.
		0	Offmode pulldown selected.
		1	Offmode pull-up selected.
27	DSPULLUDENABLE		DS0 mode pullup/down enable. This is an active low signal.
		0	Pullup/down enabled.
		1	Pullup/down disabled.
26	DSOUTVALUE		DS0 mode output value.
		0	Set value at 0.
		1	Set value at 1.
25	DSOUTENABLE		DS0 mode output enable. This is an active low signal.
		0	Output enabled.

Table 7-1. Pad Control Register Field Descriptions (continued)

Bit	Field	Value	Description
		1	Output disabled.
24	DSENABLE		DS0 mode override control. Note that DSENABLE must be set to 1 only if the PAD values during DS conflict with the system required value.
		0	IO state retains previous state when DS0 mode is active.
		1	IO state is forced to OFF mode value when DS0 mode is active.
23-20	Reserved	0	Reserved. Read returns 0.
19	SLEWCTRL		Select between faster or slower slew rate.
		0	Fast
		1	Slow[1]
18	RXACTIVE		Input enable value for the pad. Set to 0 for output only. Set to 1 for input or output.
		0	Receive disabled.
		1	Receiver enabled.
17	PULLTYPESEL		Pad pullup/down type selection.
		0	Pulldown selected.
		1	Pullup selected.
16	PULLUDEN		Pad pullup/down enable. This is an active low signal.
		0	Pullup/down enabled.
		1	Pullup/down disabled.
15-4	Reserved	0	Reserved. Read returns 0.
3-0	MUXMODE		Pad functional signal mux select.

(1) Some peripherals do not support slow slew rate. To determine which interfaces support each slew rate, see the datasheet *AM437x ARM Cortex-A9 Processors* (literature number SPRS851).

7.2.1.1 Mode Selection

The MUXMODE field in the pad control registers defines the multiplexing mode applied to the pad. Modes are referred to by their decimal (from 0 to 9) or binary (from 0b0000 to 0b1001) representation. For most pads, the reset value for the MUXMODE field in the registers is 0b111. The exceptions are pads to be used at boot time to transfer data from selected peripherals to the external flash memory.

Table 7-2. Mode Selection

MUXMODE	Selected Mode
0000b	Primary Mode = Mode 0
0001b	Mode 1
0010b	Mode 2
0011b	Mode 3
0100b	Mode 4
0101b	Mode 5
0110b	Mode 6
0111b	Mode 7
1000b	Mode 8
1001b	Mode 9

Mode 0 is the primary mode. When mode 0 is set, the function mapped to the pin corresponds to the name of the pin. Mode 1 to mode 9 are possible modes for alternate functions. On each pin, some modes are used effectively for alternate functions, while other modes are unused and correspond to no functional configuration.

NOTE: See the device-specific datasheet for a complete list of the signals corresponding to the mux modes for each pin.

7.2.1.2 Pull Selection

There is no automatic gating control to ensure that internal weak pull- down/pull up resistors on a pad are disconnected whenever the pad is configured as output. If a pad is always configured in output mode, it is recommended for user software to disable any internal pull resistor tied to it, to avoid unnecessary consumption. The following table summarizes the various possible combinations of PULLTYPESEL and PULLUDEN fields of PAD control register.

Table 7-3. Pull Selection

PULL TYPE		**Pin Behavior**
PULLTYPESEL	PULLUDENABLE	
0b	0b	Pulldown selected and activated
0b	1b	Pulldown selected but not activated
1b	0b	Pullup selected and activated
1b	1b	Pullup selected but not activated

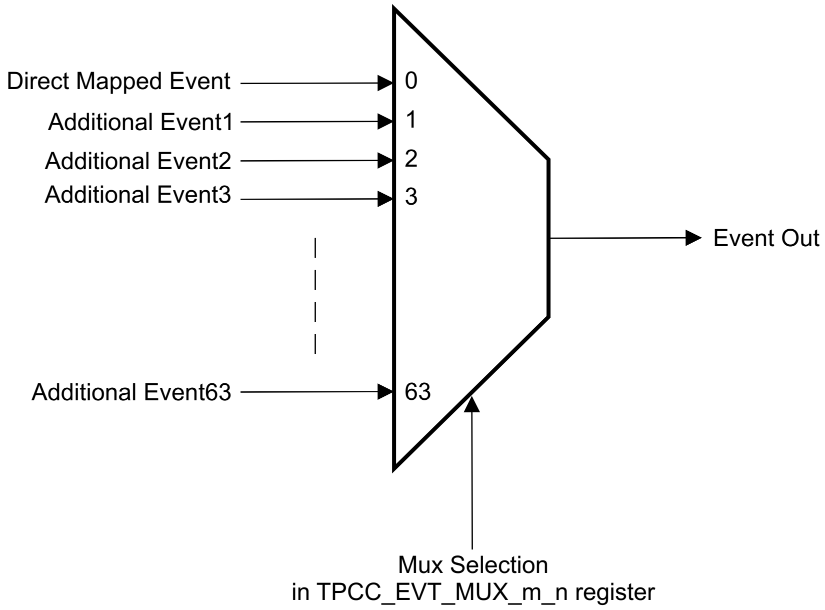
7.2.1.3 RX Active

The RXACTIVE bit is used to enable and disable the input buffer. This control can be used to help with power leakage or device isolation through the I/O. The characteristic of the signal is ultimately dictated by the mux mode the pad is put into.

7.2.2 EDMA Event Multiplexing

To accommodate the large number of possible DMA events, the device includes an event crossbar which multiplexes a direct mapped event with additional event choices for every EDMA event input. Mux control registers are defined in the Control Module to select the event to be routed to the EDMA3CC (TPCC). The direct mapped event is the default (mux selection set to ‘0’).

Figure 7-1. Event Crossbar



For every EDMA event there is a cross bar implemented in the design as shown in the figure. The direct mapped event/interrupt is always connected to Mux input[0], The additional events are connected to Mux input[1], Mux input[2], and so on, as defined in Table 10-23, *Direct Mapped*. The Mux selection value is programmed into the corresponding TPCC_EVT_MUX_n register. The EVT_MUX value can take a value from 1 to 63. Other values are reserved. By default the MUX_selection value is written to 0, which means the direct mapped event is connected to the Event output.

When one of the additional events is selected through the Cross bar programming, the direct mapped event cannot be used.

For example, when the TINT0 (Timer Interrupt 0) event, which is not directly mapped to the DMA event source, must be connected to EDMA channel 24, which is directly mapped to the SDTXEVT0 event. The user must program the EVT_MUX_24 field in TPCC_EVT_MUX_24_27 register to 22 (the value corresponding to TINT0 interrupt in the crossbar mapping). When this field is set, the TINT0 interrupt event acts as the channel 24 event trigger.

Please note: Once the field is set, the SDTXEVT0 event can no longer be handled by EDMA because it cannot be mapped to any of the other EDMACC event inputs. The user must use caution to allocate only unused direct mapped event inputs to the crossbar mapped events to ensure no compromise on the channel allocation for the used event numbers.

7.2.3 *Device Control and Status*

7.2.3.1 Control and Boot Status

The device configuration is set during power on or hardware reset (PORz sequence) by the configuration input pins (SYSBOOT[18:0]).The CTRL_STS register reflects the system boot and the device type configuration values as sampled when the power-on reset (PORz) signal is asserted. The Configuration input pins are sampled continuously during the PORz active period and the final sampled value prior to the last rising edge is latched in the register. The CTRL_STS register gives the status of the device boot process.

7.2.3.2 Interprocessor Communication

The control module has the IPC_MSG_REG (14:0) registers which is for sharing messages between the Wakeup Processor and the Cortex-A9 MPU. The Wakeup Processor TX end of event (WAKEPROC_TXEV_EOI) register provides the mechanism to clear/enable the TX Event from the Wakeup Processor to Cortex-A9 MPU Subsystem. See the WAKEPROC_TXEV_EOI register description for further detail.

For specific information on how the IPC_MSG_REG registers are used to communicate with the Wakeup Processor firmware, see Section 6.4.6, *Functional Sequencing for Power Management with Wakeup Processor*.

7.2.3.3 Initiator Priority Control

The control module provides the registers to control the bus interconnect priority and the EMIF priority.

7.2.3.3.1 Initiator Priority Control for Interconnect

The INIT_PRIORITY_n register controls the infrastructure priority at the bus interconnects. This can be used for dynamic priority escalation. There are bit fields that control the interconnect priority for each bus initiator. By default all the initiators are given equal priority and the allocation is done on a round robin basis.

The priority can take a value from 0 to 3. The following table gives the valid set of priority values.

Table 7-4. Interconnect Priority Values

Interconnect Priority Value	Remarks
00	Low priority
01	Medium priority

Table 7-4. Interconnect Priority Values (continued)

Interconnect Priority Value	Remarks
10	Reserved
11	High priority

7.2.3.3.2 Initiator Priority at EMIF

The MREQPRIO register provides an interface to change the access priorities for the various masters accessing the EMIF(DDR). Software can make use of this register to set the requestor priorities for required EMIF arbitration. The EMIF priority can take a value from 000b to 111b where 000b will be the highest priority and 111b will be lowest priority.

7.2.3.4 Peripheral Control and Status

7.2.3.4.1 USB Control and Status

The USB_CTRLn and USB_STSn registers reflect the Control and Status of the USB instances. The USB IO lines can be used as UART TX and RX lines the USB Control register bit field GPIOMODE has settings that configures the USB lines as GPIO lines. The other USB PHY control settings for controlling the OTG settings and PHY are part of the USB_CTRLn register.

The USB_STSn register gives the status of the USB PHY module. See the USB_STSn register description for further details.

7.2.3.4.2 USB Charger Detect

Each USB PHY contains circuitry which can automatically detect the presence of a charger attached to the USB port. The charger detection circuitry is compliant to the Battery Charging Specification Revision 1.1 from the USB Implementers Forum, which can be found at www.usb.org. See this document for more details on USB charger implementation.

7.2.3.4.2.1 Features

The charger detection circuitry of each PHY has the following features:

- Contains a state machine which can automatically detect the presence of a Charging Downstream Port or a Dedicated Charging Port (see the Battery Charging Specification for the definition of these terms)
- Outputs a charger enable signal (3.3 V level active high CMOS driver) when a charger is present.
- Allows you to enable/disable the circuitry to save power
- The detection circuitry requires only a 3.3-V supply to be present to operate.
- The charger detection also has a manual mode which allows the user to implement the battery charging specification in software.

7.2.3.4.2.2 Operation

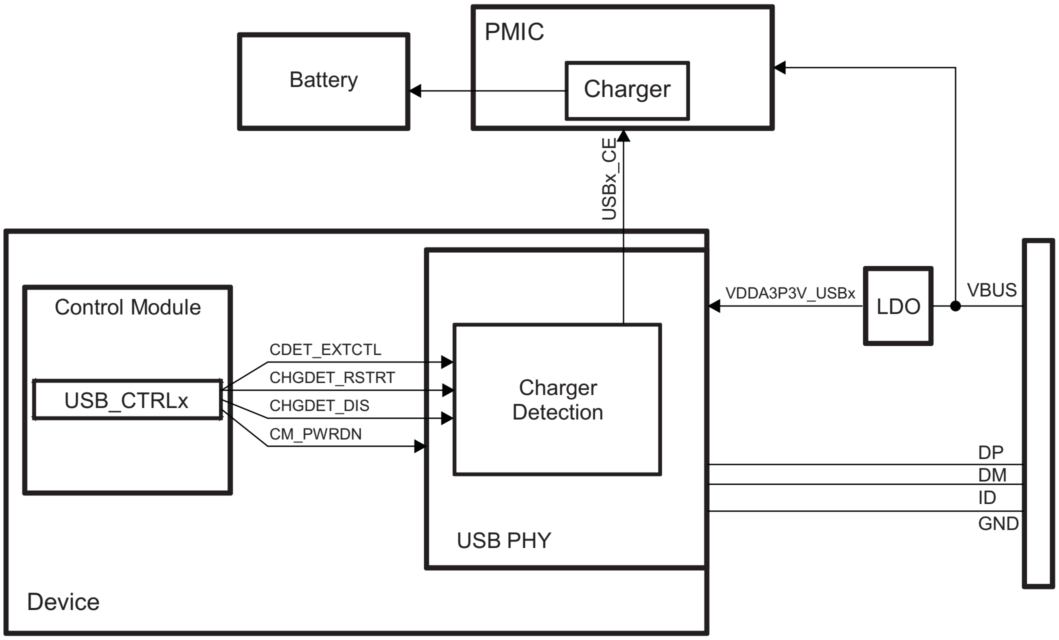
The control module gives the following interface to control the automatic charger detection circuitry:

- USB_CTRLx.CDET_EXTCTL: Turns the automatic detection on/off. Keep this bit 0 to keep the automatic detection on. Changing this to 1 enables the manual mode.
- USB_CTRLx.CHGDET_RSTRT: Restarts the charger detection state machine. To initiate the charger detection, change this bit from 1 to 0. If this bit is 1, the charger enable output (CE) is disabled.
- USB_CTRLx.CHGDET_DIS: Enables/disables the charger detection circuitry. Keep this bit 0 to keep this charger detection enabled. Setting this bit to 1 will power down the charger detection circuitry.
- USB_CTRLx.CM_PWRDN: Powers up/down the PHY which contains the charger detection circuitry. Clear this bit to 0 to enable power to the PHY.

To start the charger detection during normal operation, ensure that the PHY and charger are enabled and the automatic detection is turned on. Then, initiate a charger detection cycle by transitioning CHGDET_RSTRT from 1 to 0. If a Charging Downstream Port or a Dedicated Charging Port is detected, the charger enable signal (USBx_CE) will be driven high and remain high until the charger is disabled by either CHGDET_DIS = 1 or CHGDET_RSTRT=1. If the port remains unconnected after intiating the charger detect cycle, it will continue the detection until a charger is detected or an error condition occurs. Note that USBx_CE is not an open drain output.

To disable the charger after successful detection, you must disable the charger detect circuitry with CHGDET_DIS or CHGDET_RSTRT, even if the charger is physically disconnected.

Figure 7-2. USB Charger Detection



Charger detection can be automatically started with no power to the rest of the device. If VDDA3P3V_USBx is present, via an LDO powered by VBUS connected to a host, the charger detection state machine will automatically start and perform detection. If a charger is detected, USBx_CE will be driven high, otherwise it will be driven low.

The charger detection circuitry performs the following steps of the Battery Charging specification v1.1:

1. VBUS Detect
2. Data Contact Detect
3. Primary Detection

Secondary Detection (to distinguish between a Charging Downstream Port and a Dedicated Charging Port) is a newly added feature of the v1.2 spec and is not implemented in the charger detection state machine.

NOTE: The USBx_CE output will only operate when the corresponding USBx_ID pin is grounded (indicating USB host mode). The USBx_CE output does not operate in peripheral mode (when USBx_ID is floating).

7.2.3.4.3 Ethernet MII Mode Selection

The control module provides a mechanism to select the Mode of operation of Ethernet MII interface. The GMII_SEL register has register bit fields to select the MII/RMII/RGMII modes, clock sources, and delay mode.

7.2.3.4.4 Ethernet Module Reset Isolation Control

This feature allows the device to undergo a warm reset without disrupting the switch or traffic being routed through the switch during the reset condition. The CPSW Reset Isolation register (RESET_ISO) has an ISO_CONTROL field which controls the reset isolation feature.

If the reset isolation is enabled, any warm reset source will be blocked to the EMAC switch. If the EMAC reset isolation is NOT active (default state), then the warm reset sources are allowed to propagate as normal including to the EMAC Switch module (both reset inputs to the IP). All cold or POR resets will always propagate to the EMAC switch module as normal.

When RESET_ISO is enabled, the following registers will not be disturbed by a warm reset:

- GMII_SEL
- CONF_GPMC_A[11:0]
- CONF_GPMC_WAIT0
- CONF_GPMC_WPN
- CONF_GPMC_BEN1
- CONF_GPMC_CSN2
- CONF_GPMC_CSN3
- CONF_MII1_COL
- CONF_MII1_CRS
- CONF_MII1_RXERR
- CONF_MII1_TXEN
- CONF_MII1_RXDV
- CONF_MII1_TXD[3:0]
- CONF_MII1_TXCLK
- CONF_MII1_RXCLK
- CONF_MII1_RXD[3:0]
- CONF_RMII1_REFCLK
- CONF_MDIO
- CONF_MDC

7.2.3.4.5 Timer/eCAP Event Capture Control

The capture event input sources for Timer5, 6, 7 and eCAP0, 1, 2 are selected using the TIMER_EVT_CAPTURE and ECAP_EVT_CAPTURE registers. The following table lists the available sources for those events.

Table 7-5. Available Sources for Timer[5–7] and eCAP[0–2] Events

Event No.	Source module	Interrupt Name/Pin
0	For Timer 5 MUX input from IO signal TIMER5	TIMER5 IO pin
	For Timer 6 MUX input from IO signal TIMER6	TIMER6 IO pin
	For Timer 7 MUX input from IO signal TIMER7	TIMER7 IO pin
	For eCAP 0 MUX input from IO signal eCAP0	eCAP0 IO pin
	For eCAP 1 MUX input from IO signal eCAP1	eCAP1 IO pin
	For eCAP 2 MUX input from IO signal eCAP2	eCAP2 IO pin
1	UART0	UART0INT
2	UART1	UART1INT
3	UART2	UART2INT
4	UART3	UART3INT
5	UART4	UART4INT
6	UART5	UART5INT
7	3PGSW	3PGSWRXTHR0
8	3PGSW	3PGSWRXINT0
9	3PGSW	3PGSWTXINT0
10	3PGSW	3PGSWMISC0
11	McASP0	MCATXINT0
12	McASP0	MCARXINT0
13	McASP1	MCATXINT1
14	McASP1	MCARXINT1
15	GPIO4	GPIOINT4A
16	GPIO4	GPIOINT4B
17	GPIO0	GPIOINT0A
18	GPIO0	GPIOINT0B
19	GPIO1	GPIOINT1A
20	GPIO1	GPIOINT1B
21	GPIO2	GPIOINT2A
22	GPIO2	GPIOINT2B
23	GPIO3	GPIOINT3A
24	GPIO3	GPIOINT3B
25	DCAN0	DCAN0_INT0
26	DCAN0	DCAN0_INT1
27	DCAN0	DCAN0_PARITY
28	DCAN1	DCAN1_INT0
29	DCAN1	DCAN1_INT1
30	DCAN1	DCAN1_PARITY

7.2.3.4.6 ADC0 Capture Control

The following chip level events can be connected through the software-controlled multiplexer to the ADC0 module.

1. PRU-ICSS1 Host Event 0
2. Timer 4 Event
3. Timer 5 Event
4. Timer 6 Event
5. Timer 7 Event
6. ext_hw_trigger

This pin is the external hardware trigger to start the ADC0 channel conversion. The ADC0_EVT_CAPT register needs to be programmed to select the proper source for this conversion.

Figure 7-3. ADC0 External Hardware Events

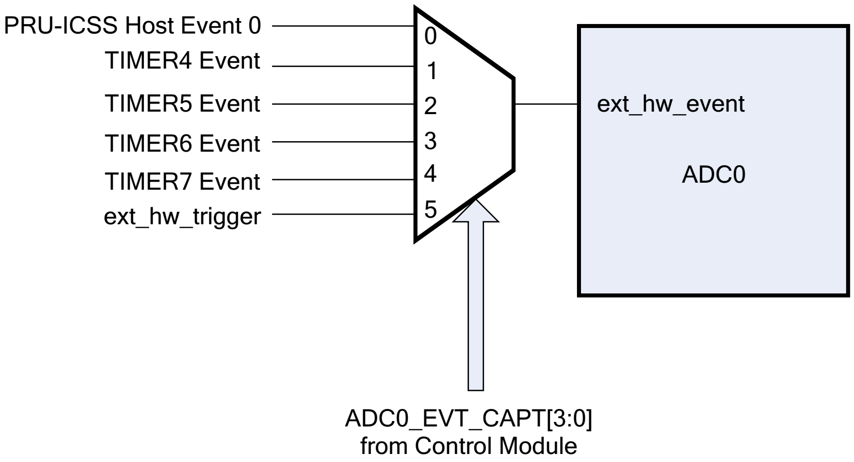


Table 7-6 contains the value to be programmed in the selection mux.

Table 7-6. Selection Mux Values

ADC0_EVT_CAPT Value	ADC0 External Event Selected
000	PRU-ICSS1 Host Event 0
001	Timer 4 Event
010	Timer 5 Event
011	Timer 6 Event
100	Timer 7 Event
101	ext_hw_trigger
110-111	Reserved

7.2.3.4.7 ADC1 Capture Control

The following chip level events can be connected through the software-controlled multiplexer to the ADC1 module.

1. PRU-ICSS1 Host Event 0
2. Timer 4 Event
3. Timer 5 Event
4. Timer 6 Event
5. Timer 7 Event
6. ext_hw_trigger

This pin is the external hardware trigger to start the ADC channel conversion. The ADC1_EVT_CAPT register needs to be programmed to select the proper source for this conversion.

Figure 7-4. ADC1 External Hardware Events

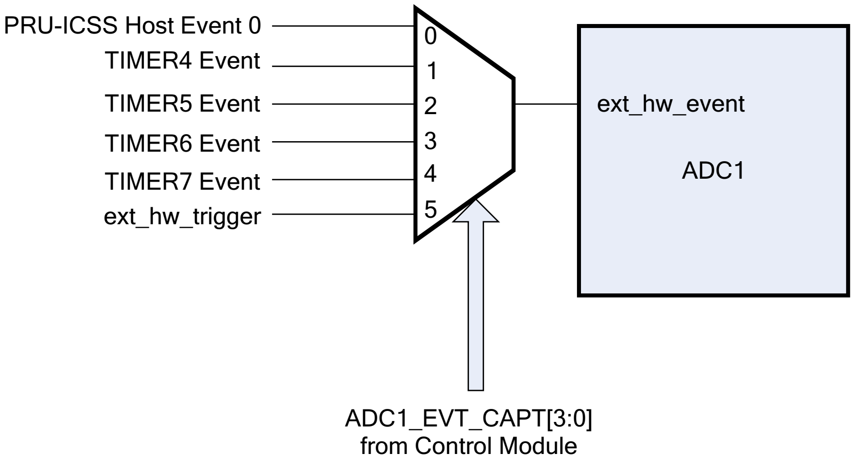


Table 7-7 contains the value to be programmed in the selection mux.

Table 7-7. Selection Mux Values

ADC1_EVT_CAPT Value	ADC1 External Event Selected
000	PRU-ICSS1 Host Event 0
001	Timer 4 Event
010	Timer 5 Event
011	Timer 6 Event
100	Timer 7 Event
101	ext_hw_trigger
110-111	Reserved

7.2.3.4.8 SRAM LDO Control

The device incorporates two instances of the SRAM LDO (VSLDO) module. One of these LDOs powers the ARM internal SRAM and the other powers the OCMC SRAMs. In the CTRL_VSLDO register, the VSLDO_CORE_AUTO_RAMP_EN bit, when set, allows the VSLDO, which powers the OCMC SRAMs, to be put into retention during deepsleep and enable lower power consumption. Since the VSLDO is shared between WKUP_PROC memories and CORE memories, the VSLDO has to be brought out of retention on any wakeup event. This bit allows this functionality and should be set to allow proper sleep/wakeup operation during Standby and DeepSleep modes. Similar functionality is not necessary for the LDO powering the ARM internal SRAM. It can be put in retention mode using PRM_LDO_SRAM_MPU_CTRL.

7.2.4 DDR IO Control Settings

Table 7-8 describes the slew rate settings available on the DDR IOs.

Table 7-8. DDR Slew Rate Control Settings(1)

sr1	sr0	Slew Rate Level
0	0	Fastest
1	0	Fast
0	1	Slow
1	1	Slowest

(1) These values are programmed in the following registers: ddr_addrctrl_ioctrl, ddr_data0_ioctrl, ddr_data1_ioctrl, ddr_data2_ioctrl, ddr_data3_ioctrl.

Table 7-9 describes the impedance control settings available on the DDR IOs.

Table 7-9. DDR Impedance Control Settings(1) (2)

I2	I1	I0	Output Impedance (R_{on})	Drive Strength $\|I_{OH}\|$, $\|I_{OL}\|$	Example: R_{on} for R_{ext} = 49.9 ohms	Example: $\|I_{OH}\|$, $\|I_{OL}\|$ for R_{ext} = 49.9 ohms
0	0	0	$1.6*R_{ext}$	$0.625*I_{out}$	80 ohms	5 mA
0	0	1	$1.33*R_{ext}$	$0.75*I_{out}$	67 ohms	6 mA
0	1	0	$1.14*R_{ext}$	$0.875*I_{out}$	57 ohms	7 mA
0	1	1	R_{ext}	I_{out}	50 ohms	8 mA
1	0	0	$0.88*R_{ext}$	$1.125*I_{out}$	44 ohms	9 mA
1	0	1	$0.8*R_{ext}$	$1.250*I_{out}$	40 ohms	10 mA
1	1	0	$0.73*R_{ext}$	$1.375*I_{out}$	36 ohms	11 mA
1	1	1	$0.67*R_{ext}$	$1.5*I_{out}$	33 ohms	12 mA

(1) These values are programmed in the following registers: ddr_addrctrl_ioctrl, ddr_data0_ioctrl, ddr_data1_ioctrl, ddr_data2_ioctrl, ddr_data3_ioctrl.

(2) R_{ext} is the external VTP compensation resistor connected to DDR_VTP terminal.

7.2.4.1 DDR IO Pin Mapping

Table 7-10 describes the address control mapping for LPDDR2/DDR3.

Table 7-10. Address Control Mapping for LPDDR2/DDR3

Device	LPDDR2	DDR3
ddr_ck	CK_t	CK
ddr_nck	CK_c	CK#
ddr_rasn	CA0	RAS#
ddr_casn	CA1	CAS#
ddr_wen	CA2	WE#
ddr_a13	CA3	A13
ddr_cke1	CKE1	CKE1
ddr_cke0	CKE0	CKE0
ddr_csn1	CS1_n	CS1#
ddr_csn0	CS0_n	CS0#
ddr_a10	CA4	A10/AP
ddr_a1	CA5	A1
ddr_a2	CA6	A2
ddr_ba0	CA7	BA0
ddr_ba1	CA8	BA1

Table 7-10. Address Control Mapping for LPDDR2/DDR3 (continued)

Device	LPDDR2	DDR3
ddr_ba2	CA9	BA2
ddr_a0	Unconn	A0
ddr_a3	Unconn	A3
ddr_a4	Unconn	A4
ddr_a5	Unconn	A5
ddr_a6	Unconn	A6
ddr_a7	Unconn	A7
ddr_a8	Unconn	A8
ddr_a9	Unconn	A9
ddr_a11	Unconn	A11
ddr_a12	Unconn	A12/BC#
ddr_a14	Unconn	A14
ddr_a15	Unconn	A15
ddr_resetn	Unconn	RESET#
ddr_odt0	Unconn	ODT0
ddr_odt1	Unconn	ODT1

7.3 Registers

7.3.1 CONTROL_MODULE Registers

Table 7-11 lists the memory-mapped registers for the CONTROL_MODULE. All register offset addresses not listed in Table 7-11 should be considered as reserved locations and the register contents should not be modified.

Table 7-11. CONTROL_MODULE Registers

Offset	Acronym	Register Name	Section
0h	CTRL_REVISION		Section 7.3.1.1
4h	CTRL_HWINFO		Section 7.3.1.2
10h	CTRL_SYSCONFIG		Section 7.3.1.3
40h	CTRL_STS		Section 7.3.1.4
1E0h	CTRL_MPU_L2		Section 7.3.1.5
428h	CTRL_CORE_SLDO		Section 7.3.1.6
42Ch	CTRL_MPU_SLDO		Section 7.3.1.7
444h	CTRL_CLK32KDIVRATIO		Section 7.3.1.8
448h	CTRL_BANDGAP		Section 7.3.1.9
44Ch	CTRL_BANDGAP_TRIM		Section 7.3.1.10
458h	CTRL_PLL_CLKINPULOW		Section 7.3.1.11
468h	CTRL_MOSC		Section 7.3.1.12
470h	CTRL_DEEPSLEEP		Section 7.3.1.13
50Ch	CTRL_DPLL_PWR_SW_STS		Section 7.3.1.14
534h	CTRL_DISPLAY_PLL_SEL		Section 7.3.1.15
600h	CTRL_DEVICE_ID		Section 7.3.1.16
604h	CTRL_DEV_FEATURE		Section 7.3.1.17
608h	CTRL_INIT_PRIORITY_0		Section 7.3.1.18
60Ch	CTRL_INIT_PRIORITY_1		Section 7.3.1.19
610h	CTRL_DEV_ATTR		Section 7.3.1.20
614h	CTRL_TPTC_CFG		Section 7.3.1.21

Table 7-11. CONTROL_MODULE Registers (continued)

Offset	Acronym	Register Name	Section
620h	CTRL_USB_CTRL0		Section 7.3.1.22
624h	CTRL_USB_STS0		Section 7.3.1.23
628h	CTRL_USB_CTRL1		Section 7.3.1.24
62Ch	CTRL_USB_STS1		Section 7.3.1.25
630h	CTRL_MAC_ID0_LO		Section 7.3.1.26
634h	CTRL_MAC_ID0_HI		Section 7.3.1.27
638h	CTRL_MAC_ID1_LO		Section 7.3.1.28
63Ch	CTRL_MAC_ID1_HI		Section 7.3.1.29
644h	CTRL_DCAN_RAMINIT		Section 7.3.1.30
64Ch	CTRL_USB_CTRL2		Section 7.3.1.31
650h	CTRL_GMII_SEL		Section 7.3.1.32
654h	CTRL_MPUSS		Section 7.3.1.33
658h	CTRL_TIMER_CASCADE		Section 7.3.1.34
664h	CTRL_PWMSS		Section 7.3.1.35
670h	CTRL_MREQPRIO_0		Section 7.3.1.36
674h	CTRL_MREQPRIO_1		Section 7.3.1.37
770h	CTRL_VDD_MPU_OPP_050		Section 7.3.1.38
774h	CTRL_VDD_MPU_OPP_100		Section 7.3.1.39
778h	CTRL_VDD_MPU_OPP_120		Section 7.3.1.40
77Ch	CTRL_VDD_MPU_OPP_TURBO		Section 7.3.1.41
780h	CTRL_VDD_MPU_OPP_NITRO		Section 7.3.1.42
7B8h	CTRL_VDD_CORE_OPP_050		Section 7.3.1.43
7BCh	CTRL_VDD_CORE_OPP_100		Section 7.3.1.44
7F4h	CTRL_USB_VID_PID		Section 7.3.1.45
800h	CTRL_CONF_GPMC_AD0		Section 7.3.1.46
804h	CTRL_CONF_GPMC_AD1		Section 7.3.1.47
808h	CTRL_CONF_GPMC_AD2		Section 7.3.1.48
80Ch	CTRL_CONF_GPMC_AD3		Section 7.3.1.49
810h	CTRL_CONF_GPMC_AD4		Section 7.3.1.50
814h	CTRL_CONF_GPMC_AD5		Section 7.3.1.51
818h	CTRL_CONF_GPMC_AD6		Section 7.3.1.52
81Ch	CTRL_CONF_GPMC_AD7		Section 7.3.1.53
820h	CTRL_CONF_GPMC_AD8		Section 7.3.1.54
824h	CTRL_CONF_GPMC_AD9		Section 7.3.1.55
828h	CTRL_CONF_GPMC_AD10		Section 7.3.1.56
82Ch	CTRL_CONF_GPMC_AD11		Section 7.3.1.57
830h	CTRL_CONF_GPMC_AD12		Section 7.3.1.58
834h	CTRL_CONF_GPMC_AD13		Section 7.3.1.59
838h	CTRL_CONF_GPMC_AD14		Section 7.3.1.60
83Ch	CTRL_CONF_GPMC_AD15		Section 7.3.1.61
840h	CTRL_CONF_GPMC_A0		Section 7.3.1.62
844h	CTRL_CONF_GPMC_A1		Section 7.3.1.63
848h	CTRL_CONF_GPMC_A2		Section 7.3.1.64
84Ch	CTRL_CONF_GPMC_A3		Section 7.3.1.65
850h	CTRL_CONF_GPMC_A4		Section 7.3.1.66
854h	CTRL_CONF_GPMC_A5		Section 7.3.1.67
858h	CTRL_CONF_GPMC_A6		Section 7.3.1.68

Table 7-11. CONTROL_MODULE Registers (continued)

Offset	Acronym	Register Name	Section
85Ch	CTRL_CONF_GPMC_A7		Section 7.3.1.69
860h	CTRL_CONF_GPMC_A8		Section 7.3.1.70
864h	CTRL_CONF_GPMC_A9		Section 7.3.1.71
868h	CTRL_CONF_GPMC_A10		Section 7.3.1.72
86Ch	CTRL_CONF_GPMC_A11		Section 7.3.1.73
870h	CTRL_CONF_GPMC_WAIT0		Section 7.3.1.74
874h	CTRL_CONF_GPMC_WPN		Section 7.3.1.75
878h	CTRL_CONF_GPMC_BE1N		Section 7.3.1.76
87Ch	CTRL_CONF_GPMC_CSN0		Section 7.3.1.77
880h	CTRL_CONF_GPMC_CSN1		Section 7.3.1.78
884h	CTRL_CONF_GPMC_CSN2		Section 7.3.1.79
888h	CTRL_CONF_GPMC_CSN3		Section 7.3.1.80
88Ch	CTRL_CONF_GPMC_CLK		Section 7.3.1.81
890h	CTRL_CONF_GPMC_ADVN_ALE		Section 7.3.1.82
894h	CTRL_CONF_GPMC_OEN_REN		Section 7.3.1.83
898h	CTRL_CONF_GPMC_WEN		Section 7.3.1.84
89Ch	CTRL_CONF_GPMC_BE0N_CLE		Section 7.3.1.85
8A0h	CTRL_CONF_DSS_DATA0		Section 7.3.1.86
8A4h	CTRL_CONF_DSS_DATA1		Section 7.3.1.87
8A8h	CTRL_CONF_DSS_DATA2		Section 7.3.1.88
8ACh	CTRL_CONF_DSS_DATA3		Section 7.3.1.89
8B0h	CTRL_CONF_DSS_DATA4		Section 7.3.1.90
8B4h	CTRL_CONF_DSS_DATA5		Section 7.3.1.91
8B8h	CTRL_CONF_DSS_DATA6		Section 7.3.1.92
8BCh	CTRL_CONF_DSS_DATA7		Section 7.3.1.93
8C0h	CTRL_CONF_DSS_DATA8		Section 7.3.1.94
8C4h	CTRL_CONF_DSS_DATA9		Section 7.3.1.95
8C8h	CTRL_CONF_DSS_DATA10		Section 7.3.1.96
8CCh	CTRL_CONF_DSS_DATA11		Section 7.3.1.97
8D0h	CTRL_CONF_DSS_DATA12		Section 7.3.1.98
8D4h	CTRL_CONF_DSS_DATA13		Section 7.3.1.99
8D8h	CTRL_CONF_DSS_DATA14		Section 7.3.1.100
8DCh	CTRL_CONF_DSS_DATA15		Section 7.3.1.101
8E0h	CTRL_CONF_DSS_VSYNC		Section 7.3.1.102
8E4h	CTRL_CONF_DSS_HSYNC		Section 7.3.1.103
8E8h	CTRL_CONF_DSS_PCLK		Section 7.3.1.104
8ECh	CTRL_CONF_DSS_AC_BIAS_EN		Section 7.3.1.105
8F0h	CTRL_CONF_MMC0_DAT3		Section 7.3.1.106
8F4h	CTRL_CONF_MMC0_DAT2		Section 7.3.1.107
8F8h	CTRL_CONF_MMC0_DAT1		Section 7.3.1.108
8FCh	CTRL_CONF_MMC0_DAT0		Section 7.3.1.109
900h	CTRL_CONF_MMC0_CLK		Section 7.3.1.110
904h	CTRL_CONF_MMC0_CMD		Section 7.3.1.111
908h	CTRL_CONF_MII1_COL		Section 7.3.1.112
90Ch	CTRL_CONF_MII1_CRS		Section 7.3.1.113
910h	CTRL_CONF_MII1_RXERR		Section 7.3.1.114
914h	CTRL_CONF_MII1_TXEN		Section 7.3.1.115

Table 7-11. CONTROL_MODULE Registers (continued)

Offset	Acronym	Register Name	Section
918h	CTRL_CONF_MII1_RXDV		Section 7.3.1.116
91Ch	CTRL_CONF_MII1_TXD3		Section 7.3.1.117
920h	CTRL_CONF_MII1_TXD2		Section 7.3.1.118
924h	CTRL_CONF_MII1_TXD1		Section 7.3.1.119
928h	CTRL_CONF_MII1_TXD0		Section 7.3.1.120
92Ch	CTRL_CONF_MII1_TXCLK		Section 7.3.1.121
930h	CTRL_CONF_MII1_RXCLK		Section 7.3.1.122
934h	CTRL_CONF_MII1_RXD3		Section 7.3.1.123
938h	CTRL_CONF_MII1_RXD2		Section 7.3.1.124
93Ch	CTRL_CONF_MII1_RXD1		Section 7.3.1.125
940h	CTRL_CONF_MII1_RXD0		Section 7.3.1.126
944h	CTRL_CONF_RMII1_REFCLK		Section 7.3.1.127
948h	CTRL_CONF_MDIO_DATA		Section 7.3.1.128
94Ch	CTRL_CONF_MDIO_CLK		Section 7.3.1.129
950h	CTRL_CONF_SPI0_SCLK		Section 7.3.1.130
954h	CTRL_CONF_SPI0_D0		Section 7.3.1.131
958h	CTRL_CONF_SPI0_D1		Section 7.3.1.132
95Ch	CTRL_CONF_SPI0_CS0		Section 7.3.1.133
960h	CTRL_CONF_SPI0_CS1		Section 7.3.1.134
964h	CTRL_CONF_ECAP0_IN_PWM0_OUT		Section 7.3.1.135
968h	CTRL_CONF_UART0_CTSN		Section 7.3.1.136
96Ch	CTRL_CONF_UART0_RTSN		Section 7.3.1.137
970h	CTRL_CONF_UART0_RXD		Section 7.3.1.138
974h	CTRL_CONF_UART0_TXD		Section 7.3.1.139
978h	CTRL_CONF_UART1_CTSN		Section 7.3.1.140
97Ch	CTRL_CONF_UART1_RTSN		Section 7.3.1.141
980h	CTRL_CONF_UART1_RXD		Section 7.3.1.142
984h	CTRL_CONF_UART1_TXD		Section 7.3.1.143
988h	CTRL_CONF_I2C0_SDA		Section 7.3.1.144
98Ch	CTRL_CONF_I2C0_SCL		Section 7.3.1.145
990h	CTRL_CONF_MCASP0_ACLKX		Section 7.3.1.146
994h	CTRL_CONF_MCASP0_FSX		Section 7.3.1.147
998h	CTRL_CONF_MCASP0_AXR0		Section 7.3.1.148
99Ch	CTRL_CONF_MCASP0_AHCLKR		Section 7.3.1.149
9A0h	CTRL_CONF_MCASP0_ACLKR		Section 7.3.1.150
9A4h	CTRL_CONF_MCASP0_FSR		Section 7.3.1.151
9A8h	CTRL_CONF_MCASP0_AXR1		Section 7.3.1.152
9ACh	CTRL_CONF_MCASP0_AHCLKX		Section 7.3.1.153
9B0h	CTRL_CONF_CAM0_HD		Section 7.3.1.154
9B4h	CTRL_CONF_CAM0_VD		Section 7.3.1.155
9B8h	CTRL_CONF_CAM0_FIELD		Section 7.3.1.156
9BCh	CTRL_CONF_CAM0_WEN		Section 7.3.1.157
9C0h	CTRL_CONF_CAM0_PCLK		Section 7.3.1.158
9C4h	CTRL_CONF_CAM0_DATA8		Section 7.3.1.159
9C8h	CTRL_CONF_CAM0_DATA9		Section 7.3.1.160
9CCh	CTRL_CONF_CAM1_DATA9		Section 7.3.1.161
9D0h	CTRL_CONF_CAM1_DATA8		Section 7.3.1.162

Table 7-11. CONTROL_MODULE Registers (continued)

Offset	Acronym	Register Name	Section
9D4h	CTRL_CONF_CAM1_HD		Section 7.3.1.163
9D8h	CTRL_CONF_CAM1_VD		Section 7.3.1.164
9DCh	CTRL_CONF_CAM1_PCLK		Section 7.3.1.165
9E0h	CTRL_CONF_CAM1_FIELD		Section 7.3.1.166
9E4h	CTRL_CONF_CAM1_WEN		Section 7.3.1.167
9E8h	CTRL_CONF_CAM1_DATA0		Section 7.3.1.168
9ECh	CTRL_CONF_CAM1_DATA1		Section 7.3.1.169
9F0h	CTRL_CONF_CAM1_DATA2		Section 7.3.1.170
9F4h	CTRL_CONF_CAM1_DATA3		Section 7.3.1.171
9F8h	CTRL_CONF_CAM1_DATA4		Section 7.3.1.172
9FCh	CTRL_CONF_CAM1_DATA5		Section 7.3.1.173
A00h	CTRL_CONF_CAM1_DATA6		Section 7.3.1.174
A04h	CTRL_CONF_CAM1_DATA7		Section 7.3.1.175
A08h	CTRL_CONF_CAM0_DATA0		Section 7.3.1.176
A0Ch	CTRL_CONF_CAM0_DATA1		Section 7.3.1.177
A10h	CTRL_CONF_CAM0_DATA2		Section 7.3.1.178
A14h	CTRL_CONF_CAM0_DATA3		Section 7.3.1.179
A18h	CTRL_CONF_CAM0_DATA4		Section 7.3.1.180
A1Ch	CTRL_CONF_CAM0_DATA5		Section 7.3.1.181
A20h	CTRL_CONF_CAM0_DATA6		Section 7.3.1.182
A24h	CTRL_CONF_CAM0_DATA7		Section 7.3.1.183
A28h	CTRL_CONF_UART3_RXD		Section 7.3.1.184
A2Ch	CTRL_CONF_UART3_TXD		Section 7.3.1.185
A30h	CTRL_CONF_UART3_CTSN		Section 7.3.1.186
A34h	CTRL_CONF_UART3_RTSN		Section 7.3.1.187
A38h	CTRL_CONF_GPIO5_8		Section 7.3.1.188
A3Ch	CTRL_CONF_GPIO5_9		Section 7.3.1.189
A40h	CTRL_CONF_GPIO5_10		Section 7.3.1.190
A44h	CTRL_CONF_GPIO5_11		Section 7.3.1.191
A48h	CTRL_CONF_GPIO5_12		Section 7.3.1.192
A4Ch	CTRL_CONF_GPIO5_13		Section 7.3.1.193
A50h	CTRL_CONF_SPI4_SCLK		Section 7.3.1.194
A54h	CTRL_CONF_SPI4_D0		Section 7.3.1.195
A58h	CTRL_CONF_SPI4_D1		Section 7.3.1.196
A5Ch	CTRL_CONF_SPI4_CS0		Section 7.3.1.197
A60h	CTRL_CONF_SPI2_SCLK		Section 7.3.1.198
A64h	CTRL_CONF_SPI2_D0		Section 7.3.1.199
A68h	CTRL_CONF_SPI2_D1		Section 7.3.1.200
A6Ch	CTRL_CONF_SPI2_CS0		Section 7.3.1.201
A70h	CTRL_CONF_XDMA_EVT_INTR0		Section 7.3.1.202
A74h	CTRL_CONF_XDMA_EVT_INTR1		Section 7.3.1.203
A78h	CTRL_CONF_CLKREQ		Section 7.3.1.204
A7Ch	CTRL_CONF_NRESETIN_OUT		Section 7.3.1.205
A84h	CTRL_CONF_NNMI		Section 7.3.1.206
A90h	CTRL_CONF_TMS		Section 7.3.1.207
A94h	CTRL_CONF_TDI		Section 7.3.1.208
A98h	CTRL_CONF_TDO		Section 7.3.1.209

Table 7-11. CONTROL_MODULE Registers (continued)

Offset	Acronym	Register Name	Section
A9Ch	CTRL_CONF_TCK		Section 7.3.1.210
AA0h	CTRL_CONF_NTRST		Section 7.3.1.211
AA4h	CTRL_CONF_EMU0		Section 7.3.1.212
AA8h	CTRL_CONF_EMU1		Section 7.3.1.213
AACh	CTRL_CONF_OSC1_IN		Section 7.3.1.214
AB0h	CTRL_CONF_OSC1_OUT		Section 7.3.1.215
AB4h	CTRL_CONF_RTC_PORZ		Section 7.3.1.216
AB8h	CTRL_CONF_EXT_WAKEUP0		Section 7.3.1.217
ABCh	CTRL_CONF_PMIC_POWER_EN0		Section 7.3.1.218
AC0h	CTRL_CONF_USB0_DRVVBUS		Section 7.3.1.219
AC4h	CTRL_CONF_USB1_DRVVBUS		Section 7.3.1.220
E00h	CTRL_CQDETECT_STS		Section 7.3.1.221
E04h	CTRL_DDR_IO		Section 7.3.1.222
E08h	CTRL_CQDETECT_STS2		Section 7.3.1.223
E0Ch	CTRL_VTP		Section 7.3.1.224
E14h	CTRL_VREF		Section 7.3.1.225
F90h	CTRL_TPCC_EVT_MUX_0_3		Section 7.3.1.226
F94h	CTRL_TPCC_EVT_MUX_4_7		Section 7.3.1.227
F98h	CTRL_TPCC_EVT_MUX_8_11		Section 7.3.1.228
F9Ch	CTRL_TPCC_EVT_MUX_12_15		Section 7.3.1.229
FA0h	CTRL_TPCC_EVT_MUX_16_19		Section 7.3.1.230
FA4h	CTRL_TPCC_EVT_MUX_20_23		Section 7.3.1.231
FA8h	CTRL_TPCC_EVT_MUX_24_27		Section 7.3.1.232
FACh	CTRL_TPCC_EVT_MUX_28_31		Section 7.3.1.233
FB0h	CTRL_TPCC_EVT_MUX_32_35		Section 7.3.1.234
FB4h	CTRL_TPCC_EVT_MUX_36_39		Section 7.3.1.235
FB8h	CTRL_TPCC_EVT_MUX_40_43		Section 7.3.1.236
FBCh	CTRL_TPCC_EVT_MUX_44_47		Section 7.3.1.237
FC0h	CTRL_TPCC_EVT_MUX_48_51		Section 7.3.1.238
FC4h	CTRL_TPCC_EVT_MUX_52_55		Section 7.3.1.239
FC8h	CTRL_TPCC_EVT_MUX_56_59		Section 7.3.1.240
FCCh	CTRL_TPCC_EVT_MUX_60_63		Section 7.3.1.241
FD0h	CTRL_TIMER_EVT_CAPT		Section 7.3.1.242
FD4h	CTRL_ECAP_EVT_CAPT		Section 7.3.1.243
FD8h	CTRL_ADC0_EVT_CAPT		Section 7.3.1.244
FDCh	CTRL_ADC1_EVT_CAPT		Section 7.3.1.245
1000h	CTRL_RESET_ISO		Section 7.3.1.246
1318h	CTRL_DPLL_PWR_SW		Section 7.3.1.247
131Ch	CTRL_DDR_CKE		Section 7.3.1.248
1320h	CTRL_VSLDO		Section 7.3.1.249
1324h	CTRL_WAKEPROC_TXEV_EOI		Section 7.3.1.250
1328h	CTRL_IPC_MSG_REG0		Section 7.3.1.251
132Ch	CTRL_IPC_MSG_REG1		Section 7.3.1.252
1330h	CTRL_IPC_MSG_REG2		Section 7.3.1.253
1334h	CTRL_IPC_MSG_REG3		Section 7.3.1.254
1338h	CTRL_IPC_MSG_REG4		Section 7.3.1.255
133Ch	CTRL_IPC_MSG_REG5		Section 7.3.1.256

Table 7-11. CONTROL_MODULE Registers (continued)

Offset	Acronym	Register Name	Section
1340h	CTRL_IPC_MSG_REG6		Section 7.3.1.257
1344h	CTRL_IPC_MSG_REG7		Section 7.3.1.258
1348h	CTRL_IPC_MSG_REG8		Section 7.3.1.259
134Ch	CTRL_IPC_MSG_REG9		Section 7.3.1.260
1350h	CTRL_IPC_MSG_REG10		Section 7.3.1.261
1354h	CTRL_IPC_MSG_REG11		Section 7.3.1.262
1358h	CTRL_IPC_MSG_REG12		Section 7.3.1.263
135Ch	CTRL_IPC_MSG_REG13		Section 7.3.1.264
1360h	CTRL_IPC_MSG_REG14		Section 7.3.1.265
1364h	CTRL_IPC_INTR		Section 7.3.1.266
138Ch	CTRL_DPLL_PWR_SW_CTRL2		Section 7.3.1.267
1390h	CTRL_DPLL_PWR_SW_STS2		Section 7.3.1.268
1394h	CTRL_RESET_MISC		Section 7.3.1.269
1404h	CTRL_DDR_ADDRCTRL_IOCTRL		Section 7.3.1.270
1408h	CTRL_DDR_ADDRCTRL_WD0_IOCTRL		Section 7.3.1.271
140Ch	CTRL_DDR_ADDRCTRL_WD1_IOCTRL		Section 7.3.1.272
1440h	CTRL_DDR_DATA0_IOCTRL		Section 7.3.1.273
1444h	CTRL_DDR_DATA1_IOCTRL		Section 7.3.1.274
1448h	CTRL_DDR_DATA2_IOCTRL		Section 7.3.1.275
144Ch	CTRL_DDR_DATA3_IOCTRL		Section 7.3.1.276
1460h	CTRL_EMIF_SDRAM_CONFIG_EXT		Section 7.3.1.277
1464h	CTRL_EMIF_SDRAM_STS_EXT		Section 7.3.1.278
3000h	CTRL_DISPPLL_CLKCTRL		Section 7.3.1.279
3004h	CTRL_DISPPLL_TEN		Section 7.3.1.280
3008h	CTRL_DISPPLL_TENIV		Section 7.3.1.281
300Ch	CTRL_DISPPLL_M2NDIV		Section 7.3.1.282
3010h	CTRL_DISPPLL_MN2DIV		Section 7.3.1.283
3014h	CTRL_DISPPLL_FRACDIV		Section 7.3.1.284
3018h	CTRL_DISPPLL_BWCTRL		Section 7.3.1.285
301Ch	CTRL_DISPPLL_FRACCTRL		Section 7.3.1.286
3020h	CTRL_DISPPLL_STS		Section 7.3.1.287
3024h	CTRL_DISPPLL_M3DIV		Section 7.3.1.288
3028h	CTRL_DISPPLL_RAMPCTRL		Section 7.3.1.289

7.3.1.1 CTRL_REVISION Register (Offset = 0h) [reset = 4F000100h]

Register mask: FFFFFFFFh

CTRL_REVISION is shown in Figure 7-5 and described in Table 7-12.

Return to Summary Table.

Figure 7-5. CTRL_REVISION Register

31	30	29	28	27	26	25	24
IP_REV_SCHEME		RESERVED		IP_REV_FUNC			
R-1h		R-0h		R-F00h			
23	22	21	20	19	18	17	16
IP_REV_FUNC							
R-F00h							
15	14	13	12	11	10	9	8
IP_REV_RTL					IP_REV_MAJOR		
R-0h					R-1h		
7	6	5	4	3	2	1	0
IP_REV_CUSTOM		IP_REV_MINOR					
R-0h		R-0h					

Table 7-12. CTRL_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	IP_REV_SCHEME	R	1h	Scheme value
29-28	RESERVED	R	0h	
27-16	IP_REV_FUNC	R	F00h	Function value
15-11	IP_REV_RTL	R	0h	RTL Version value
10-8	IP_REV_MAJOR	R	1h	Major Revision value
7-6	IP_REV_CUSTOM	R	0h	Custom Version value
5-0	IP_REV_MINOR	R	0h	Minor Revision value

7.3.1.2 CTRL_HWINFO Register (Offset = 4h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_HWINFO is shown in Figure 7-6 and described in Table 7-13.

Return to Summary Table.

Figure 7-6. CTRL_HWINFO Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">IP_HWINFO</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 7-13. CTRL_HWINFO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IP_HWINFO	R	0h	IP Module dependent

7.3.1.3 CTRL_SYSCONFIG Register (Offset = 10h) [reset = 2Ah]

Register mask: FFFFFFFFh

CTRL_SYSCONFIG is shown in Figure 7-7 and described in Table 7-14.

Return to Summary Table.

Figure 7-7. CTRL_SYSCONFIG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="2">STANDBY</td><td colspan="2">IDLEMODE</td><td>FREEEMU</td><td>RESERVED</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="2">R-2h</td><td colspan="2">R/W-2h</td><td>R-1h</td><td>R/W-0h</td></tr>
</table>

Table 7-14. CTRL_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5-4	STANDBY	R	2h	Configure local initiator state management 00 : Force Standby 01 : No Standby Mode 10 : Smart Standby 11 : Smart Standby wakeup capable Reserved in Control Module since it has no local initiator.
3-2	IDLEMODE	R/W	2h	Configure local target state management 00 : Force Idle 01 : No Idle 10 : Smart Idle 11 : Smart Idle wakeup capable
1	FREEEMU	R	1h	Sensitivity to Emulation suspend input. 0 : Module is sensitive to EMU suspend 1 : Module not sensitive to EMU suspend
0	RESERVED	R/W	0h	

7.3.1.4 CTRL_STS Register (Offset = 40h) [reset = X]

CTRL_STS is shown in Figure 7-8 and described in Table 7-15.

Return to Summary Table.

Note: Some of the bits in this register have different functionality depending on the silicon revision.

Figure 7-8. CTRL_STS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="5">RESERVED</td><td>SYSBOOT18</td><td>SYSBOOT17</td><td>SYSBOOT16</td></tr>
<tr><td colspan="5">R/W-0h</td><td>R/W-X</td><td>R/W-X</td><td>R/W-X</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="2">SYSBOOT15_14</td><td colspan="2">SYSBOOT13_12</td><td colspan="2">ADMUX</td><td>WAITEN</td><td>BW</td></tr>
<tr><td colspan="2">R/W-X</td><td colspan="2">R/W-X</td><td colspan="2">R/W-X</td><td>R/W-X</td><td>R/W-X</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="5">RESERVED</td><td colspan="3">DEVTYPE</td></tr>
<tr><td colspan="5">R/W-0h</td><td colspan="3">R-X</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">SYSBOOT0</td></tr>
<tr><td colspan="8">R/W-X</td></tr>
</table>

Table 7-15. CTRL_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26	SYSBOOT18	R/W	X	SYSBOOT 18: (PG1.2 definition) 0 – Do not route EXTCLK to CLKOUT2 1 – Route 25MHz EXTCLK to CLKOUT2 if SYSBOOT[5] = 0 Route 50MHz EXTCLK to CLKOUT2 if SYSBOOT[5] = 1 (PG1.1 definition) Reserved
25	SYSBOOT17	R/W	X	Used by Control Module to determine whether CLKOUT1 pin is selected as default mux mode on device pin xdma_event_intr0. 1: CLKOUT1 is selected as default mux mode 0: Pinmux Mode0 function is selected as the default mux mode
24	SYSBOOT16	R/W	X	SYSBOOT 16: (PG1.2 definition) If USB_CL boot mode is attempted by the ROM, 0 – Port 0 USB DM/DP not swapped 1 – Port 0 USB DM/DP is swapped If USB_MS boot mode is attempted by the ROM, 0 – Port 1 USB DM/DP not swapped 1 – Port 1 USB DM/DP is swapped If both USB_MS and USB_CL boot modes are attempted by the ROM, then DM/DP of both USB ports will be swapped (PG1.1 definition) Reserved
23-22	SYSBOOT15_14	R/W	X	Used to select crystal clock frequency. This register bitfield is valid only when crystal_freq_source (bit 31) is 0. 00: Selects 19.2MHz 01: Selects 24MHz 10: Selects 25MHz 11: Selects 26MHz
21-20	SYSBOOT13_12	R/W	X	Reserved for future use

Table 7-15. CTRL_STS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-18	ADMUX	R/W	X	GPMC CS0 Default Address Muxing 00 : No Addr/Data Muxing 01 : Addr/Addr/Data Muxing 10 : Addr/Data Muxing 11 : Reserved
17	WAITEN	R/W	X	GPMC CS0 Default Wait Enable 0 : Ignore WAIT input 1 : Use WAIT input
16	BW	R/W	X	GPMC CS0 Default Bus Width 0: 8 bit data bus 1: 16 bit data bus
15-11	RESERVED	R/W	0h	
10-8	DEVTYPE	R	X	000: Reserved 001: Reserved 010: Reserved 011: General Purpose (GP) Device 111: Reserved
7-0	SYSBOOT0	R/W	X	SYSBOOT0 (bits 7 to 6): Used by Boot ROM for selecting MII/RMII/RGMII modes(1) 00b - MII 01b - RMII 10b - RGMII with internal delay 11b - RGMII without internal delay SYSBOOT0(bits 5 to 0): ROM Boot Selection

7.3.1.5 CTRL_MPU_L2 Register (Offset = 1E0h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_MPU_L2 is shown in Figure 7-9 and described in Table 7-16.

Return to Summary Table.

Figure 7-9. CTRL_MPU_L2 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							PIUSEL2SRAM
R/W-0h							R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED							
R/W-0h							

Table 7-16. CTRL_MPU_L2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	
16	PIUSEL2SRAM	R/W	0h	Enables MPUSS L2 Cache as SRAM 0 - Select L2 cache operation 1 - Configure L2 cache as L3 OCMC RAM
15-0	RESERVED	R/W	0h	

7.3.1.6 CTRL_CORE_SLDO Register (Offset = 428h) [reset = X]

CTRL_CORE_SLDO is shown in Figure 7-10 and described in Table 7-17.

Return to Summary Table.

Figure 7-10. CTRL_CORE_SLDO Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">VSET</td><td colspan="16">RESERVED</td></tr>
<tr><td colspan="6">R/W-0h</td><td colspan="10">R/W-X</td><td colspan="16">R/W-X</td></tr>
</table>

Table 7-17. CTRL_CORE_SLDO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R/W	0h	
25-16	VSET	R/W	X	Trims VDDAR
15-0	RESERVED	R/W	X	

7.3.1.7 CTRL_MPU_SLDO Register (Offset = 42Ch) [reset = X]

CTRL_MPU_SLDO is shown in Figure 7-11 and described in Table 7-18.

Return to Summary Table.

Figure 7-11. CTRL_MPU_SLDO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VSET										RESERVED															
R/W-0h						R/W-X										R/W-X															

Table 7-18. CTRL_MPU_SLDO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R/W	0h	
25-16	VSET	R/W	X	Trims VDDAR
15-0	RESERVED	R/W	X	

7.3.1.8 CTRL_CLK32KDIVRATIO Register (Offset = 444h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_CLK32KDIVRATIO is shown in Figure 7-12 and described in Table 7-19.

Return to Summary Table.

Figure 7-12. CTRL_CLK32KDIVRATIO Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED							CLKDIVOPP50_EN
R/W-0h							R/W-0h

Table 7-19. CTRL_CLK32KDIVRATIO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	CLKDIVOPP50_EN	R/W	0h	0 : OPP100 operation use ratio for 24MHz to 32KHz division 1 : OPP50 operation use ratio for 12MHz to 32KHz division

7.3.1.9 CTRL_BANDGAP Register (Offset = 448h) [reset = X]

CTRL_BANDGAP is shown in Figure 7-13 and described in Table 7-20.

Return to Summary Table.

Figure 7-13. CTRL_BANDGAP Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DTEMP							
R-X							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CBIASSEL	BGROFF	TMPSOFF	SOC	CLRZ	CONTCONV	ECOZ	TSHUT
R/W-X	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h	R-X	R-X

Table 7-20. CTRL_BANDGAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-8	DTEMP	R	X	temperature data from ADC
7	CBIASSEL	R/W	X	high uses resister divider as reference rather than bandgap. Wait 150us after BGROFF=0 then pull it low
6	BGROFF	R/W	0h	high turns off bandgap (OFF mode)
5	TMPSOFF	R/W	1h	high turns off temperature sensor
4	SOC	R/W	0h	high transition starts new ADC conversion cycle
3	CLRZ	R/W	0h	low resets the digital outputs
2	CONTCONV	R/W	0h	high is continuous conversion mode low is single conversion mode
1	ECOZ	R	X	when low DTEMP is valid
0	TSHUT	R	X	goes high during thermal shutdown event (147C)

7.3.1.10 CTRL_BANDGAP_TRIM Register (Offset = 44Ch) [reset = X]

CTRL_BANDGAP_TRIM is shown in Figure 7-14 and described in Table 7-21.

Return to Summary Table.

Figure 7-14. CTRL_BANDGAP_TRIM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DTRBGAPC								DTRBGAPV							
R/W-X								R/W-X							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTRTEMPS								DTRTEMPSC							
R/W-X								R/W-X							

Table 7-21. CTRL_BANDGAP_TRIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	DTRBGAPC	R/W	X	trim the output voltage of bandgap
23-16	DTRBGAPV	R/W	X	trim the output voltage of bandgap
15-8	DTRTEMPS	R/W	X	trim the temperature sensor
7-0	DTRTEMPSC	R/W	X	trim the temperature sensor

7.3.1.11 CTRL_PLL_CLKINPULOW Register (Offset = 458h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_PLL_CLKINPULOW is shown in Figure 7-15 and described in Table 7-22.

Return to Summary Table.

Figure 7-15. CTRL_PLL_CLKINPULOW Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED					DDR_PLL_CLKINPULOW_SEL	DISP_PLL_CLKINPULOW_SEL	MPU_DPLL_CLKINPULOW_SEL
R/W-0h					R/W-0h	R/W-0h	R/W-0h

Table 7-22. CTRL_PLL_CLKINPULOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2	DDR_PLL_CLKINPULOW_SEL	R/W	0h	0 : Select CORE_CLKOUT_M6 clock as CLKINPULOW 1 : Select PER_CLKOUT_M2 clock as CLKINPULOW
1	DISP_PLL_CLKINPULOW_SEL	R/W	0h	0 : Select CORE_CLKOUT_M6 clock as CLKINPULOW 1 : Select PER_CLKOUT_M2 clock as CLKINPULOW
0	MPU_DPLL_CLKINPULOW_SEL	R/W	0h	0 : Select CORE_CLKOUT_M6 clock as CLKINPULOW 1 : Select PER_CLKOUT_M2 clock as CLKINPULOW

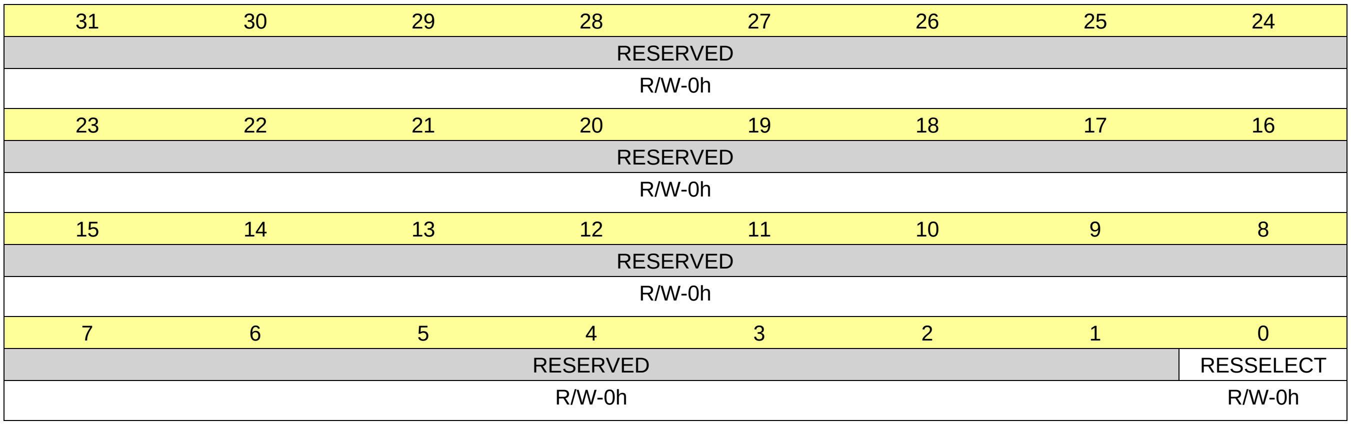
7.3.1.12 CTRL_MOSC Register (Offset = 468h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_MOSC is shown in Figure 7-16 and described in Table 7-23.

Return to Summary Table.

Figure 7-16. CTRL_MOSC Register



<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="7">RESERVED</td><td>RESSELECT</td></tr>
<tr><td colspan="7">R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 7-23. CTRL_MOSC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	RESSELECT	R/W	0h	0 : When reselect is low an internal 1megohm resistor is connected between padxi and padxo for oscillator bias. 1 : When resselect is asserted (high) the internal resistor is disconnected. For oscillation with a crystal while resselect is high an external resistor must be connected between padxi and padxo to provide bias.

7.3.1.13 CTRL_DEEPSLEEP Register (Offset = 470h) [reset = 6A75h]

Register mask: FFFFFFFFh

CTRL_DEEPSLEEP is shown in Figure 7-17 and described in Table 7-24.

Return to Summary Table.

Figure 7-17. CTRL_DEEPSLEEP Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED					FORCE_DSPADCONF_EN	DSEN	RESERVED
R/W-0h					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
DSCOUNT							
R/W-6A75h							
7	6	5	4	3	2	1	0
DSCOUNT							
R/W-6A75h							

Table 7-24. CTRL_DEEPSLEEP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	0h	
18	FORCE_DSPADCONF_EN	R/W	0h	Forces the DSPADCONF by overriding control from PRCM
17	DSEN	R/W	0h	Deep Sleep Enable When this bit is set the master oscillator clock is gated
16	RESERVED	R/W	0h	
15-0	DSCOUNT	R/W	6A75h	Programmable count of how many OSC clocks needs to be seen before exiting deep sleep mode

7.3.1.14 CTRL_DPLL_PWR_SW_STS Register (Offset = 50Ch) [reset = X]

CTRL_DPLL_PWR_SW_STS is shown in Figure 7-18 and described in Table 7-25.

Return to Summary Table.

Figure 7-18. CTRL_DPLL_PWR_SW_STS Register

31	30	29	28	27	26	25	24
RESERVED						PGOODOUT_DDR	PONOUT_DDR
R-0h						R-X	R-X
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED						PGOODOUT_DISP	PONOUT_DISP
R-0h						R-X	R-X
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED						PGOODOUT_PER	PONOUT_PER
R-0h						R-X	R-X
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PGOODOUT_MPU	PONOUT_MPU
R-0h						R-X	R-X

Table 7-25. CTRL_DPLL_PWR_SW_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25	PGOODOUT_DDR	R	X	PGOODOUT signals from DDR DPLL
24	PONOUT_DDR	R	X	PONOUT signal from DDR DPLL
23-18	RESERVED	R	0h	
17	PGOODOUT_DISP	R	X	PGOODOUT signal from DISP DPLL
16	PONOUT_DISP	R	X	PONOUT signal from DISP DPLL
15-10	RESERVED	R	0h	
9	PGOODOUT_PER	R	X	PGOODOUT signal from PER DPLL
8	PONOUT_PER	R	X	PONOUT signal from PER DPLL
7-2	RESERVED	R	0h	
1	PGOODOUT_MPU	R	X	PGOODOUT signal from MPU DPLL
0	PONOUT_MPU	R	X	PONOUT signal from MPU DPLL

7.3.1.15 CTRL_DISPLAY_PLL_SEL Register (Offset = 534h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_DISPLAY_PLL_SEL is shown in Figure 7-19 and described in Table 7-26.

Return to Summary Table.

Figure 7-19. CTRL_DISPLAY_PLL_SEL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="7">RESERVED</td><td>SELECT</td></tr>
<tr><td colspan="7">R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 7-26. CTRL_DISPLAY_PLL_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	SELECT	R/W	0h	0 : PRCM controls display PLL 1 : DISPPLL_* controls display PLL

7.3.1.16 CTRL_DEVICE_ID Register (Offset = 600h) [reset = X]

CTRL_DEVICE_ID is shown in Figure 7-20 and described in Table 7-27.

Return to Summary Table.

Figure 7-20. CTRL_DEVICE_ID Register

31	30	29	28	27	26	25	24
DEVREV				PARTNUM			
R-X				R-X			

23	22	21	20	19	18	17	16
PARTNUM							
R-X							

15	14	13	12	11	10	9	8
PARTNUM				MFGR			
R-X				R-X			

7	6	5	4	3	2	1	0
MFGR							ID_LSB
R-X							R-1h

Table 7-27. CTRL_DEVICE_ID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	DEVREV	R	X	Device revision
27-12	PARTNUM	R	X	Device part number (unique JTAG ID)
11-1	MFGR	R	X	Manufacturer's JTAG ID
0	ID_LSB	R	1h	Reserved - always 1

7.3.1.17 CTRL_DEV_FEATURE Register (Offset = 604h) [reset = 0h]

Register mask: D0008000h

CTRL_DEV_FEATURE is shown in Figure 7-21 and described in Table 7-28.

Return to Summary Table.

Figure 7-21. CTRL_DEV_FEATURE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEV_FEATURE_BITS																															
R-0h																															

Table 7-28. CTRL_DEV_FEATURE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEV_FEATURE_BITS	R	0h	Device-dependent. See Device Feature Comparison table in device data manual.

7.3.1.18 CTRL_INIT_PRIORITY_0 Register (Offset = 608h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_INIT_PRIORITY_0 is shown in Figure 7-22 and described in Table 7-29.

Return to Summary Table.

Figure 7-22. CTRL_INIT_PRIORITY_0 Register

31	30	29	28	27	26	25	24
RESERVED				TCWR2		TCRD2	
R/W-0h				R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
TCWR1		TCRD1		TCWR0		TCRD0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
P1500		RESERVED					
R/W-0h		R/W-0h					
7	6	5	4	3	2	1	0
RESERVED		PRU_ICSS1		PRU_ICSS0		HOST_ARM	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 7-29. CTRL_INIT_PRIORITY_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-26	TCWR2	R/W	0h	TPTC 2 Write Port initiator priority
25-24	TCRD2	R/W	0h	TPTC 2 Read Port initiator priority
23-22	TCWR1	R/W	0h	TPTC 1 Write Port initiator priority
21-20	TCRD1	R/W	0h	TPTC 1 Read Port initiator priority
19-18	TCWR0	R/W	0h	TPTC 1 Write Port initiator priority
17-16	TCRD0	R/W	0h	TPTC 1 Read Port initiator priority
15-14	P1500	R/W	0h	P1500 Port Initiator priority
13-6	RESERVED	R/W	0h	
5-4	PRU_ICSS1	R/W	0h	PRU-ICSS1 initiator priority
3-2	PRU_ICSS0	R/W	0h	Reserved
1-0	HOST_ARM	R/W	0h	Host ARM MPU initiator priority

7.3.1.19 CTRL_INIT_PRIORITY_1 Register (Offset = 60Ch) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_INIT_PRIORITY_1 is shown in Figure 7-23 and described in Table 7-30.

Return to Summary Table.

Figure 7-23. CTRL_INIT_PRIORITY_1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">DBG</td><td colspan="2">RESERVED</td><td colspan="2">SGX</td><td colspan="4">RESERVED</td></tr>
<tr><td colspan="6">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="2">VPFE1</td><td colspan="2">VPFE0</td><td colspan="4">RESERVED</td><td colspan="2">DSS</td><td colspan="2">CPSW</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 7-30. CTRL_INIT_PRIORITY_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R/W	0h	
25-24	DBG	R/W	0h	Debug Subsystem initiator priority
23-22	RESERVED	R/W	0h	
21-20	SGX	R/W	0h	SGX initiator priority
19-12	RESERVED	R/W	0h	
11-10	VPFE1	R/W	0h	VPFE1 initiator priority
9-8	VPFE0	R/W	0h	VPFE0 initiator priority
7-4	RESERVED	R/W	0h	
3-2	DSS	R/W	0h	DSS DMA port initiator priority
1-0	CPSW	R/W	0h	CPSW initiator priority

7.3.1.20 CTRL_DEV_ATTR Register (Offset = 610h) [reset = X]

CTRL_DEV_ATTR is shown in Figure 7-24 and described in Table 7-31.

Return to Summary Table.

Figure 7-24. CTRL_DEV_ATTR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	PACKAGE_TYPE		RESERVED	MPU_MAX_FREQ			
R-0h	R-X		R-0h	R-X			
7	6	5	4	3	2	1	0
MPU_MAX_FREQ							
R-X							

Table 7-31. CTRL_DEV_ATTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	X	Reserved. Reset value can vary
14-13	PACKAGE_TYPE	R	X	00b - ZDN 01b - Reserved 10b - Reserved 11b - Reserved
12	RESERVED	R	X	Reserved. Reset value can vary
11-0	MPU_MAX_FREQ	R	X	0xFFE 300MHz ARM MPU Maximum 0xFFA 600MHz ARM MPU Maximum 0xFE2 800MHz ARM MPU Maximum 0xFC2 1GHz ARM MPU Maximum

7.3.1.21 CTRL_TPTC_CFG Register (Offset = 614h) [reset = 3Fh]

Register mask: FFFFFFFFh

CTRL_TPTC_CFG is shown in Figure 7-25 and described in Table 7-32.

Return to Summary Table.

Figure 7-25. CTRL_TPTC_CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										TC2DBS		TC1DBS		TC0DBS	
R/W-0h										R/W-3h		R/W-3h		R/W-3h	

Table 7-32. CTRL_TPTC_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5-4	TC2DBS	R/W	3h	TC2 Default Burst Size 00 = 16 byte 01 = 32 byte 10 = 64 byte 11 = 128 byte
3-2	TC1DBS	R/W	3h	TC1 Default Burst Size
1-0	TC0DBS	R/W	3h	TC0 Default Burst Size

7.3.1.22 CTRL_USB_CTRL0 Register (Offset = 620h) [reset = 3C006007h]

Register mask: FFFFFFFFh

CTRL_USB_CTRL0 is shown in Figure 7-26 and described in Table 7-33.

Return to Summary Table.

Figure 7-26. CTRL_USB_CTRL0 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-3Ch							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DATAPOLARITY_INV	RESERVED	USB_WUEN	OTGSESSENDEN	OTGVDET_EN	DMGPIO_PD	DPGPIO_PD	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED	GPIO_SIG_CROSS	GPIO_SIG_INV	GPIOMODE	RESERVED	CDET_EXTCTL	DPPULLUP	DMPULLDN
R/W-0h	R/W-1h	R/W-1h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CHGVSRC_EN	CHGISINK_EN	SINKONDP	SRCONDM	CHGDET_RSTRT	CHGDET_DIS	OTG_PWRDN	CM_PWRDN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h	R/W-1h

Table 7-33. CTRL_USB_CTRL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	3Ch	Reserved. Any writes to this register must keep these bits set to 0x3C.
23	DATAPOLARITY_INV	R/W	0h	Data Polarity Invert: 0: DP/DM (normal polarity matching port definition) 1: DM/DP (inverted polarity of port definition)
22	RESERVED	R/W	0h	
21	USB_WUEN	R/W	0h	USB Wakeup Enable
20	OTGSESSENDEN	R/W	0h	Session End Detect Enable 0 : Disable Session End Comparator 1 : Turns on Session End Comparator
19	OTGVDET_EN	R/W	0h	VBUS Detect Enable 0 : Disable VBUS Detect Enable 1 : Turns on all comparators except Session End comparator
18	DMGPIO_PD	R/W	0h	Pull-down on DM in GPIO Mode 0 : Enables pull-down 1 : Disables pull-down
17	DPGPIO_PD	R/W	0h	Pull-down on DP in GPIO Mode 0 : Enables pull-down 1 : Disables pull-down
16-15	RESERVED	R/W	0h	
14	GPIO_SIG_CROSS	R/W	1h	UART TX -> DM. UART RX -> DP.
13	GPIO_SIG_INV	R/W	1h	UART TX -> Invert -> DP. UART RX -> Invert -> DM.
12	GPIOMODE	R/W	0h	GPIO Mode 0 : USB Mode 1 : GPIO Mode (UART Mode)
11	RESERVED	R/W	0h	
10	CDET_EXTCTL	R/W	0h	Bypass the charger detection state machine 0 : Charger detection on 1 : Charger detection is bypassed

Table 7-33. CTRL_USB_CTRL0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	DPPULLUP	R/W	0h	Pull-up on DP line 0 : No effect 1 : Enable pull-up on DP line
8	DMPULLDN	R/W	0h	Pull-down on DM line 0 : No effect 1 : Enable pull-down on DM line
7	CHGVSRC_EN	R/W	0h	Enable VSRC on DP line (Host Charger case)
6	CHGISINK_EN	R/W	0h	Enable ISINK on DM line (Host Charger case)
5	SINKONDP	R/W	0h	Sink on DP 0 : Sink on DM 1 : Sink on DP
4	SRCONDM	R/W	0h	Source on DM 0 : Source on DP 1 : Source on DM
3	CHGDET_RSTRT	R/W	0h	Restart Charger Detect
2	CHGDET_DIS	R/W	1h	Charger Detect Disable 0 : Enable 1 : Disable
1	OTG_PWRDN	R/W	1h	Power down the USB OTG PHY 1 : PHY Powered down 0 : PHY in normal mode
0	CM_PWRDN	R/W	1h	Power down the USB CM PHY 1 : PHY Powered down 0 : PHY in normal mode

7.3.1.23 CTRL_USB_STS0 Register (Offset = 624h) [reset = X]

CTRL_USB_STS0 is shown in Figure 7-27 and described in Table 7-34.

Return to Summary Table.

Figure 7-27. CTRL_USB_STS0 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							WUEVT
R-0h							R-X
7	6	5	4	3	2	1	0
CHGDETSTS			CDET_DMDET	CDET_DPDET	CDET_DATADET	CHGDETECT	CHGDETDONE
R-X			R-X	R-X	R-X	R-X	R-X

Table 7-34. CTRL_USB_STS0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	WUEVT	R	X	Wakeup Event
7-5	CHGDETSTS	R	X	Charge Detection Status 000: Wait State (When a D+WPU and D-15K are connected it enters into this state and will remain in this state unless it enters into other state) 001: No Contact 010: PS/2 011: Unknown error 100: Dedicated charger (valid if CE is HIGH) 101: HOST charger (valid if CE is HIGH) 110: PC 111: Interrupt (if any of the pullup is enabled charger detect routine gets interrupted and will restart from the beginning if the same is disabled)
4	CDET_DMDET	R	X	DM Comparator Output
3	CDET_DPDET	R	X	DP Comparator Output
2	CDET_DATADET	R	X	Charger Comparator Output
1	CHGDETECT	R	X	Charger Detection Status 0 : Charger was not detected 1 : Charger was detected
0	CHGDETDONE	R	X	Charger Detection Protocol Done

7.3.1.24 CTRL_USB_CTRL1 Register (Offset = 628h) [reset = 3C006007h]

Register mask: FFFFFFFFh

CTRL_USB_CTRL1 is shown in Figure 7-28 and described in Table 7-35.

Return to Summary Table.

Figure 7-28. CTRL_USB_CTRL1 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-3Ch							

23	22	21	20	19	18	17	16
DATAPOLARITY_INV	RESERVED	USB_WUEN	OTGSESSENDEN	OTGVDET_EN	DMGPIO_PD	DPGPIO_PD	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED	GPIO_SIG_CROSS	GPIO_SIG_INV	GPIOMODE	RESERVED	CDET_EXTCTL	DPPULLUP	DMPULLDN
R/W-0h	R/W-1h	R/W-1h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
CHGVSRC_EN	CHGISINK_EN	SINKONDP	SRCONDM	CHGDET_RSTRT	CHGDET_DIS	OTG_PWRDN	CM_PWRDN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h	R/W-1h

Table 7-35. CTRL_USB_CTRL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	3Ch	Reserved. Any writes to this register must keep these bits set to 0x3C.
23	DATAPOLARITY_INV	R/W	0h	Data Polarity Invert: 0: DP/DM (normal polarity matching port definition) 1: DM/DP (inverted polarity of port definition)
22	RESERVED	R/W	0h	
21	USB_WUEN	R/W	0h	USB Wakeup Enable
20	OTGSESSENDEN	R/W	0h	Session End Detect Enable 0 : Disable Session End Comparator 1 : Turns on Session End Comparator
19	OTGVDET_EN	R/W	0h	VBUS Detect Enable 0 : Disable VBUS Detect Enable 1 : Turns on all comparators except Session End comparator
18	DMGPIO_PD	R/W	0h	Pull-down on DM in GPIO Mode 0 : Enables pull-down 1 : Disables pull-down
17	DPGPIO_PD	R/W	0h	Pull-down on DP in GPIO Mode 0 : Enables pull-down 1 : Disables pull-down
16-15	RESERVED	R/W	0h	
14	GPIO_SIG_CROSS	R/W	1h	UART TX -> DM. UART RX -> DP.
13	GPIO_SIG_INV	R/W	1h	UART TX -> INV -> DP. UART RX -> INV -> DM.
12	GPIOMODE	R/W	0h	GPIO Mode 0 : USB Mode 1 : GPIO Mode (UART)
11	RESERVED	R/W	0h	
10	CDET_EXTCTL	R/W	0h	Bypass the charger detection state machine 0 : Charger detection on 1 : Charger detection is bypassed

Table 7-35. CTRL_USB_CTRL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	DPPULLUP	R/W	0h	Pull-up on DP line 0 : No effect 1 : Enable pull-up on DP line
8	DMPULLDN	R/W	0h	Pull-down on DM line 0 : No effect 1 : Enable pull-down on DM line
7	CHGVSRC_EN	R/W	0h	Enable VSRC on DP line (Host Charger case)
6	CHGISINK_EN	R/W	0h	Enable ISINK on DM line (Host Charger case)
5	SINKONDP	R/W	0h	Sink on DP 0 : Sink on DM 1 : Sink on DP
4	SRCONDM	R/W	0h	Source on DM 0 : Source on DP 1 : Source on DM
3	CHGDET_RSTRT	R/W	0h	Restart Charger Detect
2	CHGDET_DIS	R/W	1h	Charger Detect Disable 0 : Enable 1 : Disable
1	OTG_PWRDN	R/W	1h	Power down the USB OTG PHY 1 : PHY Powered down 0 : PHY in normal mode
0	CM_PWRDN	R/W	1h	Power down the USB CM PHY 1 : PHY Powered down 0 : PHY in normal mode

7.3.1.25 CTRL_USB_STS1 Register (Offset = 62Ch) [reset = X]

CTRL_USB_STS1 is shown in Figure 7-29 and described in Table 7-36.

Return to Summary Table.

Figure 7-29. CTRL_USB_STS1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							WUEVT
R-0h							R-X
7	6	5	4	3	2	1	0
CHGDETSTS			CDET_DMDET	CDET_DPDET	CDET_DATADET	CHGDETECT	CHGDETDONE
R-X			R-X	R-X	R-X	R-X	R-X

Table 7-36. CTRL_USB_STS1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	WUEVT	R	X	Wakeup Event
7-5	CHGDETSTS	R	X	Charge Detection Status 000: Wait State (When a D+WPU and D-15K are connected it enters into this state and will remain in this state unless it enters into other state) 001: No Contact 010: PS/2 011: Unknown error 100: Dedicated charger (valid if CE is HIGH) 101: HOST charger (valid if CE is HIGH) 110: PC 111: Interrupt (if any of the pullup is enabled charger detect routine gets interrupted and will restart from the beginning if the same is disabled)
4	CDET_DMDET	R	X	DM Comparator Output
3	CDET_DPDET	R	X	DP Comparator Output
2	CDET_DATADET	R	X	Charger Comparator Output
1	CHGDETECT	R	X	Charger Detection Status 0 : Charger was no detected 1 : Charger was detected
0	CHGDETDONE	R	X	Charger Detection Protocol Done

7.3.1.26 CTRL_MAC_ID0_LO Register (Offset = 630h) [reset = X]

CTRL_MAC_ID0_LO is shown in Figure 7-30 and described in Table 7-37.

Return to Summary Table.

Figure 7-30. CTRL_MAC_ID0_LO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MACADDR_47_40								MACADDR_39_32							
R-X								R-X							

Table 7-37. CTRL_MAC_ID0_LO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	MACADDR_47_40	R	X	MAC0 Address - Byte 5. Reset value is device-dependent.
7-0	MACADDR_39_32	R	X	MAC0 Address - Byte 4. Reset value is device-dependent.

7.3.1.27 CTRL_MAC_ID0_HI Register (Offset = 634h) [reset = X]

CTRL_MAC_ID0_HI is shown in Figure 7-31 and described in Table 7-38.

Return to Summary Table.

Figure 7-31. CTRL_MAC_ID0_HI Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MACADDR_31_24								MACADDR_23_16							
R-X								R-X							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MACADDR_15_8								MACADDR_7_0							
R-X								R-X							

Table 7-38. CTRL_MAC_ID0_HI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MACADDR_31_24	R	X	MAC0 Address - Byte 3. Reset value is device-dependent.
23-16	MACADDR_23_16	R	X	MAC0 Address - Byte 2. Reset value is device-dependent.
15-8	MACADDR_15_8	R	X	MAC0 Address - Byte 1. Reset value is device-dependent.
7-0	MACADDR_7_0	R	X	MAC0 Address - Byte 0. Reset value is device-dependent.

7.3.1.28 CTRL_MAC_ID1_LO Register (Offset = 638h) [reset = X]

CTRL_MAC_ID1_LO is shown in Figure 7-32 and described in Table 7-39.

Return to Summary Table.

Figure 7-32. CTRL_MAC_ID1_LO Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">MACADDR_47_40</td><td colspan="8">MACADDR_39_32</td></tr>
<tr><td colspan="8">R-X</td><td colspan="8">R-X</td></tr>
</table>

Table 7-39. CTRL_MAC_ID1_LO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	MACADDR_47_40	R	X	MAC1 Address - Byte 5. Reset value is device-dependent.
7-0	MACADDR_39_32	R	X	MAC1 Address - Byte 4. Reset value is device-dependent.

7.3.1.29 CTRL_MAC_ID1_HI Register (Offset = 63Ch) [reset = X]

CTRL_MAC_ID1_HI is shown in Figure 7-33 and described in Table 7-40.

Return to Summary Table.

Figure 7-33. CTRL_MAC_ID1_HI Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MACADDR_31_24								MACADDR_23_16							
R-X								R-X							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MACADDR_15_8								MACADDR_7_0							
R-X								R-X							

Table 7-40. CTRL_MAC_ID1_HI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MACADDR_31_24	R	X	MAC1 Address - Byte 3. Reset value is device-dependent.
23-16	MACADDR_23_16	R	X	MAC1 Address - Byte 2. Reset value is device-dependent.
15-8	MACADDR_15_8	R	X	MAC1 Address - Byte 1. Reset value is device-dependent.
7-0	MACADDR_7_0	R	X	MAC1 Address - Byte 0. Reset value is device-dependent.

7.3.1.30 CTRL_DCAN_RAMINIT Register (Offset = 644h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_DCAN_RAMINIT is shown in Figure 7-34 and described in Table 7-41.

Return to Summary Table.

Figure 7-34. CTRL_DCAN_RAMINIT Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED						DCAN1_RAMINIT_DONE	DCAN0_RAMINIT_DONE
R/W-0h						W1C-0h	W1C-0h
7	6	5	4	3	2	1	0
RESERVED						DCAN1_RAMINIT_START	DCAN0_RAMINIT_START
R/W-0h						R/W-0h	R/W-0h

Table 7-41. CTRL_DCAN_RAMINIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	0h	
9	DCAN1_RAMINIT_DONE	W1C	0h	1: DCAN1 RAM Initialization complete 0: DCAN1 RAM Initialization NOT complete
8	DCAN0_RAMINIT_DONE	W1C	0h	1: DCAN0 RAM Initialization complete 0: DCAN0 RAM Initialization NOT complete
7-2	RESERVED	R/W	0h	
1	DCAN1_RAMINIT_START	R/W	0h	A transition from 0 to 1 will start DCAN1 RAM initialization sequence.
0	DCAN0_RAMINIT_START	R/W	0h	A transition from 0 to 1 will start DCAN0 RAM initialization sequence.

7.3.1.31 CTRL_USB_CTRL2 Register (Offset = 64Ch) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_USB_CTRL2 is shown in Figure 7-35 and described in Table 7-42.

Return to Summary Table.

Figure 7-35. CTRL_USB_CTRL2 Register

31	30	29	28	27	26	25	24
RESERVED					PHY1_FILTER_THR_VBUSVALID		PHY1_FILTER_THR_AVALID
R/W-0h					R/W-0h		R/W-0h
23	22	21	20	19	18	17	16
PHY1_FILTER_THR_AVALID	PHY1_FILTER_THR_BVALID		PHY1_FILTER_THR_SESSEND		PHY1_FILTER_THR_IDDIG		PHY1_FILTER_BYPASS
R/W-0h	R/W-0h		R/W-0h		R/W-0h		R/W-0h
15	14	13	12	11	10	9	8
RESERVED					PHY0_FILTER_THR_VBUSVALID		PHY0_FILTER_THR_AVALID
R/W-0h					R/W-0h		R/W-0h
7	6	5	4	3	2	1	0
PHY0_FILTER_THR_AVALID	PHY0_FILTER_THR_BVALID		PHY0_FILTER_THR_SESSEND		PHY0_FILTER_THR_IDDIG		PHY0_FILTER_BYPASS
R/W-0h	R/W-0h		R/W-0h		R/W-0h		R/W-0h

Table 7-42. CTRL_USB_CTRL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-25	PHY1_FILTER_THR_VBUSVALID	R/W	0h	PHY1_FILTER_THRESHOLD for VBUSVALID to fliter out oscillations from PHY1. 0 = 1us 1 = 100us 2 = 5ms 3 = 500ms
24-23	PHY1_FILTER_THR_AVALID	R/W	0h	PHY1_FILTER_THRESHOLD for AVALID to fliter out oscillations from PHY1. 0 = 1us 1 = 100us 2 = 5ms 3 = 500ms
22-21	PHY1_FILTER_THR_BVALID	R/W	0h	PHY1_FILTER_THRESHOLD for AVALID to fliter out oscillations from PHY1. 0 = 1us 1 = 100us 2 = 5ms 3 = 500ms
20-19	PHY1_FILTER_THR_SESSEND	R/W	0h	PHY1_FILTER_THRESHOLD for AVALID to fliter out oscillations from PHY1. 0 = 1us 1 = 100us 2 = 5ms 3 = 500ms
18-17	PHY1_FILTER_THR_IDDIG	R/W	0h	PHY1_FILTER_THRESHOLD for AVALID to fliter out oscillations from PHY1. 0 = 1us 1 = 100us 2 = 5ms 3 = 500ms
16	PHY1_FILTER_BYPASS	R/W	0h	0 = Filter to filter oscillations from PHY1 is not bypassed 1 = Filter is bypassed
15-11	RESERVED	R/W	0h	

Table 7-42. CTRL_USB_CTRL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-9	PHY0_FILTER_THR_VBUSVALID	R/W	0h	PHY0_FILTER_THRESHOLD for VBUSVALID to fliter out oscillations from PHY0. 0 = 1us 1 = 100us 2 = 5ms 3 = 500ms
8-7	PHY0_FILTER_THR_AVALID	R/W	0h	PHY0_FILTER_THRESHOLD for AVALID to fliter out oscillations from PHY0. 0 = 1us 1 = 100us 2 = 5ms 3 = 500ms
6-5	PHY0_FILTER_THR_BVALID	R/W	0h	PHY0_FILTER_THRESHOLD for AVALID to fliter out oscillations from PHY0. 0 = 1us 1 = 100us 2 = 5ms 3 = 500ms
4-3	PHY0_FILTER_THR_SESSEND	R/W	0h	PHY0_FILTER_THRESHOLD for AVALID to fliter out oscillations from PHY0. 0 = 1us 1 = 100us 2 = 5ms 3 = 500ms
2-1	PHY0_FILTER_THR_IDDIG	R/W	0h	PHY0_FILTER_THRESHOLD for AVALID to fliter out oscillations from PHY0. 0 = 1us 1 = 100us 2 = 5ms 3 = 500ms
0	PHY0_FILTER_BYPASS	R/W	0h	0 - Filter to filter oscillations from PHY0 is not bypassed 1 - Filter is bypassed

7.3.1.32 CTRL_GMII_SEL Register (Offset = 650h) [reset = C0h]

Register mask: FFFFFFFFh

CTRL_GMII_SEL is shown in Figure 7-36 and described in Table 7-43.

Return to Summary Table.

Figure 7-36. CTRL_GMII_SEL Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RMII2_IO_CLK_EN	RMII1_IO_CLK_EN	RGMII2_IDMODE	RGMII1_IDMODE	GMII2_SEL		GMII1_SEL	
R/W-1h	R/W-1h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	

Table 7-43. CTRL_GMII_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7	RMII2_IO_CLK_EN	R/W	1h	1 : Enable RMII clock to be sourced from chip pin 0 : Enable RMII clock to be sourced from PLL
6	RMII1_IO_CLK_EN	R/W	1h	1 : Enable RMII clock to be sourced from chip pin 0 : Enable RMII clock to be sourced from PLL
5	RGMII2_IDMODE	R/W	0h	RGMII2 Internal Delay Mode 0 : Internal Delay 1 : No Internal Delay
4	RGMII1_IDMODE	R/W	0h	RGMII1 Internal Delay Mode 0 : Internal Delay 1 : No Internal Delay
3-2	GMII2_SEL	R/W	0h	00 : Port2 GMII/MII Mode 01 : Port2 RMII Mode 10 : Port2 RGMII Mode 11 : Not Used
1-0	GMII1_SEL	R/W	0h	00 : Port1 GMII/MII Mode 01 : Port1 RMII Mode 10 : Port1 RGMII Mode 11 : Not Used

7.3.1.33 CTRL_MPUSS Register (Offset = 654h) [reset = X]

CTRL_MPUSS is shown in Figure 7-37 and described in Table 7-44.

Return to Summary Table.

Figure 7-37. CTRL_MPUSS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="5">RESERVED</td><td colspan="3">HSDCC</td></tr>
<tr><td colspan="5">R/W-0h</td><td colspan="3">R/W-X</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">HSDCC</td><td colspan="2">PIL2SRAMCLKDIV</td><td>RESERVED</td></tr>
<tr><td colspan="5">R/W-X</td><td colspan="2">R/W-2h</td><td>R/W-0h</td></tr>
</table>

Table 7-44. CTRL_MPUSS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-3	HSDCC	R/W	X	(Bit 3): CLK (Bits 6 to 4): ADDR (Bits 10 to 7): DATA
2-1	PIL2SRAMCLKDIV	R/W	2h	00 : MPU_CLK/2 01 : MPU_CLK/3 10 : MPU_CLK/4 11 : MPU_CLK/6
0	RESERVED	R/W	0h	

7.3.1.34 CTRL_TIMER_CASCADE Register (Offset = 658h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_TIMER_CASCADE is shown in Figure 7-38 and described in Table 7-45.

Return to Summary Table.

Figure 7-38. CTRL_TIMER_CASCADE Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						TIMER45CASCADE_EN	TIMER23CASCADE_EN
R/W-0h						R/W-0h	R/W-0h

Table 7-45. CTRL_TIMER_CASCADE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	TIMER45CASCADE_EN	R/W	0h	1 - Enables cascading of timer4 & timer5
0	TIMER23CASCADE_EN	R/W	0h	1 - Enables cascading of timer2 & timer3

7.3.1.35 CTRL_PWMSS Register (Offset = 664h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_PWMSS is shown in Figure 7-39 and described in Table 7-46.

Return to Summary Table.

Figure 7-39. CTRL_PWMSS Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	PWMSS5_TBCLKEN	PWMMS4_TBCLKEN	PWMSS3_TBCLKEN	PWM_SYNCSEL	PWMSS2_TBCLKEN	PWMMS1_TBCLKEN	PWMSS0_TBCLKEN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 7-46. CTRL_PWMSS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R/W	0h	
6	PWMSS5_TBCLKEN	R/W	0h	Timebase clock enable for PWMSS5
5	PWMMS4_TBCLKEN	R/W	0h	Timebase clock enable for PWMSS4
4	PWMSS3_TBCLKEN	R/W	0h	Timebase clock enable for PWMSS3
3	PWM_SYNCSEL	R/W	0h	0: PWM3 sync is from PWM2 (daisy chained) 1: PWM3 sync is from PIN (eHRPWM3)
2	PWMSS2_TBCLKEN	R/W	0h	Timebase clock enable for PWMSS2
1	PWMMS1_TBCLKEN	R/W	0h	Timebase clock enable for PWMSS1
0	PWMSS0_TBCLKEN	R/W	0h	Timebase clock enable for PWMSS0

7.3.1.36 CTRL_MREQPRIO_0 Register (Offset = 670h) [reset = 44444444h]

Register mask: FFFFFFFFh

CTRL_MREQPRIO_0 is shown in Figure 7-40 and described in Table 7-47.

Return to Summary Table.

Figure 7-40. CTRL_MREQPRIO_0 Register

31	30	29	28	27	26	25	24
RESERVED	SGX			RESERVED	USB1		
R/W-0h	R/W-4h			R/W-0h	R/W-4h		
23	22	21	20	19	18	17	16
RESERVED	USB0			RESERVED	CPSW		
R/W-0h	R/W-4h			R/W-0h	R/W-4h		
15	14	13	12	11	10	9	8
RESERVED	PRU_ICSS1_PRU1			RESERVED	PRU_ICSS1_PRU0		
R/W-0h	R/W-4h			R/W-0h	R/W-4h		
7	6	5	4	3	2	1	0
RESERVED	SAB_INIT1			RESERVED	SAB_INIT0		
R/W-0h	R/W-4h			R/W-0h	R/W-4h		

Table 7-47. CTRL_MREQPRIO_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30-28	SGX	R/W	4h	MReqPriority for SGX Initiator OCP Interface
27	RESERVED	R/W	0h	
26-24	USB1	R/W	4h	Reserved : AXI does not support
23	RESERVED	R/W	0h	
22-20	USB0	R/W	4h	Reserved : AXI does not support
19	RESERVED	R/W	0h	
18-16	CPSW	R/W	4h	MReqPriority for CPSW Initiator OCP Interface
15	RESERVED	R/W	0h	
14-12	PRU_ICSS1_PRU1	R/W	4h	MReqPriority for PRU-ICSS1 PRU1Initiator OCP Interface
11	RESERVED	R/W	0h	
10-8	PRU_ICSS1_PRU0	R/W	4h	MReqPriority for PRU-ICSS1 PRU0 Initiator OCP Interface
7	RESERVED	R/W	0h	
6-4	SAB_INIT1	R/W	4h	MReqPriority for MPUSS Initiator 1 OCP Interface
3	RESERVED	R/W	0h	
2-0	SAB_INIT0	R/W	4h	This is reserved and not used since the MPUSS IP itself supports this signal.

7.3.1.37 CTRL_MREQPRIO_1 Register (Offset = 674h) [reset = 00444440h]

Register mask: FFFFFFFFh

CTRL_MREQPRIO_1 is shown in Figure 7-41 and described in Table 7-48.

Return to Summary Table.

Figure 7-41. CTRL_MREQPRIO_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-88h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	VPFE1			RESERVED	VPFE0			RESERVED	DSS			RESERVED			
R/W-88h	R/W-4h			R/W-0h	R/W-4h			R/W-0h	R/W-4h			R/W-0h			

Table 7-48. CTRL_MREQPRIO_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W	88h	
14-12	VPFE1	R/W	4h	MReqPriority for VPFE1 OCP Interface
11	RESERVED	R/W	0h	
10-8	VPFE0	R/W	4h	MReqPriority for VPFE0 OCP Interface
7	RESERVED	R/W	0h	
6-4	DSS	R/W	4h	MReqPriority for DSS OCP Interface
3-0	RESERVED	R/W	0h	

7.3.1.38 CTRL_VDD_MPU_OPP_050 Register (Offset = 770h) [reset = X]

CTRL_VDD_MPU_OPP_050 is shown in Figure 7-42 and described in Table 7-49.

Return to Summary Table.

Figure 7-42. CTRL_VDD_MPU_OPP_050 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								NTARGET																							
R-X								R-X																							

Table 7-49. CTRL_VDD_MPU_OPP_050 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	X	
23-0	NTARGET	R	X	Ntarget value for MPU Voltage domain OPP50. Reset value is device-dependent.

7.3.1.39 CTRL_VDD_MPU_OPP_100 Register (Offset = 774h) [reset = X]

CTRL_VDD_MPU_OPP_100 is shown in Figure 7-43 and described in Table 7-50.

Return to Summary Table.

Figure 7-43. CTRL_VDD_MPU_OPP_100 Register

31-24	23-0
RESERVED	NTARGET
R-X	R-X

Table 7-50. CTRL_VDD_MPU_OPP_100 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	X	
23-0	NTARGET	R	X	Ntarget value for MPU Voltage domain OPP100. Reset value is device-dependent.

7.3.1.40 CTRL_VDD_MPU_OPP_120 Register (Offset = 778h) [reset = X]

CTRL_VDD_MPU_OPP_120 is shown in Figure 7-44 and described in Table 7-51.

Return to Summary Table.

Figure 7-44. CTRL_VDD_MPU_OPP_120 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="24">NTARGET</td></tr>
<tr><td colspan="8">R-X</td><td colspan="24">R-X</td></tr>
</table>

Table 7-51. CTRL_VDD_MPU_OPP_120 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	X	
23-0	NTARGET	R	X	Ntarget value for MPU Voltage domain OPP120. Reset value is device-dependent.

7.3.1.41 CTRL_VDD_MPU_OPP_TURBO Register (Offset = 77Ch) [reset = X]

CTRL_VDD_MPU_OPP_TURBO is shown in Figure 7-45 and described in Table 7-52.

Return to Summary Table.

Figure 7-45. CTRL_VDD_MPU_OPP_TURBO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								NTARGET																							
R-X								R-X																							

Table 7-52. CTRL_VDD_MPU_OPP_TURBO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	X	
23-0	NTARGET	R	X	Ntarget value for MPU Voltage domain OPPTURBO. Reset value is device-dependent.

7.3.1.42 CTRL_VDD_MPU_OPP_NITRO Register (Offset = 780h) [reset = X]

CTRL_VDD_MPU_OPP_NITRO is shown in Figure 7-46 and described in Table 7-53.

Return to Summary Table.

Figure 7-46. CTRL_VDD_MPU_OPP_NITRO Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="24">NTARGET</td></tr>
<tr><td colspan="8">R-X</td><td colspan="24">R-X</td></tr>
</table>

Table 7-53. CTRL_VDD_MPU_OPP_NITRO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	X	
23-0	NTARGET	R	X	Ntarget value for MPU Voltage domain OPPNITRO. Reset value is device-dependent.

7.3.1.43 CTRL_VDD_CORE_OPP_050 Register (Offset = 7B8h) [reset = X]

CTRL_VDD_CORE_OPP_050 is shown in Figure 7-47 and described in Table 7-54.

Return to Summary Table.

Figure 7-47. CTRL_VDD_CORE_OPP_050 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="24">NTARGET</td></tr>
<tr><td colspan="8">R-X</td><td colspan="24">R-X</td></tr>
</table>

Table 7-54. CTRL_VDD_CORE_OPP_050 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	X	
23-0	NTARGET	R	X	Ntarget value for CORE Voltage domain OPP50. Reset value is device-dependent.

7.3.1.44 CTRL_VDD_CORE_OPP_100 Register (Offset = 7BCh) [reset = X]

CTRL_VDD_CORE_OPP_100 is shown in Figure 7-48 and described in Table 7-55.

Return to Summary Table.

Figure 7-48. CTRL_VDD_CORE_OPP_100 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="24">NTARGET</td></tr>
<tr><td colspan="8">R-X</td><td colspan="24">R-X</td></tr>
</table>

Table 7-55. CTRL_VDD_CORE_OPP_100 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	X	
23-0	NTARGET	R	X	Ntarget value for CORE Voltage domain OPP100. Reset value is device-dependent.

7.3.1.45 CTRL_USB_VID_PID Register (Offset = 7F4h) [reset = X]

CTRL_USB_VID_PID is shown in Figure 7-49 and described in Table 7-56.

Return to Summary Table.

Figure 7-49. CTRL_USB_VID_PID Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">USB_VID</td><td colspan="16">USB_PID</td></tr>
<tr><td colspan="16">R-X</td><td colspan="16">R-X</td></tr>
</table>

Table 7-56. CTRL_USB_VID_PID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	USB_VID	R	X	USB Vendor ID
15-0	USB_PID	R	X	USB Product ID

7.3.1.46 CTRL_CONF_GPMC_AD0 Register (Offset = 800h) [reset = X]

CTRL_CONF_GPMC_AD0 is shown in Figure 7-50 and described in Table 7-57.

Return to Summary Table.

Figure 7-50. CTRL_CONF_GPMC_AD0 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_AD0_WUEVT	CONF_GPMC_AD0_WUEN	CONF_GPMC_AD0_DSPULLTYPESELECT	CONF_GPMC_AD0_DSPULLUDEN	CONF_GPMC_AD0_DS0OUTVALUE	CONF_GPMC_AD0_DS0OUTEN	CONF_GPMC_AD0_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPMC_AD0_SLEWCTRL	CONF_GPMC_AD0_RXACTIVE	CONF_GPMC_AD0_PUTYPESEL	CONF_GPMC_AD0_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPMC_AD0_MMODE			
R/W-0h				R/W-0h			

Table 7-57. CTRL_CONF_GPMC_AD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_AD0_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_AD0_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_AD0_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_AD0_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_AD0_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_AD0_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_AD0_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_AD0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_AD0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_AD0_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-57. CTRL_CONF_GPMC_AD0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_AD0_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_AD0_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.47 CTRL_CONF_GPMC_AD1 Register (Offset = 804h) [reset = X]

CTRL_CONF_GPMC_AD1 is shown in Figure 7-51 and described in Table 7-58.

Return to Summary Table.

Figure 7-51. CTRL_CONF_GPMC_AD1 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_AD1_WUEVT	CONF_GPMC_AD1_WUEN	CONF_GPMC_AD1_DSPULLTYPESELECT	CONF_GPMC_AD1_DSPULLUDEN	CONF_GPMC_AD1_DS0OUTVALUE	CONF_GPMC_AD1_DS0OUTEN	CONF_GPMC_AD1_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_AD1_SLEWCTRL	CONF_GPMC_AD1_RXACTIVE	CONF_GPMC_AD1_PUTYPESEL	CONF_GPMC_AD1_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_AD1_MMODE			
R/W-0h				R/W-0h			

Table 7-58. CTRL_CONF_GPMC_AD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_AD1_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_AD1_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_AD1_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_AD1_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_AD1_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_AD1_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_AD1_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_AD1_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_AD1_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_AD1_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-58. CTRL_CONF_GPMC_AD1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_AD1_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_AD1_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.48 CTRL_CONF_GPMC_AD2 Register (Offset = 808h) [reset = X]

CTRL_CONF_GPMC_AD2 is shown in Figure 7-52 and described in Table 7-59.

Return to Summary Table.

Figure 7-52. CTRL_CONF_GPMC_AD2 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_AD2_WUEVT	CONF_GPMC_AD2_WUEN	CONF_GPMC_AD2_DSPULLTYPESELECT	CONF_GPMC_AD2_DSPULLUDEN	CONF_GPMC_AD2_DS0OUTVALUE	CONF_GPMC_AD2_DS0OUTEN	CONF_GPMC_AD2_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPMC_AD2_SLEWCTRL	CONF_GPMC_AD2_RXACTIVE	CONF_GPMC_AD2_PUTYPESEL	CONF_GPMC_AD2_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPMC_AD2_MMODE			
R/W-0h				R/W-0h			

Table 7-59. CTRL_CONF_GPMC_AD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_AD2_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_AD2_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_AD2_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_AD2_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_AD2_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_AD2_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_AD2_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_AD2_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_AD2_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_AD2_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-59. CTRL_CONF_GPMC_AD2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_AD2_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_AD2_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.49 CTRL_CONF_GPMC_AD3 Register (Offset = 80Ch) [reset = X]

CTRL_CONF_GPMC_AD3 is shown in Figure 7-53 and described in Table 7-60.

Return to Summary Table.

Figure 7-53. CTRL_CONF_GPMC_AD3 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_AD3_WUEVT	CONF_GPMC_AD3_WUEN	CONF_GPMC_AD3_DSPULLTYPESELECT	CONF_GPMC_AD3_DSPULLUDEN	CONF_GPMC_AD3_DS0OUTVALUE	CONF_GPMC_AD3_DS0OUTEN	CONF_GPMC_AD3_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_AD3_SLEWCTRL	CONF_GPMC_AD3_RXACTIVE	CONF_GPMC_AD3_PUTYPESEL	CONF_GPMC_AD3_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_AD3_MMODE			
R/W-0h				R/W-0h			

Table 7-60. CTRL_CONF_GPMC_AD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_AD3_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_AD3_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_AD3_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_AD3_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_AD3_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_AD3_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_AD3_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_AD3_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_AD3_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_AD3_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-60. CTRL_CONF_GPMC_AD3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_AD3_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_AD3_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.50 CTRL_CONF_GPMC_AD4 Register (Offset = 810h) [reset = X]

CTRL_CONF_GPMC_AD4 is shown in Figure 7-54 and described in Table 7-61.

Return to Summary Table.

Figure 7-54. CTRL_CONF_GPMC_AD4 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_AD4_WUEVT	CONF_GPMC_AD4_WUEN	CONF_GPMC_AD4_DSPULLTYPESELECT	CONF_GPMC_AD4_DSPULLUDEN	CONF_GPMC_AD4_DS0OUTVALUE	CONF_GPMC_AD4_DS0OUTEN	CONF_GPMC_AD4_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPMC_AD4_SLEWCTRL	CONF_GPMC_AD4_RXACTIVE	CONF_GPMC_AD4_PUTYPESEL	CONF_GPMC_AD4_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPMC_AD4_MMODE			
R/W-0h				R/W-0h			

Table 7-61. CTRL_CONF_GPMC_AD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_AD4_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_AD4_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_AD4_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_AD4_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_AD4_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_AD4_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_AD4_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_AD4_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_AD4_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_AD4_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-61. CTRL_CONF_GPMC_AD4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_AD4_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_AD4_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.51 CTRL_CONF_GPMC_AD5 Register (Offset = 814h) [reset = X]

CTRL_CONF_GPMC_AD5 is shown in Figure 7-55 and described in Table 7-62.

Return to Summary Table.

Figure 7-55. CTRL_CONF_GPMC_AD5 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_AD5_WUEVT	CONF_GPMC_AD5_WUEN	CONF_GPMC_AD5_DSPULLTYPESELECT	CONF_GPMC_AD5_DSPULLUDEN	CONF_GPMC_AD5_DS0OUTVALUE	CONF_GPMC_AD5_DS0OUTEN	CONF_GPMC_AD5_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPMC_AD5_SLEWCTRL	CONF_GPMC_AD5_RXACTIVE	CONF_GPMC_AD5_PUTYPESEL	CONF_GPMC_AD5_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPMC_AD5_MMODE			
R/W-0h				R/W-0h			

Table 7-62. CTRL_CONF_GPMC_AD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_AD5_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_AD5_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_AD5_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_AD5_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_AD5_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_AD5_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_AD5_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_AD5_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_AD5_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_AD5_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-62. CTRL_CONF_GPMC_AD5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_AD5_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_AD5_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.52 CTRL_CONF_GPMC_AD6 Register (Offset = 818h) [reset = X]

CTRL_CONF_GPMC_AD6 is shown in Figure 7-56 and described in Table 7-63.

Return to Summary Table.

Figure 7-56. CTRL_CONF_GPMC_AD6 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_AD6_WUEVT	CONF_GPMC_AD6_WUEN	CONF_GPMC_AD6_DSPULLTYPESELECT	CONF_GPMC_AD6_DSPULLUDEN	CONF_GPMC_AD6_DS0OUTVALUE	CONF_GPMC_AD6_DS0OUTEN	CONF_GPMC_AD6_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPMC_AD6_SLEWCTRL	CONF_GPMC_AD6_RXACTIVE	CONF_GPMC_AD6_PUTYPESEL	CONF_GPMC_AD6_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPMC_AD6_MMODE			
R/W-0h				R/W-0h			

Table 7-63. CTRL_CONF_GPMC_AD6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_AD6_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_AD6_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_AD6_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_AD6_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_AD6_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_AD6_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_AD6_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_AD6_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_AD6_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_AD6_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-63. CTRL_CONF_GPMC_AD6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_AD6_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_AD6_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.53 CTRL_CONF_GPMC_AD7 Register (Offset = 81Ch) [reset = X]

CTRL_CONF_GPMC_AD7 is shown in Figure 7-57 and described in Table 7-64.

Return to Summary Table.

Figure 7-57. CTRL_CONF_GPMC_AD7 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_AD7_WUEVT	CONF_GPMC_AD7_WUEN	CONF_GPMC_AD7_DSPULLTYPESELECT	CONF_GPMC_AD7_DSPULLUDEN	CONF_GPMC_AD7_DS0OUTVALUE	CONF_GPMC_AD7_DS0OUTEN	CONF_GPMC_AD7_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_AD7_SLEWCTRL	CONF_GPMC_AD7_RXACTIVE	CONF_GPMC_AD7_PUTYPESEL	CONF_GPMC_AD7_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_AD7_MMODE			
R/W-0h				R/W-0h			

Table 7-64. CTRL_CONF_GPMC_AD7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_AD7_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_AD7_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_AD7_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_AD7_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_AD7_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_AD7_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_AD7_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_AD7_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_AD7_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_AD7_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-64. CTRL_CONF_GPMC_AD7 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_AD7_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_AD7_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.54 CTRL_CONF_GPMC_AD8 Register (Offset = 820h) [reset = X]

CTRL_CONF_GPMC_AD8 is shown in Figure 7-58 and described in Table 7-65.

Return to Summary Table.

Figure 7-58. CTRL_CONF_GPMC_AD8 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_AD8_WUEVT	CONF_GPMC_AD8_WUEN	CONF_GPMC_AD8_DSPULLTYPESELECT	CONF_GPMC_AD8_DSPULLUDEN	CONF_GPMC_AD8_DS0OUTVALUE	CONF_GPMC_AD8_DS0OUTEN	CONF_GPMC_AD8_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPMC_AD8_SLEWCTRL	CONF_GPMC_AD8_RXACTIVE	CONF_GPMC_AD8_PUTYPESEL	CONF_GPMC_AD8_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPMC_AD8_MMODE			
R/W-0h				R/W-0h			

Table 7-65. CTRL_CONF_GPMC_AD8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_AD8_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_AD8_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_AD8_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_AD8_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_AD8_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_AD8_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_AD8_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_AD8_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_AD8_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_AD8_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-65. CTRL_CONF_GPMC_AD8 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_AD8_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_AD8_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.55 CTRL_CONF_GPMC_AD9 Register (Offset = 824h) [reset = X]

CTRL_CONF_GPMC_AD9 is shown in Figure 7-59 and described in Table 7-66.

Return to Summary Table.

Figure 7-59. CTRL_CONF_GPMC_AD9 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_AD9_WUEVT	CONF_GPMC_AD9_WUEN	CONF_GPMC_AD9_DSPULLTYPESELECT	CONF_GPMC_AD9_DSPULLUDEN	CONF_GPMC_AD9_DS0OUTVALUE	CONF_GPMC_AD9_DS0OUTEN	CONF_GPMC_AD9_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_AD9_SLEWCTRL	CONF_GPMC_AD9_RXACTIVE	CONF_GPMC_AD9_PUTYPESEL	CONF_GPMC_AD9_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_AD9_MMODE			
R/W-0h				R/W-0h			

Table 7-66. CTRL_CONF_GPMC_AD9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_AD9_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_AD9_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_AD9_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_AD9_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_AD9_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_AD9_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_AD9_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_AD9_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_AD9_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_AD9_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-66. CTRL_CONF_GPMC_AD9 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_AD9_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_AD9_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.56 CTRL_CONF_GPMC_AD10 Register (Offset = 828h) [reset = X]

CTRL_CONF_GPMC_AD10 is shown in Figure 7-60 and described in Table 7-67.

Return to Summary Table.

Figure 7-60. CTRL_CONF_GPMC_AD10 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_AD10_WUEVT	CONF_GPMC_AD10_WUEN	CONF_GPMC_AD10_DSPULLTYPESELECT	CONF_GPMC_AD10_DSPULLUDEN	CONF_GPMC_AD10_DS0OUTVALUE	CONF_GPMC_AD10_DS0OUTEN	CONF_GPMC_AD10_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_AD10_SLEWCTRL	CONF_GPMC_AD10_RXACTIVE	CONF_GPMC_AD10_PUTYPESEL	CONF_GPMC_AD10_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_AD10_MMODE			
R/W-0h				R/W-0h			

Table 7-67. CTRL_CONF_GPMC_AD10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_AD10_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_AD10_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_AD10_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_AD10_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_AD10_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_AD10_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_AD10_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_AD10_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_AD10_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_AD10_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-67. CTRL_CONF_GPMC_AD10 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_AD10_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_AD10_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.57 CTRL_CONF_GPMC_AD11 Register (Offset = 82Ch) [reset = X]

CTRL_CONF_GPMC_AD11 is shown in Figure 7-61 and described in Table 7-68.

Return to Summary Table.

Figure 7-61. CTRL_CONF_GPMC_AD11 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_AD11_WUEVT	CONF_GPMC_AD11_WUEN	CONF_GPMC_AD11_DSPULLTYPESELECT	CONF_GPMC_AD11_DSPULLUDEN	CONF_GPMC_AD11_DS0OUTVALUE	CONF_GPMC_AD11_DS0OUTEN	CONF_GPMC_AD11_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_AD11_SLEWCTRL	CONF_GPMC_AD11_RXACTIVE	CONF_GPMC_AD11_PUTYPESEL	CONF_GPMC_AD11_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_AD11_MMODE			
R/W-0h				R/W-0h			

Table 7-68. CTRL_CONF_GPMC_AD11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_AD11_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_AD11_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_AD11_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_AD11_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_AD11_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_AD11_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_AD11_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_AD11_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_AD11_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_AD11_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-68. CTRL_CONF_GPMC_AD11 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_AD11_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_AD11_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.58 CTRL_CONF_GPMC_AD12 Register (Offset = 830h) [reset = X]

CTRL_CONF_GPMC_AD12 is shown in Figure 7-62 and described in Table 7-69.

Return to Summary Table.

Figure 7-62. CTRL_CONF_GPMC_AD12 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>RESERVED</td><td>CONF_GPMC_AD12_WUEVT</td><td>CONF_GPMC_AD12_WUEN</td><td>CONF_GPMC_AD12_DSPULLTYPESELECT</td><td>CONF_GPMC_AD12_DSPULLUDEN</td><td>CONF_GPMC_AD12_DS0OUTVALUE</td><td>CONF_GPMC_AD12_DS0OUTEN</td><td>CONF_GPMC_AD12_DS0EN</td></tr>
<tr><td>R/W-0h</td><td>R-X</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-1h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td>CONF_GPMC_AD12_SLEWCTRL</td><td>CONF_GPMC_AD12_RXACTIVE</td><td>CONF_GPMC_AD12_PUTYPESEL</td><td>CONF_GPMC_AD12_PUDEN</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R/W-0h</td><td>R/W-1h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">CONF_GPMC_AD12_MMODE</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 7-69. CTRL_CONF_GPMC_AD12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_AD12_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_AD12_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_AD12_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_AD12_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_AD12_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_AD12_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_AD12_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_AD12_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_AD12_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_AD12_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-69. CTRL_CONF_GPMC_AD12 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_AD12_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_AD12_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.59 CTRL_CONF_GPMC_AD13 Register (Offset = 834h) [reset = X]

CTRL_CONF_GPMC_AD13 is shown in Figure 7-63 and described in Table 7-70.

Return to Summary Table.

Figure 7-63. CTRL_CONF_GPMC_AD13 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_AD13_WUEVT	CONF_GPMC_AD13_WUEN	CONF_GPMC_AD13_DSPULLTYPESELECT	CONF_GPMC_AD13_DSPULLUDEN	CONF_GPMC_AD13_DS0OUTVALUE	CONF_GPMC_AD13_DS0OUTEN	CONF_GPMC_AD13_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_AD13_SLEWCTRL	CONF_GPMC_AD13_RXACTIVE	CONF_GPMC_AD13_PUTYPESEL	CONF_GPMC_AD13_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_AD13_MMODE			
R/W-0h				R/W-0h			

Table 7-70. CTRL_CONF_GPMC_AD13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_AD13_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_AD13_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_AD13_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_AD13_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_AD13_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_AD13_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_AD13_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_AD13_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_AD13_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_AD13_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-70. CTRL_CONF_GPMC_AD13 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_AD13_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_AD13_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.60 CTRL_CONF_GPMC_AD14 Register (Offset = 838h) [reset = X]

CTRL_CONF_GPMC_AD14 is shown in Figure 7-64 and described in Table 7-71.

Return to Summary Table.

Figure 7-64. CTRL_CONF_GPMC_AD14 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_AD14_WUEVT	CONF_GPMC_AD14_WUEN	CONF_GPMC_AD14_DSPULLTYPESELECT	CONF_GPMC_AD14_DSPULLUDEN	CONF_GPMC_AD14_DS0OUTVALUE	CONF_GPMC_AD14_DS0OUTEN	CONF_GPMC_AD14_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_AD14_SLEWCTRL	CONF_GPMC_AD14_RXACTIVE	CONF_GPMC_AD14_PUTYPESEL	CONF_GPMC_AD14_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_AD14_MMODE			
R/W-0h				R/W-0h			

Table 7-71. CTRL_CONF_GPMC_AD14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_AD14_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_AD14_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_AD14_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_AD14_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_AD14_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_AD14_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_AD14_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_AD14_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_AD14_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_AD14_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-71. CTRL_CONF_GPMC_AD14 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_AD14_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_AD14_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.61 CTRL_CONF_GPMC_AD15 Register (Offset = 83Ch) [reset = X]

CTRL_CONF_GPMC_AD15 is shown in Figure 7-65 and described in Table 7-72.

Return to Summary Table.

Figure 7-65. CTRL_CONF_GPMC_AD15 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_AD15_WUEVT	CONF_GPMC_AD15_WUEN	CONF_GPMC_AD15_DSPULLTYPESELECT	CONF_GPMC_AD15_DSPULLUDEN	CONF_GPMC_AD15_DS0OUTVALUE	CONF_GPMC_AD15_DS0OUTEN	CONF_GPMC_AD15_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_AD15_SLEWCTRL	CONF_GPMC_AD15_RXACTIVE	CONF_GPMC_AD15_PUTYPESEL	CONF_GPMC_AD15_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_AD15_MMODE			
R/W-0h				R/W-0h			

Table 7-72. CTRL_CONF_GPMC_AD15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_AD15_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_AD15_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_AD15_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_AD15_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_AD15_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_AD15_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_AD15_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_AD15_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_AD15_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_AD15_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-72. CTRL_CONF_GPMC_AD15 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_AD15_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_AD15_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.62 CTRL_CONF_GPMC_A0 Register (Offset = 840h) [reset = X]

CTRL_CONF_GPMC_A0 is shown in Figure 7-66 and described in Table 7-73.

Return to Summary Table.

Figure 7-66. CTRL_CONF_GPMC_A0 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_A0_WUEVT	CONF_GPMC_A0_WUEN	CONF_GPMC_A0_DSPULLTYPESELECT	CONF_GPMC_A0_DSPULLUDEN	CONF_GPMC_A0_DS0OUTVALUE	CONF_GPMC_A0_DS0OUTEN	CONF_GPMC_A0_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPMC_A0_SLEWCTRL	CONF_GPMC_A0_RXACTIVE	CONF_GPMC_A0_PUTYPESEL	CONF_GPMC_A0_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPMC_A0_MMODE			
R/W-0h				R/W-0h			

Table 7-73. CTRL_CONF_GPMC_A0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_A0_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_A0_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_A0_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_A0_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_A0_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_A0_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_A0_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_A0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_A0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_A0_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-73. CTRL_CONF_GPMC_A0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_A0_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_A0_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.63 CTRL_CONF_GPMC_A1 Register (Offset = 844h) [reset = X]

CTRL_CONF_GPMC_A1 is shown in Figure 7-67 and described in Table 7-74.

Return to Summary Table.

Figure 7-67. CTRL_CONF_GPMC_A1 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_A1_WUEVT	CONF_GPMC_A1_WUEN	CONF_GPMC_A1_DSPULLTYPESELECT	CONF_GPMC_A1_DSPULLUDEN	CONF_GPMC_A1_DS0OUTVALUE	CONF_GPMC_A1_DS0OUTEN	CONF_GPMC_A1_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_A1_SLEWCTRL	CONF_GPMC_A1_RXACTIVE	CONF_GPMC_A1_PUTYPESEL	CONF_GPMC_A1_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_A1_MMODE			
R/W-0h				R/W-0h			

Table 7-74. CTRL_CONF_GPMC_A1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_A1_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_A1_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_A1_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_A1_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_A1_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_A1_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_A1_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_A1_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_A1_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_A1_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-74. CTRL_CONF_GPMC_A1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_A1_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_A1_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.64 CTRL_CONF_GPMC_A2 Register (Offset = 848h) [reset = X]

CTRL_CONF_GPMC_A2 is shown in Figure 7-68 and described in Table 7-75.

Return to Summary Table.

Figure 7-68. CTRL_CONF_GPMC_A2 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_A2_WUEVT	CONF_GPMC_A2_WUEN	CONF_GPMC_A2_DSPULLTYPESELECT	CONF_GPMC_A2_DSPULLUDEN	CONF_GPMC_A2_DS0OUTVALUE	CONF_GPMC_A2_DS0OUTEN	CONF_GPMC_A2_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPMC_A2_SLEWCTRL	CONF_GPMC_A2_RXACTIVE	CONF_GPMC_A2_PUTYPESEL	CONF_GPMC_A2_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPMC_A2_MMODE			
R/W-0h				R/W-0h			

Table 7-75. CTRL_CONF_GPMC_A2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_A2_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_A2_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_A2_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_A2_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_A2_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_A2_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_A2_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_A2_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_A2_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_A2_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-75. CTRL_CONF_GPMC_A2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_A2_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_A2_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.65 CTRL_CONF_GPMC_A3 Register (Offset = 84Ch) [reset = X]

CTRL_CONF_GPMC_A3 is shown in Figure 7-69 and described in Table 7-76.

Return to Summary Table.

Figure 7-69. CTRL_CONF_GPMC_A3 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_A3_WUEVT	CONF_GPMC_A3_WUEN	CONF_GPMC_A3_DSPULLTYPESELECT	CONF_GPMC_A3_DSPULLUDEN	CONF_GPMC_A3_DS0OUTVALUE	CONF_GPMC_A3_DS0OUTEN	CONF_GPMC_A3_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_A3_SLEWCTRL	CONF_GPMC_A3_RXACTIVE	CONF_GPMC_A3_PUTYPESEL	CONF_GPMC_A3_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_A3_MMODE			
R/W-0h				R/W-0h			

Table 7-76. CTRL_CONF_GPMC_A3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_A3_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_A3_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_A3_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_A3_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_A3_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_A3_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_A3_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_A3_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_A3_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_A3_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-76. CTRL_CONF_GPMC_A3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_A3_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_A3_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.66 CTRL_CONF_GPMC_A4 Register (Offset = 850h) [reset = X]

CTRL_CONF_GPMC_A4 is shown in Figure 7-70 and described in Table 7-77.

Return to Summary Table.

Figure 7-70. CTRL_CONF_GPMC_A4 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_A4_WUEVT	CONF_GPMC_A4_WUEN	CONF_GPMC_A4_DSPULLTYPESELECT	CONF_GPMC_A4_DSPULLUDEN	CONF_GPMC_A4_DS0OUTVALUE	CONF_GPMC_A4_DS0OUTEN	CONF_GPMC_A4_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_A4_SLEWCTRL	CONF_GPMC_A4_RXACTIVE	CONF_GPMC_A4_PUTYPESEL	CONF_GPMC_A4_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_A4_MMODE			
R/W-0h				R/W-0h			

Table 7-77. CTRL_CONF_GPMC_A4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_A4_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_A4_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_A4_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_A4_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_A4_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_A4_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_A4_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_A4_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_A4_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_A4_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-77. CTRL_CONF_GPMC_A4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_A4_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_A4_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.67 CTRL_CONF_GPMC_A5 Register (Offset = 854h) [reset = X]

CTRL_CONF_GPMC_A5 is shown in Figure 7-71 and described in Table 7-78.

Return to Summary Table.

Figure 7-71. CTRL_CONF_GPMC_A5 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_A5_WUEVT	CONF_GPMC_A5_WUEN	CONF_GPMC_A5_DSPULLTYPESELECT	CONF_GPMC_A5_DSPULLUDEN	CONF_GPMC_A5_DS0OUTVALUE	CONF_GPMC_A5_DS0OUTEN	CONF_GPMC_A5_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_A5_SLEWCTRL	CONF_GPMC_A5_RXACTIVE	CONF_GPMC_A5_PUTYPESEL	CONF_GPMC_A5_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_A5_MMODE			
R/W-0h				R/W-0h			

Table 7-78. CTRL_CONF_GPMC_A5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_A5_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_A5_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_A5_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_A5_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_A5_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_A5_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_A5_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_A5_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_A5_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_A5_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-78. CTRL_CONF_GPMC_A5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_A5_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_A5_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.68 CTRL_CONF_GPMC_A6 Register (Offset = 858h) [reset = X]

CTRL_CONF_GPMC_A6 is shown in Figure 7-72 and described in Table 7-79.

Return to Summary Table.

Figure 7-72. CTRL_CONF_GPMC_A6 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_A6_WUEVT	CONF_GPMC_A6_WUEN	CONF_GPMC_A6_DSPULLTYPESELECT	CONF_GPMC_A6_DSPULLUDEN	CONF_GPMC_A6_DS0OUTVALUE	CONF_GPMC_A6_DS0OUTEN	CONF_GPMC_A6_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPMC_A6_SLEWCTRL	CONF_GPMC_A6_RXACTIVE	CONF_GPMC_A6_PUTYPESEL	CONF_GPMC_A6_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPMC_A6_MMODE			
R/W-0h				R/W-0h			

Table 7-79. CTRL_CONF_GPMC_A6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_A6_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_A6_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_A6_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_A6_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_A6_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_A6_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_A6_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_A6_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_A6_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_A6_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-79. CTRL_CONF_GPMC_A6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_A6_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_A6_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.69 CTRL_CONF_GPMC_A7 Register (Offset = 85Ch) [reset = X]

CTRL_CONF_GPMC_A7 is shown in Figure 7-73 and described in Table 7-80.

Return to Summary Table.

Figure 7-73. CTRL_CONF_GPMC_A7 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_A7_WUEVT	CONF_GPMC_A7_WUEN	CONF_GPMC_A7_DSPULLTYPESELECT	CONF_GPMC_A7_DSPULLUDEN	CONF_GPMC_A7_DS0OUTVALUE	CONF_GPMC_A7_DS0OUTEN	CONF_GPMC_A7_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPMC_A7_SLEWCTRL	CONF_GPMC_A7_RXACTIVE	CONF_GPMC_A7_PUTYPESEL	CONF_GPMC_A7_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPMC_A7_MMODE			
R/W-0h				R/W-0h			

Table 7-80. CTRL_CONF_GPMC_A7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_A7_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_A7_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_A7_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_A7_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_A7_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_A7_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_A7_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_A7_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_A7_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_A7_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-80. CTRL_CONF_GPMC_A7 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_A7_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_A7_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.70 CTRL_CONF_GPMC_A8 Register (Offset = 860h) [reset = X]

CTRL_CONF_GPMC_A8 is shown in Figure 7-74 and described in Table 7-81.

Return to Summary Table.

Figure 7-74. CTRL_CONF_GPMC_A8 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_A8_WUEVT	CONF_GPMC_A8_WUEN	CONF_GPMC_A8_DSPULLTYPESELECT	CONF_GPMC_A8_DSPULLUDEN	CONF_GPMC_A8_DS0OUTVALUE	CONF_GPMC_A8_DS0OUTEN	CONF_GPMC_A8_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_A8_SLEWCTRL	CONF_GPMC_A8_RXACTIVE	CONF_GPMC_A8_PUTYPESEL	CONF_GPMC_A8_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_A8_MMODE			
R/W-0h				R/W-0h			

Table 7-81. CTRL_CONF_GPMC_A8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_A8_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_A8_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_A8_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_A8_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_A8_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_A8_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_A8_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_A8_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_A8_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_A8_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-81. CTRL_CONF_GPMC_A8 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_A8_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_A8_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.71 CTRL_CONF_GPMC_A9 Register (Offset = 864h) [reset = X]

CTRL_CONF_GPMC_A9 is shown in Figure 7-75 and described in Table 7-82.

Return to Summary Table.

Figure 7-75. CTRL_CONF_GPMC_A9 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_A9_WUEVT	CONF_GPMC_A9_WUEN	CONF_GPMC_A9_DSPULLTYPESELECT	CONF_GPMC_A9_DSPULLUDEN	CONF_GPMC_A9_DS0OUTVALUE	CONF_GPMC_A9_DS0OUTEN	CONF_GPMC_A9_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPMC_A9_SLEWCTRL	CONF_GPMC_A9_RXACTIVE	CONF_GPMC_A9_PUTYPESEL	CONF_GPMC_A9_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPMC_A9_MMODE			
R/W-0h				R/W-0h			

Table 7-82. CTRL_CONF_GPMC_A9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_A9_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_A9_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_A9_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_A9_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_A9_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_A9_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_A9_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_A9_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_A9_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_A9_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-82. CTRL_CONF_GPMC_A9 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_A9_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_A9_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.72 CTRL_CONF_GPMC_A10 Register (Offset = 868h) [reset = X]

CTRL_CONF_GPMC_A10 is shown in Figure 7-76 and described in Table 7-83.

Return to Summary Table.

Figure 7-76. CTRL_CONF_GPMC_A10 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_A10_WUEVT	CONF_GPMC_A10_WUEN	CONF_GPMC_A10_DSPULLTYPESELECT	CONF_GPMC_A10_DSPULLUDEN	CONF_GPMC_A10_DS0OUTVALUE	CONF_GPMC_A10_DS0OUTEN	CONF_GPMC_A10_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_A10_SLEWCTRL	CONF_GPMC_A10_RXACTIVE	CONF_GPMC_A10_PUTYPESEL	CONF_GPMC_A10_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_A10_MMODE			
R/W-0h				R/W-0h			

Table 7-83. CTRL_CONF_GPMC_A10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_A10_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_A10_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_A10_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_A10_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_A10_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_A10_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_A10_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_A10_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_A10_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_A10_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-83. CTRL_CONF_GPMC_A10 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_A10_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_A10_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.73 CTRL_CONF_GPMC_A11 Register (Offset = 86Ch) [reset = X]

CTRL_CONF_GPMC_A11 is shown in Figure 7-77 and described in Table 7-84.

Return to Summary Table.

Figure 7-77. CTRL_CONF_GPMC_A11 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_A11_WUEVT	CONF_GPMC_A11_WUEN	CONF_GPMC_A11_DSPULLTYPESELECT	CONF_GPMC_A11_DSPULLUDEN	CONF_GPMC_A11_DS0OUTVALUE	CONF_GPMC_A11_DS0OUTEN	CONF_GPMC_A11_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPMC_A11_SLEWCTRL	CONF_GPMC_A11_RXACTIVE	CONF_GPMC_A11_PUTYPESEL	CONF_GPMC_A11_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPMC_A11_MMODE			
R/W-0h				R/W-0h			

Table 7-84. CTRL_CONF_GPMC_A11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_A11_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_A11_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_A11_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_A11_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_A11_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_A11_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_A11_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_A11_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_A11_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_A11_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-84. CTRL_CONF_GPMC_A11 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_A11_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_A11_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.74 CTRL_CONF_GPMC_WAIT0 Register (Offset = 870h) [reset = X]

CTRL_CONF_GPMC_WAIT0 is shown in Figure 7-78 and described in Table 7-85.

Return to Summary Table.

Figure 7-78. CTRL_CONF_GPMC_WAIT0 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_WAIT0_WUEVT	CONF_GPMC_WAIT0_WUEN	CONF_GPMC_WAIT0_DSPULLTYPESELECT	CONF_GPMC_WAIT0_DSPULLUDEN	CONF_GPMC_WAIT0_DS0OUTVALUE	CONF_GPMC_WAIT0_DS0OUTEN	CONF_GPMC_WAIT0_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_WAIT0_SLEWCTRL	CONF_GPMC_WAIT0_RXACTIVE	CONF_GPMC_WAIT0_PUTYPESEL	CONF_GPMC_WAIT0_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_WAIT0_MMODE			
R/W-0h				R/W-0h			

Table 7-85. CTRL_CONF_GPMC_WAIT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_WAIT0_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_WAIT0_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_WAIT0_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_WAIT0_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_WAIT0_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_WAIT0_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_WAIT0_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_WAIT0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_WAIT0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_WAIT0_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-85. CTRL_CONF_GPMC_WAIT0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_WAIT0_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_WAIT0_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.75 CTRL_CONF_GPMC_WPN Register (Offset = 874h) [reset = X]

CTRL_CONF_GPMC_WPN is shown in Figure 7-79 and described in Table 7-86.

Return to Summary Table.

Figure 7-79. CTRL_CONF_GPMC_WPN Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_WPN_WUEVT	CONF_GPMC_WPN_WUEN	CONF_GPMC_WPN_DSPULLTYPESELECT	CONF_GPMC_WPN_DSPULLUDEN	CONF_GPMC_WPN_DS0OUTVALUE	CONF_GPMC_WPN_DS0OUTEN	CONF_GPMC_WPN_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPMC_WPN_SLEWCTRL	CONF_GPMC_WPN_RXACTIVE	CONF_GPMC_WPN_PUTYPESEL	CONF_GPMC_WPN_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPMC_WPN_MMODE			
R/W-0h				R/W-0h			

Table 7-86. CTRL_CONF_GPMC_WPN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_WPN_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_WPN_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_WPN_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_WPN_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_WPN_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_WPN_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_WPN_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_WPN_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_WPN_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_WPN_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-86. CTRL_CONF_GPMC_WPN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_WPN_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_WPN_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.76 CTRL_CONF_GPMC_BE1N Register (Offset = 878h) [reset = X]

CTRL_CONF_GPMC_BE1N is shown in Figure 7-80 and described in Table 7-87.

Return to Summary Table.

Figure 7-80. CTRL_CONF_GPMC_BE1N Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_BE1N_WUEVT	CONF_GPMC_BE1N_WUEN	CONF_GPMC_BE1N_DSPULLTYPESELECT	CONF_GPMC_BE1N_DSPULLUDEN	CONF_GPMC_BE1N_DS0OUTVALUE	CONF_GPMC_BE1N_DS0OUTEN	CONF_GPMC_BE1N_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_BE1N_SLEWCTRL	CONF_GPMC_BE1N_RXACTIVE	CONF_GPMC_BE1N_PUTYPESEL	CONF_GPMC_BE1N_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_BE1N_MMODE			
R/W-0h				R/W-0h			

Table 7-87. CTRL_CONF_GPMC_BE1N Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_BE1N_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_BE1N_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_BE1N_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_BE1N_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_BE1N_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_BE1N_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_BE1N_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_BE1N_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_BE1N_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_BE1N_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-87. CTRL_CONF_GPMC_BE1N Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_BE1N_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_BE1N_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.77 CTRL_CONF_GPMC_CSN0 Register (Offset = 87Ch) [reset = X]

CTRL_CONF_GPMC_CSN0 is shown in Figure 7-81 and described in Table 7-88.

Return to Summary Table.

Figure 7-81. CTRL_CONF_GPMC_CSN0 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_CSN0_WUEVT	CONF_GPMC_CSN0_WUEN	CONF_GPMC_CSN0_DSPULLTYPESELECT	CONF_GPMC_CSN0_DSPULLUDEN	CONF_GPMC_CSN0_DS0OUTVALUE	CONF_GPMC_CSN0_DS0OUTEN	CONF_GPMC_CSN0_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPMC_CSN0_SLEWCTRL	CONF_GPMC_CSN0_RXACTIVE	CONF_GPMC_CSN0_PUTYPESEL	CONF_GPMC_CSN0_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPMC_CSN0_MMODE			
R/W-0h				R/W-0h			

Table 7-88. CTRL_CONF_GPMC_CSN0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_CSN0_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_CSN0_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_CSN0_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_CSN0_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_CSN0_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_CSN0_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_CSN0_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_CSN0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_CSN0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_CSN0_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-88. CTRL_CONF_GPMC_CSN0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_CSN0_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_CSN0_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.78 CTRL_CONF_GPMC_CSN1 Register (Offset = 880h) [reset = X]

CTRL_CONF_GPMC_CSN1 is shown in Figure 7-82 and described in Table 7-89.

Return to Summary Table.

Figure 7-82. CTRL_CONF_GPMC_CSN1 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_CSN1_WUEVT	CONF_GPMC_CSN1_WUEN	CONF_GPMC_CSN1_DSPULLTYPESELECT	CONF_GPMC_CSN1_DSPULLUDEN	CONF_GPMC_CSN1_DS0OUTVALUE	CONF_GPMC_CSN1_DS0OUTEN	CONF_GPMC_CSN1_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPMC_CSN1_SLEWCTRL	CONF_GPMC_CSN1_RXACTIVE	CONF_GPMC_CSN1_PUTYPESEL	CONF_GPMC_CSN1_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPMC_CSN1_MMODE			
R/W-0h				R/W-0h			

Table 7-89. CTRL_CONF_GPMC_CSN1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_CSN1_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_CSN1_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_CSN1_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_CSN1_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_CSN1_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_CSN1_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_CSN1_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_CSN1_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_CSN1_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_CSN1_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-89. CTRL_CONF_GPMC_CSN1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_CSN1_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_CSN1_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.79 CTRL_CONF_GPMC_CSN2 Register (Offset = 884h) [reset = X]

CTRL_CONF_GPMC_CSN2 is shown in Figure 7-83 and described in Table 7-90.

Return to Summary Table.

Figure 7-83. CTRL_CONF_GPMC_CSN2 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_CSN2_WUEVT	CONF_GPMC_CSN2_WUEN	CONF_GPMC_CSN2_DSPULLTYPESELECT	CONF_GPMC_CSN2_DSPULLUDEN	CONF_GPMC_CSN2_DS0OUTVALUE	CONF_GPMC_CSN2_DS0OUTEN	CONF_GPMC_CSN2_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPMC_CSN2_SLEWCTRL	CONF_GPMC_CSN2_RXACTIVE	CONF_GPMC_CSN2_PUTYPESEL	CONF_GPMC_CSN2_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPMC_CSN2_MMODE			
R/W-0h				R/W-0h			

Table 7-90. CTRL_CONF_GPMC_CSN2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_CSN2_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_CSN2_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_CSN2_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_CSN2_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_CSN2_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_CSN2_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_CSN2_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_CSN2_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_CSN2_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_CSN2_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-90. CTRL_CONF_GPMC_CSN2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_CSN2_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_CSN2_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.80 CTRL_CONF_GPMC_CSN3 Register (Offset = 888h) [reset = X]

CTRL_CONF_GPMC_CSN3 is shown in Figure 7-84 and described in Table 7-91.

Return to Summary Table.

Figure 7-84. CTRL_CONF_GPMC_CSN3 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_CSN3_WUEVT	CONF_GPMC_CSN3_WUEN	CONF_GPMC_CSN3_DSPULLTYPESELECT	CONF_GPMC_CSN3_DSPULLUDEN	CONF_GPMC_CSN3_DS0OUTVALUE	CONF_GPMC_CSN3_DS0OUTEN	CONF_GPMC_CSN3_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_CSN3_SLEWCTRL	CONF_GPMC_CSN3_RXACTIVE	CONF_GPMC_CSN3_PUTYPESEL	CONF_GPMC_CSN3_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_CSN3_MMODE			
R/W-0h				R/W-0h			

Table 7-91. CTRL_CONF_GPMC_CSN3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_CSN3_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_CSN3_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_CSN3_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_CSN3_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_CSN3_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_CSN3_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_CSN3_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_CSN3_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_CSN3_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_CSN3_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-91. CTRL_CONF_GPMC_CSN3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_CSN3_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_CSN3_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.81 CTRL_CONF_GPMC_CLK Register (Offset = 88Ch) [reset = X]

CTRL_CONF_GPMC_CLK is shown in Figure 7-85 and described in Table 7-92.

Return to Summary Table.

Figure 7-85. CTRL_CONF_GPMC_CLK Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_CLK_WUEVT	CONF_GPMC_CLK_WUEN	CONF_GPMC_CLK_DSPULLTYPESELECT	CONF_GPMC_CLK_DSPULLUDEN	CONF_GPMC_CLK_DS0OUTVALUE	CONF_GPMC_CLK_DS0OUTEN	CONF_GPMC_CLK_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_CLK_SLEWCTRL	CONF_GPMC_CLK_RXACTIVE	CONF_GPMC_CLK_PUTYPESEL	CONF_GPMC_CLK_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_CLK_MMODE			
R/W-0h				R/W-0h			

Table 7-92. CTRL_CONF_GPMC_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_CLK_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_CLK_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_CLK_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_CLK_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_CLK_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_CLK_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_CLK_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_CLK_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_CLK_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_CLK_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-92. CTRL_CONF_GPMC_CLK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_CLK_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_CLK_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.82 CTRL_CONF_GPMC_ADVN_ALE Register (Offset = 890h) [reset = X]

CTRL_CONF_GPMC_ADVN_ALE is shown in Figure 7-86 and described in Table 7-93.

Return to Summary Table.

Figure 7-86. CTRL_CONF_GPMC_ADVN_ALE Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_ADVN_ALE_WUEVT	CONF_GPMC_ADVN_ALE_WUEN	CONF_GPMC_ADVN_ALE_DSPULLTYPESELECT	CONF_GPMC_ADVN_ALE_DSPULLUDEN	CONF_GPMC_ADVN_ALE_DS0OUTVALUE	CONF_GPMC_ADVN_ALE_DS0OUTEN	CONF_GPMC_ADVN_ALE_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_ADVN_ALE_SLEWCTRL	CONF_GPMC_ADVN_ALE_RXACTIVE	CONF_GPMC_ADVN_ALE_PUTYPESEL	CONF_GPMC_ADVN_ALE_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_ADVN_ALE_MMODE			
R/W-0h				R/W-0h			

Table 7-93. CTRL_CONF_GPMC_ADVN_ALE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_ADVN_ALE_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_ADVN_ALE_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_ADVN_ALE_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_ADVN_ALE_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_ADVN_ALE_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_ADVN_ALE_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_ADVN_ALE_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_ADVN_ALE_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_ADVN_ALE_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_ADVN_ALE_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-93. CTRL_CONF_GPMC_ADVN_ALE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_ADVN_ALE_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_ADVN_ALE_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.83 CTRL_CONF_GPMC_OEN_REN Register (Offset = 894h) [reset = X]

CTRL_CONF_GPMC_OEN_REN is shown in Figure 7-87 and described in Table 7-94.

Return to Summary Table.

Figure 7-87. CTRL_CONF_GPMC_OEN_REN Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_OEN_REN_WUEVT	CONF_GPMC_OEN_REN_WUEN	CONF_GPMC_OEN_REN_DSPULLTYPESELECT	CONF_GPMC_OEN_REN_DSPULLUDEN	CONF_GPMC_OEN_REN_DS0OUTVALUE	CONF_GPMC_OEN_REN_DS0OUTEN	CONF_GPMC_OEN_REN_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_OEN_REN_SLEWCTRL	CONF_GPMC_OEN_REN_RXACTIVE	CONF_GPMC_OEN_REN_PUTYPESEL	CONF_GPMC_OEN_REN_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_OEN_REN_MMODE			
R/W-0h				R/W-0h			

Table 7-94. CTRL_CONF_GPMC_OEN_REN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_OEN_REN_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_OEN_REN_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_OEN_REN_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_OEN_REN_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_OEN_REN_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_OEN_REN_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_OEN_REN_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_OEN_REN_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_OEN_REN_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_OEN_REN_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-94. CTRL_CONF_GPMC_OEN_REN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_OEN_REN_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_OEN_REN_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.84 CTRL_CONF_GPMC_WEN Register (Offset = 898h) [reset = X]

CTRL_CONF_GPMC_WEN is shown in Figure 7-88 and described in Table 7-95.

Return to Summary Table.

Figure 7-88. CTRL_CONF_GPMC_WEN Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_WEN_WUEVT	CONF_GPMC_WEN_WUEN	CONF_GPMC_WEN_DSPULLTYPESELECT	CONF_GPMC_WEN_DSPULLUDEN	CONF_GPMC_WEN_DS0OUTVALUE	CONF_GPMC_WEN_DS0OUTEN	CONF_GPMC_WEN_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPMC_WEN_SLEWCTRL	CONF_GPMC_WEN_RXACTIVE	CONF_GPMC_WEN_PUTYPESEL	CONF_GPMC_WEN_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPMC_WEN_MMODE			
R/W-0h				R/W-0h			

Table 7-95. CTRL_CONF_GPMC_WEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_WEN_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_WEN_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_WEN_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_WEN_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_WEN_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_WEN_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_WEN_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_WEN_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_WEN_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_WEN_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-95. CTRL_CONF_GPMC_WEN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_WEN_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_WEN_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.85 CTRL_CONF_GPMC_BE0N_CLE Register (Offset = 89Ch) [reset = X]

CTRL_CONF_GPMC_BE0N_CLE is shown in Figure 7-89 and described in Table 7-96.

Return to Summary Table.

Figure 7-89. CTRL_CONF_GPMC_BE0N_CLE Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPMC_BE0N_CLE_WUEVT	CONF_GPMC_BE0N_CLE_WUEN	CONF_GPMC_BE0N_CLE_DSPULLTYPESELECT	CONF_GPMC_BE0N_CLE_DSPULLUDEN	CONF_GPMC_BE0N_CLE_DS0OUTVALUE	CONF_GPMC_BE0N_CLE_DS0OUTEN	CONF_GPMC_BE0N_CLE_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPMC_BE0N_CLE_SLEWCTRL	CONF_GPMC_BE0N_CLE_RXACTIVE	CONF_GPMC_BE0N_CLE_PUTYPESEL	CONF_GPMC_BE0N_CLE_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPMC_BE0N_CLE_MMODE			
R/W-0h				R/W-0h			

Table 7-96. CTRL_CONF_GPMC_BE0N_CLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPMC_BE0N_CLE_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPMC_BE0N_CLE_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPMC_BE0N_CLE_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPMC_BE0N_CLE_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPMC_BE0N_CLE_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPMC_BE0N_CLE_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPMC_BE0N_CLE_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPMC_BE0N_CLE_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPMC_BE0N_CLE_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPMC_BE0N_CLE_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-96. CTRL_CONF_GPMC_BE0N_CLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPMC_BE0N_CLE_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPMC_BE0N_CLE_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.86 CTRL_CONF_DSS_DATA0 Register (Offset = 8A0h) [reset = X]

CTRL_CONF_DSS_DATA0 is shown in Figure 7-90 and described in Table 7-97.

Return to Summary Table.

Figure 7-90. CTRL_CONF_DSS_DATA0 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_DSS_DATA0_WUEVT	CONF_DSS_DATA0_WUEN	CONF_DSS_DATA0_DSPULLTYPESELECT	CONF_DSS_DATA0_DSPULLUDEN	CONF_DSS_DATA0_DS0OUTVALUE	CONF_DSS_DATA0_DS0OUTEN	CONF_DSS_DATA0_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_DSS_DATA0_SLEWCTRL	CONF_DSS_DATA0_RXACTIVE	CONF_DSS_DATA0_PUTYPESEL	CONF_DSS_DATA0_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_DSS_DATA0_MMODE			
R/W-0h				R/W-0h			

Table 7-97. CTRL_CONF_DSS_DATA0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_DATA0_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_DATA0_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_DATA0_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_DATA0_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_DATA0_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_DATA0_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_DATA0_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_DATA0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_DATA0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_DATA0_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-97. CTRL_CONF_DSS_DATA0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_DATA0_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_DATA0_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.87 CTRL_CONF_DSS_DATA1 Register (Offset = 8A4h) [reset = X]

CTRL_CONF_DSS_DATA1 is shown in Figure 7-91 and described in Table 7-98.

Return to Summary Table.

Figure 7-91. CTRL_CONF_DSS_DATA1 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_DSS_DATA1_WUEVT	CONF_DSS_DATA1_WUEN	CONF_DSS_DATA1_DSPULLTYPESELECT	CONF_DSS_DATA1_DSPULLUDEN	CONF_DSS_DATA1_DS0OUTVALUE	CONF_DSS_DATA1_DS0OUTEN	CONF_DSS_DATA1_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_DSS_DATA1_SLEWCTRL	CONF_DSS_DATA1_RXACTIVE	CONF_DSS_DATA1_PUTYPESEL	CONF_DSS_DATA1_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_DSS_DATA1_MMODE			
R/W-0h				R/W-0h			

Table 7-98. CTRL_CONF_DSS_DATA1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_DATA1_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_DATA1_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_DATA1_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_DATA1_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_DATA1_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_DATA1_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_DATA1_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_DATA1_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_DATA1_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_DATA1_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-98. CTRL_CONF_DSS_DATA1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_DATA1_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_DATA1_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.88 CTRL_CONF_DSS_DATA2 Register (Offset = 8A8h) [reset = X]

CTRL_CONF_DSS_DATA2 is shown in Figure 7-92 and described in Table 7-99.

Return to Summary Table.

Figure 7-92. CTRL_CONF_DSS_DATA2 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_DSS_DATA2_WUEVT	CONF_DSS_DATA2_WUEN	CONF_DSS_DATA2_DSPULLTYPESELECT	CONF_DSS_DATA2_DSPULLUDEN	CONF_DSS_DATA2_DS0OUTVALUE	CONF_DSS_DATA2_DS0OUTEN	CONF_DSS_DATA2_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_DSS_DATA2_SLEWCTRL	CONF_DSS_DATA2_RXACTIVE	CONF_DSS_DATA2_PUTYPESEL	CONF_DSS_DATA2_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_DSS_DATA2_MMODE			
R/W-0h				R/W-0h			

Table 7-99. CTRL_CONF_DSS_DATA2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_DATA2_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_DATA2_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_DATA2_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_DATA2_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_DATA2_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_DATA2_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_DATA2_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_DATA2_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_DATA2_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_DATA2_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-99. CTRL_CONF_DSS_DATA2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_DATA2_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_DATA2_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.89 CTRL_CONF_DSS_DATA3 Register (Offset = 8ACh) [reset = X]

CTRL_CONF_DSS_DATA3 is shown in Figure 7-93 and described in Table 7-100.

Return to Summary Table.

Figure 7-93. CTRL_CONF_DSS_DATA3 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_DSS_DATA3_WUEVT	CONF_DSS_DATA3_WUEN	CONF_DSS_DATA3_DSPULLTYPESELECT	CONF_DSS_DATA3_DSPULLUDEN	CONF_DSS_DATA3_DS0OUTVALUE	CONF_DSS_DATA3_DS0OUTEN	CONF_DSS_DATA3_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_DSS_DATA3_SLEWCTRL	CONF_DSS_DATA3_RXACTIVE	CONF_DSS_DATA3_PUTYPESEL	CONF_DSS_DATA3_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_DSS_DATA3_MMODE			
R/W-0h				R/W-0h			

Table 7-100. CTRL_CONF_DSS_DATA3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_DATA3_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_DATA3_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_DATA3_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_DATA3_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_DATA3_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_DATA3_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_DATA3_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_DATA3_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_DATA3_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_DATA3_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-100. CTRL_CONF_DSS_DATA3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_DATA3_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_DATA3_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.90 CTRL_CONF_DSS_DATA4 Register (Offset = 8B0h) [reset = X]

CTRL_CONF_DSS_DATA4 is shown in Figure 7-94 and described in Table 7-101.

Return to Summary Table.

Figure 7-94. CTRL_CONF_DSS_DATA4 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_DSS_DATA4_WUEVT	CONF_DSS_DATA4_WUEN	CONF_DSS_DATA4_DSPULLTYPESELECT	CONF_DSS_DATA4_DSPULLUDEN	CONF_DSS_DATA4_DS0OUTVALUE	CONF_DSS_DATA4_DS0OUTEN	CONF_DSS_DATA4_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_DSS_DATA4_SLEWCTRL	CONF_DSS_DATA4_RXACTIVE	CONF_DSS_DATA4_PUTYPESEL	CONF_DSS_DATA4_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_DSS_DATA4_MMODE			
R/W-0h				R/W-0h			

Table 7-101. CTRL_CONF_DSS_DATA4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_DATA4_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_DATA4_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_DATA4_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_DATA4_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_DATA4_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_DATA4_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_DATA4_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_DATA4_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_DATA4_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_DATA4_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-101. CTRL_CONF_DSS_DATA4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_DATA4_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_DATA4_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.91 CTRL_CONF_DSS_DATA5 Register (Offset = 8B4h) [reset = X]

CTRL_CONF_DSS_DATA5 is shown in Figure 7-95 and described in Table 7-102.

Return to Summary Table.

Figure 7-95. CTRL_CONF_DSS_DATA5 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_DSS_DATA5_WUEVT	CONF_DSS_DATA5_WUEN	CONF_DSS_DATA5_DSPULLTYPESELECT	CONF_DSS_DATA5_DSPULLUDEN	CONF_DSS_DATA5_DS0OUTVALUE	CONF_DSS_DATA5_DS0OUTEN	CONF_DSS_DATA5_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_DSS_DATA5_SLEWCTRL	CONF_DSS_DATA5_RXACTIVE	CONF_DSS_DATA5_PUTYPESEL	CONF_DSS_DATA5_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_DSS_DATA5_MMODE			
R/W-0h				R/W-0h			

Table 7-102. CTRL_CONF_DSS_DATA5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_DATA5_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_DATA5_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_DATA5_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_DATA5_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_DATA5_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_DATA5_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_DATA5_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_DATA5_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_DATA5_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_DATA5_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-102. CTRL_CONF_DSS_DATA5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_DATA5_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_DATA5_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.92 CTRL_CONF_DSS_DATA6 Register (Offset = 8B8h) [reset = X]

CTRL_CONF_DSS_DATA6 is shown in Figure 7-96 and described in Table 7-103.

Return to Summary Table.

Figure 7-96. CTRL_CONF_DSS_DATA6 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>RESERVED</td><td>CONF_DSS_DATA6_WUEVT</td><td>CONF_DSS_DATA6_WUEN</td><td>CONF_DSS_DATA6_DSPULLTYPESELECT</td><td>CONF_DSS_DATA6_DSPULLUDEN</td><td>CONF_DSS_DATA6_DS0OUTVALUE</td><td>CONF_DSS_DATA6_DS0OUTEN</td><td>CONF_DSS_DATA6_DS0EN</td></tr>
<tr><td>R/W-0h</td><td>R-X</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-1h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td>CONF_DSS_DATA6_SLEWCTRL</td><td>CONF_DSS_DATA6_RXACTIVE</td><td>CONF_DSS_DATA6_PUTYPESEL</td><td>CONF_DSS_DATA6_PUDEN</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R/W-0h</td><td>R/W-1h</td><td>R/W-0h</td><td>R/W-1h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">CONF_DSS_DATA6_MMODE</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 7-103. CTRL_CONF_DSS_DATA6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_DATA6_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_DATA6_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_DATA6_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_DATA6_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_DATA6_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_DATA6_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_DATA6_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_DATA6_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_DATA6_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_DATA6_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-103. CTRL_CONF_DSS_DATA6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_DATA6_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_DATA6_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.93 CTRL_CONF_DSS_DATA7 Register (Offset = 8BCh) [reset = X]

CTRL_CONF_DSS_DATA7 is shown in Figure 7-97 and described in Table 7-104.

Return to Summary Table.

Figure 7-97. CTRL_CONF_DSS_DATA7 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_DSS_DATA7_WUEVT	CONF_DSS_DATA7_WUEN	CONF_DSS_DATA7_DSPULLTYPESELECT	CONF_DSS_DATA7_DSPULLUDEN	CONF_DSS_DATA7_DS0OUTVALUE	CONF_DSS_DATA7_DS0OUTEN	CONF_DSS_DATA7_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_DSS_DATA7_SLEWCTRL	CONF_DSS_DATA7_RXACTIVE	CONF_DSS_DATA7_PUTYPESEL	CONF_DSS_DATA7_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_DSS_DATA7_MMODE			
R/W-0h				R/W-0h			

Table 7-104. CTRL_CONF_DSS_DATA7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_DATA7_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_DATA7_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_DATA7_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_DATA7_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_DATA7_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_DATA7_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_DATA7_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_DATA7_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_DATA7_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_DATA7_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-104. CTRL_CONF_DSS_DATA7 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_DATA7_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_DATA7_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.94 CTRL_CONF_DSS_DATA8 Register (Offset = 8C0h) [reset = X]

CTRL_CONF_DSS_DATA8 is shown in Figure 7-98 and described in Table 7-105.

Return to Summary Table.

Figure 7-98. CTRL_CONF_DSS_DATA8 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_DSS_DATA8_WUEVT	CONF_DSS_DATA8_WUEN	CONF_DSS_DATA8_DSPULLTYPESELECT	CONF_DSS_DATA8_DSPULLUDEN	CONF_DSS_DATA8_DS0OUTVALUE	CONF_DSS_DATA8_DS0OUTEN	CONF_DSS_DATA8_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_DSS_DATA8_SLEWCTRL	CONF_DSS_DATA8_RXACTIVE	CONF_DSS_DATA8_PUTYPESEL	CONF_DSS_DATA8_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_DSS_DATA8_MMODE			
R/W-0h				R/W-0h			

Table 7-105. CTRL_CONF_DSS_DATA8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_DATA8_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_DATA8_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_DATA8_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_DATA8_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_DATA8_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_DATA8_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_DATA8_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_DATA8_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_DATA8_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_DATA8_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-105. CTRL_CONF_DSS_DATA8 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_DATA8_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_DATA8_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.95 CTRL_CONF_DSS_DATA9 Register (Offset = 8C4h) [reset = X]

CTRL_CONF_DSS_DATA9 is shown in Figure 7-99 and described in Table 7-106.

Return to Summary Table.

Figure 7-99. CTRL_CONF_DSS_DATA9 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_DSS_DATA9_WUEVT	CONF_DSS_DATA9_WUEN	CONF_DSS_DATA9_DSPULLTYPESELECT	CONF_DSS_DATA9_DSPULLUDEN	CONF_DSS_DATA9_DS0OUTVALUE	CONF_DSS_DATA9_DS0OUTEN	CONF_DSS_DATA9_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_DSS_DATA9_SLEWCTRL	CONF_DSS_DATA9_RXACTIVE	CONF_DSS_DATA9_PUTYPESEL	CONF_DSS_DATA9_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_DSS_DATA9_MMODE			
R/W-0h				R/W-0h			

Table 7-106. CTRL_CONF_DSS_DATA9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_DATA9_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_DATA9_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_DATA9_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_DATA9_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_DATA9_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_DATA9_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_DATA9_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_DATA9_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_DATA9_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_DATA9_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-106. CTRL_CONF_DSS_DATA9 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_DATA9_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_DATA9_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.96 CTRL_CONF_DSS_DATA10 Register (Offset = 8C8h) [reset = X]

CTRL_CONF_DSS_DATA10 is shown in Figure 7-100 and described in Table 7-107.

Return to Summary Table.

Figure 7-100. CTRL_CONF_DSS_DATA10 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_DSS_DATA10_WUEVT	CONF_DSS_DATA10_WUEN	CONF_DSS_DATA10_DSPULLTYPESELECT	CONF_DSS_DATA10_DSPULLUDEN	CONF_DSS_DATA10_DS0OUTVALUE	CONF_DSS_DATA10_DS0OUTEN	CONF_DSS_DATA10_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_DSS_DATA10_SLEWCTRL	CONF_DSS_DATA10_RXACTIVE	CONF_DSS_DATA10_PUTYPESEL	CONF_DSS_DATA10_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_DSS_DATA10_MMODE			
R/W-0h				R/W-0h			

Table 7-107. CTRL_CONF_DSS_DATA10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_DATA10_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_DATA10_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_DATA10_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_DATA10_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_DATA10_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_DATA10_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_DATA10_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_DATA10_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_DATA10_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_DATA10_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-107. CTRL_CONF_DSS_DATA10 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_DATA10_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_DATA10_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.97 CTRL_CONF_DSS_DATA11 Register (Offset = 8CCh) [reset = X]

CTRL_CONF_DSS_DATA11 is shown in Figure 7-101 and described in Table 7-108.

Return to Summary Table.

Figure 7-101. CTRL_CONF_DSS_DATA11 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_DSS_DATA11_WUEVT	CONF_DSS_DATA11_WUEN	CONF_DSS_DATA11_DSPULLTYPESELECT	CONF_DSS_DATA11_DSPULLUDEN	CONF_DSS_DATA11_DS0OUTVALUE	CONF_DSS_DATA11_DS0OUTEN	CONF_DSS_DATA11_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_DSS_DATA11_SLEWCTRL	CONF_DSS_DATA11_RXACTIVE	CONF_DSS_DATA11_PUTYPESEL	CONF_DSS_DATA11_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_DSS_DATA11_MMODE			
R/W-0h				R/W-0h			

Table 7-108. CTRL_CONF_DSS_DATA11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_DATA11_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_DATA11_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_DATA11_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_DATA11_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_DATA11_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_DATA11_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_DATA11_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_DATA11_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_DATA11_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_DATA11_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-108. CTRL_CONF_DSS_DATA11 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_DATA11_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_DATA11_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.98 CTRL_CONF_DSS_DATA12 Register (Offset = 8D0h) [reset = X]

CTRL_CONF_DSS_DATA12 is shown in Figure 7-102 and described in Table 7-109.

Return to Summary Table.

Figure 7-102. CTRL_CONF_DSS_DATA12 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_DSS_DATA12_WUEVT	CONF_DSS_DATA12_WUEN	CONF_DSS_DATA12_DSPULLTYPESELECT	CONF_DSS_DATA12_DSPULLUDEN	CONF_DSS_DATA12_DS0OUTVALUE	CONF_DSS_DATA12_DS0OUTEN	CONF_DSS_DATA12_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_DSS_DATA12_SLEWCTRL	CONF_DSS_DATA12_RXACTIVE	CONF_DSS_DATA12_PUTYPESEL	CONF_DSS_DATA12_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_DSS_DATA12_MMODE			
R/W-0h				R/W-0h			

Table 7-109. CTRL_CONF_DSS_DATA12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_DATA12_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_DATA12_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_DATA12_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_DATA12_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_DATA12_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_DATA12_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_DATA12_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_DATA12_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_DATA12_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_DATA12_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-109. CTRL_CONF_DSS_DATA12 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_DATA12_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_DATA12_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.99 CTRL_CONF_DSS_DATA13 Register (Offset = 8D4h) [reset = X]

CTRL_CONF_DSS_DATA13 is shown in Figure 7-103 and described in Table 7-110.

Return to Summary Table.

Figure 7-103. CTRL_CONF_DSS_DATA13 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_DSS_DATA13_WUEVT	CONF_DSS_DATA13_WUEN	CONF_DSS_DATA13_DSPULLTYPESELECT	CONF_DSS_DATA13_DSPULLUDEN	CONF_DSS_DATA13_DS0OUTVALUE	CONF_DSS_DATA13_DS0OUTEN	CONF_DSS_DATA13_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_DSS_DATA13_SLEWCTRL	CONF_DSS_DATA13_RXACTIVE	CONF_DSS_DATA13_PUTYPESEL	CONF_DSS_DATA13_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_DSS_DATA13_MMODE			
R/W-0h				R/W-0h			

Table 7-110. CTRL_CONF_DSS_DATA13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_DATA13_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_DATA13_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_DATA13_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_DATA13_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_DATA13_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_DATA13_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_DATA13_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_DATA13_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_DATA13_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_DATA13_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-110. CTRL_CONF_DSS_DATA13 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_DATA13_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_DATA13_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.100 CTRL_CONF_DSS_DATA14 Register (Offset = 8D8h) [reset = X]

CTRL_CONF_DSS_DATA14 is shown in Figure 7-104 and described in Table 7-111.

Return to Summary Table.

Figure 7-104. CTRL_CONF_DSS_DATA14 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_DSS_DATA14_WUEVT	CONF_DSS_DATA14_WUEN	CONF_DSS_DATA14_DSPULLTYPESELECT	CONF_DSS_DATA14_DSPULLUDEN	CONF_DSS_DATA14_DS0OUTVALUE	CONF_DSS_DATA14_DS0OUTEN	CONF_DSS_DATA14_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_DSS_DATA14_SLEWCTRL	CONF_DSS_DATA14_RXACTIVE	CONF_DSS_DATA14_PUTYPESEL	CONF_DSS_DATA14_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_DSS_DATA14_MMODE			
R/W-0h				R/W-0h			

Table 7-111. CTRL_CONF_DSS_DATA14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_DATA14_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_DATA14_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_DATA14_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_DATA14_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_DATA14_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_DATA14_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_DATA14_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_DATA14_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_DATA14_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_DATA14_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-111. CTRL_CONF_DSS_DATA14 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_DATA14_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_DATA14_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.101 CTRL_CONF_DSS_DATA15 Register (Offset = 8DCh) [reset = X]

CTRL_CONF_DSS_DATA15 is shown in Figure 7-105 and described in Table 7-112.

Return to Summary Table.

Figure 7-105. CTRL_CONF_DSS_DATA15 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_DSS_DATA15_WUEVT	CONF_DSS_DATA15_WUEN	CONF_DSS_DATA15_DSPULLTYPESELECT	CONF_DSS_DATA15_DSPULLUDEN	CONF_DSS_DATA15_DS0OUTVALUE	CONF_DSS_DATA15_DS0OUTEN	CONF_DSS_DATA15_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_DSS_DATA15_SLEWCTRL	CONF_DSS_DATA15_RXACTIVE	CONF_DSS_DATA15_PUTYPESEL	CONF_DSS_DATA15_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_DSS_DATA15_MMODE			
R/W-0h				R/W-0h			

Table 7-112. CTRL_CONF_DSS_DATA15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_DATA15_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_DATA15_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_DATA15_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_DATA15_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_DATA15_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_DATA15_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_DATA15_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_DATA15_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_DATA15_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_DATA15_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-112. CTRL_CONF_DSS_DATA15 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_DATA15_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_DATA15_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.102 CTRL_CONF_DSS_VSYNC Register (Offset = 8E0h) [reset = X]

CTRL_CONF_DSS_VSYNC is shown in Figure 7-106 and described in Table 7-113.

Return to Summary Table.

Figure 7-106. CTRL_CONF_DSS_VSYNC Register

31	30	29	28	27	26	25	24
RESERVED	CONF_DSS_VSYNC_WUEVT	CONF_DSS_VSYNC_WUEN	CONF_DSS_VSYNC_DSPULLTYPESELECT	CONF_DSS_VSYNC_DSPULLUDEN	CONF_DSS_VSYNC_DS0OUTVALUE	CONF_DSS_VSYNC_DS0OUTEN	CONF_DSS_VSYNC_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_DSS_VSYNC_SLEWCTRL	CONF_DSS_VSYNC_RXACTIVE	CONF_DSS_VSYNC_PUTYPESEL	CONF_DSS_VSYNC_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_DSS_VSYNC_MMODE			
R/W-0h				R/W-0h			

Table 7-113. CTRL_CONF_DSS_VSYNC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_VSYNC_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_VSYNC_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_VSYNC_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_VSYNC_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_VSYNC_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_VSYNC_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_VSYNC_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_VSYNC_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_VSYNC_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_VSYNC_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-113. CTRL_CONF_DSS_VSYNC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_VSYNC_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_VSYNC_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.103 CTRL_CONF_DSS_HSYNC Register (Offset = 8E4h) [reset = X]

CTRL_CONF_DSS_HSYNC is shown in Figure 7-107 and described in Table 7-114.

Return to Summary Table.

Figure 7-107. CTRL_CONF_DSS_HSYNC Register

31	30	29	28	27	26	25	24
RESERVED	CONF_DSS_HSYNC_WUEVT	CONF_DSS_HSYNC_WUEN	CONF_DSS_HSYNC_DSPULLTYPESELECT	CONF_DSS_HSYNC_DSPULLUDEN	CONF_DSS_HSYNC_DS0OUTVALUE	CONF_DSS_HSYNC_DS0OUTEN	CONF_DSS_HSYNC_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_DSS_HSYNC_SLEWCTRL	CONF_DSS_HSYNC_RXACTIVE	CONF_DSS_HSYNC_PUTYPESEL	CONF_DSS_HSYNC_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_DSS_HSYNC_MMODE			
R/W-0h				R/W-0h			

Table 7-114. CTRL_CONF_DSS_HSYNC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_HSYNC_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_HSYNC_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_HSYNC_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_HSYNC_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_HSYNC_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_HSYNC_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_HSYNC_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_HSYNC_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_HSYNC_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_HSYNC_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-114. CTRL_CONF_DSS_HSYNC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_HSYNC_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_HSYNC_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.104 CTRL_CONF_DSS_PCLK Register (Offset = 8E8h) [reset = X]

CTRL_CONF_DSS_PCLK is shown in Figure 7-108 and described in Table 7-115.

Return to Summary Table.

Figure 7-108. CTRL_CONF_DSS_PCLK Register

31	30	29	28	27	26	25	24
RESERVED	CONF_DSS_PCLK_WUEVT	CONF_DSS_PCLK_WUEN	CONF_DSS_PCLK_DSPULLTYPESELECT	CONF_DSS_PCLK_DSPULLUDEN	CONF_DSS_PCLK_DS0OUTVALUE	CONF_DSS_PCLK_DS0OUTEN	CONF_DSS_PCLK_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_DSS_PCLK_SLEWCTRL	CONF_DSS_PCLK_RXACTIVE	CONF_DSS_PCLK_PUTYPESEL	CONF_DSS_PCLK_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_DSS_PCLK_MMODE			
R/W-0h				R/W-0h			

Table 7-115. CTRL_CONF_DSS_PCLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_PCLK_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_PCLK_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_PCLK_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_PCLK_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_PCLK_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_PCLK_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_PCLK_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_PCLK_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_PCLK_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_PCLK_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-115. CTRL_CONF_DSS_PCLK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_PCLK_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_PCLK_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.105 CTRL_CONF_DSS_AC_BIAS_EN Register (Offset = 8ECh) [reset = X]

CTRL_CONF_DSS_AC_BIAS_EN is shown in Figure 7-109 and described in Table 7-116.

Return to Summary Table.

Figure 7-109. CTRL_CONF_DSS_AC_BIAS_EN Register

31	30	29	28	27	26	25	24
RESERVED	CONF_DSS_AC_BIAS_EN_WUEVT	CONF_DSS_AC_BIAS_EN_WUEN	CONF_DSS_AC_BIAS_EN_DSPULLTYPESELECT	CONF_DSS_AC_BIAS_EN_DSPULLUDEN	CONF_DSS_AC_BIAS_EN_DS0OUTVALUE	CONF_DSS_AC_BIAS_EN_DS0OUTEN	CONF_DSS_AC_BIAS_EN_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_DSS_AC_BIAS_EN_SLEWCTRL	CONF_DSS_AC_BIAS_EN_RXACTIVE	CONF_DSS_AC_BIAS_EN_PUTYPESEL	CONF_DSS_AC_BIAS_EN_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_DSS_AC_BIAS_EN_MMODE			
R/W-0h				R/W-0h			

Table 7-116. CTRL_CONF_DSS_AC_BIAS_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_DSS_AC_BIAS_EN_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_DSS_AC_BIAS_EN_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_DSS_AC_BIAS_EN_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_DSS_AC_BIAS_EN_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_DSS_AC_BIAS_EN_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_DSS_AC_BIAS_EN_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_DSS_AC_BIAS_EN_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_DSS_AC_BIAS_EN_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_DSS_AC_BIAS_EN_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_DSS_AC_BIAS_EN_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-116. CTRL_CONF_DSS_AC_BIAS_EN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_DSS_AC_BIAS_EN_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_DSS_AC_BIAS_EN_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.106 CTRL_CONF_MMC0_DAT3 Register (Offset = 8F0h) [reset = X]

CTRL_CONF_MMC0_DAT3 is shown in Figure 7-110 and described in Table 7-117.

Return to Summary Table.

Figure 7-110. CTRL_CONF_MMC0_DAT3 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MMC0_DAT3_WUEVT	CONF_MMC0_DAT3_WUEN	CONF_MMC0_DAT3_DSPULLTYPESELECT	CONF_MMC0_DAT3_DSPULLUDEN	CONF_MMC0_DAT3_DS0OUTVALUE	CONF_MMC0_DAT3_DS0OUTEN	CONF_MMC0_DAT3_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_MMC0_DAT3_SLEWCTRL	CONF_MMC0_DAT3_RXACTIVE	CONF_MMC0_DAT3_PUTYPESEL	CONF_MMC0_DAT3_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_MMC0_DAT3_MMODE			
R/W-0h				R/W-7h			

Table 7-117. CTRL_CONF_MMC0_DAT3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MMC0_DAT3_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MMC0_DAT3_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MMC0_DAT3_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MMC0_DAT3_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MMC0_DAT3_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MMC0_DAT3_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MMC0_DAT3_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MMC0_DAT3_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MMC0_DAT3_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MMC0_DAT3_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-117. CTRL_CONF_MMC0_DAT3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MMC0_DAT3_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MMC0_DAT3_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.107 CTRL_CONF_MMC0_DAT2 Register (Offset = 8F4h) [reset = X]

CTRL_CONF_MMC0_DAT2 is shown in Figure 7-111 and described in Table 7-118.

Return to Summary Table.

Figure 7-111. CTRL_CONF_MMC0_DAT2 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MMC0_DAT2_WUEVT	CONF_MMC0_DAT2_WUEN	CONF_MMC0_DAT2_DSPULLTYPESELECT	CONF_MMC0_DAT2_DSPULLUDEN	CONF_MMC0_DAT2_DS0OUTVALUE	CONF_MMC0_DAT2_DS0OUTEN	CONF_MMC0_DAT2_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_MMC0_DAT2_SLEWCTRL	CONF_MMC0_DAT2_RXACTIVE	CONF_MMC0_DAT2_PUTYPESEL	CONF_MMC0_DAT2_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_MMC0_DAT2_MMODE			
R/W-0h				R/W-7h			

Table 7-118. CTRL_CONF_MMC0_DAT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MMC0_DAT2_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MMC0_DAT2_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MMC0_DAT2_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MMC0_DAT2_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MMC0_DAT2_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MMC0_DAT2_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MMC0_DAT2_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MMC0_DAT2_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MMC0_DAT2_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MMC0_DAT2_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-118. CTRL_CONF_MMC0_DAT2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MMC0_DAT2_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MMC0_DAT2_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.108 CTRL_CONF_MMC0_DAT1 Register (Offset = 8F8h) [reset = X]

CTRL_CONF_MMC0_DAT1 is shown in Figure 7-112 and described in Table 7-119.

Return to Summary Table.

Figure 7-112. CTRL_CONF_MMC0_DAT1 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MMC0_DAT1_WUEVT	CONF_MMC0_DAT1_WUEN	CONF_MMC0_DAT1_DSPULLTYPESELECT	CONF_MMC0_DAT1_DSPULLUDEN	CONF_MMC0_DAT1_DS0OUTVALUE	CONF_MMC0_DAT1_DS0OUTEN	CONF_MMC0_DAT1_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_MMC0_DAT1_SLEWCTRL	CONF_MMC0_DAT1_RXACTIVE	CONF_MMC0_DAT1_PUTYPESEL	CONF_MMC0_DAT1_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_MMC0_DAT1_MMODE			
R/W-0h				R/W-7h			

Table 7-119. CTRL_CONF_MMC0_DAT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MMC0_DAT1_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MMC0_DAT1_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MMC0_DAT1_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MMC0_DAT1_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MMC0_DAT1_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MMC0_DAT1_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MMC0_DAT1_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MMC0_DAT1_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MMC0_DAT1_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MMC0_DAT1_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-119. CTRL_CONF_MMC0_DAT1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MMC0_DAT1_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MMC0_DAT1_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.109 CTRL_CONF_MMC0_DAT0 Register (Offset = 8FCh) [reset = X]

CTRL_CONF_MMC0_DAT0 is shown in Figure 7-113 and described in Table 7-120.

Return to Summary Table.

Figure 7-113. CTRL_CONF_MMC0_DAT0 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MMC0_DAT0_WUEVT	CONF_MMC0_DAT0_WUEN	CONF_MMC0_DAT0_DSPULLTYPESELECT	CONF_MMC0_DAT0_DSPULLUDEN	CONF_MMC0_DAT0_DS0OUTVALUE	CONF_MMC0_DAT0_DS0OUTEN	CONF_MMC0_DAT0_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_MMC0_DAT0_SLEWCTRL	CONF_MMC0_DAT0_RXACTIVE	CONF_MMC0_DAT0_PUTYPESEL	CONF_MMC0_DAT0_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_MMC0_DAT0_MMODE			
R/W-0h				R/W-7h			

Table 7-120. CTRL_CONF_MMC0_DAT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MMC0_DAT0_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MMC0_DAT0_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MMC0_DAT0_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MMC0_DAT0_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MMC0_DAT0_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MMC0_DAT0_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MMC0_DAT0_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MMC0_DAT0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MMC0_DAT0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MMC0_DAT0_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-120. CTRL_CONF_MMC0_DAT0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MMC0_DAT0_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MMC0_DAT0_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.110 CTRL_CONF_MMC0_CLK Register (Offset = 900h) [reset = X]

CTRL_CONF_MMC0_CLK is shown in Figure 7-114 and described in Table 7-121.

Return to Summary Table.

Figure 7-114. CTRL_CONF_MMC0_CLK Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MMC0_CLK_WUEVT	CONF_MMC0_CLK_WUEN	CONF_MMC0_CLK_DSPULLTYPESELECT	CONF_MMC0_CLK_DSPULLUDEN	CONF_MMC0_CLK_DS0OUTVALUE	CONF_MMC0_CLK_DS0OUTEN	CONF_MMC0_CLK_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_MMC0_CLK_SLEWCTRL	CONF_MMC0_CLK_RXACTIVE	CONF_MMC0_CLK_PUTYPESEL	CONF_MMC0_CLK_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_MMC0_CLK_MMODE			
R/W-0h				R/W-7h			

Table 7-121. CTRL_CONF_MMC0_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MMC0_CLK_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MMC0_CLK_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MMC0_CLK_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MMC0_CLK_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MMC0_CLK_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MMC0_CLK_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MMC0_CLK_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MMC0_CLK_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MMC0_CLK_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MMC0_CLK_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-121. CTRL_CONF_MMC0_CLK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MMC0_CLK_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MMC0_CLK_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.111 CTRL_CONF_MMC0_CMD Register (Offset = 904h) [reset = X]

CTRL_CONF_MMC0_CMD is shown in Figure 7-115 and described in Table 7-122.

Return to Summary Table.

Figure 7-115. CTRL_CONF_MMC0_CMD Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MMC0_CMD_WUEVT	CONF_MMC0_CMD_WUEN	CONF_MMC0_CMD_DSPULLTYPESELECT	CONF_MMC0_CMD_DSPULLUDEN	CONF_MMC0_CMD_DS0OUTVALUE	CONF_MMC0_CMD_DS0OUTEN	CONF_MMC0_CMD_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_MMC0_CMD_SLEWCTRL	CONF_MMC0_CMD_RXACTIVE	CONF_MMC0_CMD_PUTYPESEL	CONF_MMC0_CMD_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_MMC0_CMD_MMODE			
R/W-0h				R/W-7h			

Table 7-122. CTRL_CONF_MMC0_CMD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MMC0_CMD_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MMC0_CMD_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MMC0_CMD_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MMC0_CMD_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MMC0_CMD_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MMC0_CMD_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MMC0_CMD_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MMC0_CMD_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MMC0_CMD_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MMC0_CMD_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-122. CTRL_CONF_MMC0_CMD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MMC0_CMD_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MMC0_CMD_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.112 CTRL_CONF_MII1_COL Register (Offset = 908h) [reset = X]

CTRL_CONF_MII1_COL is shown in Figure 7-116 and described in Table 7-123.

Return to Summary Table.

Figure 7-116. CTRL_CONF_MII1_COL Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MII1_COL_WUEVT	CONF_MII1_COL_WUEN	CONF_MII1_COL_DSPULLTYPESELECT	CONF_MII1_COL_DSPULLUDEN	CONF_MII1_COL_DS0OUTVALUE	CONF_MII1_COL_DS0OUTEN	CONF_MII1_COL_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_MII1_COL_SLEWCTRL	CONF_MII1_COL_RXACTIVE	CONF_MII1_COL_PUTYPESEL	CONF_MII1_COL_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_MII1_COL_MMODE			
R/W-0h				R/W-7h			

Table 7-123. CTRL_CONF_MII1_COL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MII1_COL_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MII1_COL_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MII1_COL_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MII1_COL_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MII1_COL_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MII1_COL_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MII1_COL_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MII1_COL_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MII1_COL_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MII1_COL_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-123. CTRL_CONF_MII1_COL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MII1_COL_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MII1_COL_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.113 CTRL_CONF_MII1_CRS Register (Offset = 90Ch) [reset = X]

CTRL_CONF_MII1_CRS is shown in Figure 7-117 and described in Table 7-124.

Return to Summary Table.

Figure 7-117. CTRL_CONF_MII1_CRS Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MII1_CRS_WUEVT	CONF_MII1_CRS_WUEN	CONF_MII1_CRS_DSPULLTYPESELECT	CONF_MII1_CRS_DSPULLUDEN	CONF_MII1_CRS_DS0OUTVALUE	CONF_MII1_CRS_DS0OUTEN	CONF_MII1_CRS_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_MII1_CRS_SLEWCTRL	CONF_MII1_CRS_RXACTIVE	CONF_MII1_CRS_PUTYPESEL	CONF_MII1_CRS_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_MII1_CRS_MMODE			
R/W-0h				R/W-7h			

Table 7-124. CTRL_CONF_MII1_CRS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MII1_CRS_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MII1_CRS_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MII1_CRS_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MII1_CRS_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MII1_CRS_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MII1_CRS_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MII1_CRS_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MII1_CRS_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MII1_CRS_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MII1_CRS_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-124. CTRL_CONF_MII1_CRS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MII1_CRS_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MII1_CRS_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.114 CTRL_CONF_MII1_RXERR Register (Offset = 910h) [reset = X]

CTRL_CONF_MII1_RXERR is shown in Figure 7-118 and described in Table 7-125.

Return to Summary Table.

Figure 7-118. CTRL_CONF_MII1_RXERR Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MII1_RXERR_WUEVT	CONF_MII1_RXERR_WUEN	CONF_MII1_RXERR_DSPULLTYPESELECT	CONF_MII1_RXERR_DSPULLUDEN	CONF_MII1_RXERR_DS0OUTVALUE	CONF_MII1_RXERR_DS0OUTEN	CONF_MII1_RXERR_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_MII1_RXERR_SLEWCTRL	CONF_MII1_RXERR_RXACTIVE	CONF_MII1_RXERR_PUTYPESEL	CONF_MII1_RXERR_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_MII1_RXERR_MMODE			
R/W-0h				R/W-7h			

Table 7-125. CTRL_CONF_MII1_RXERR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MII1_RXERR_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MII1_RXERR_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MII1_RXERR_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MII1_RXERR_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MII1_RXERR_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MII1_RXERR_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MII1_RXERR_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MII1_RXERR_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MII1_RXERR_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MII1_RXERR_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-125. CTRL_CONF_MII1_RXERR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MII1_RXERR_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MII1_RXERR_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.115 CTRL_CONF_MII1_TXEN Register (Offset = 914h) [reset = X]

CTRL_CONF_MII1_TXEN is shown in Figure 7-119 and described in Table 7-126.

Return to Summary Table.

Figure 7-119. CTRL_CONF_MII1_TXEN Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MII1_TXEN_WUEVT	CONF_MII1_TXEN_WUEN	CONF_MII1_TXEN_DSPULLTYPESELECT	CONF_MII1_TXEN_DSPULLUDEN	CONF_MII1_TXEN_DS0OUTVALUE	CONF_MII1_TXEN_DS0OUTEN	CONF_MII1_TXEN_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_MII1_TXEN_SLEWCTRL	CONF_MII1_TXEN_RXACTIVE	CONF_MII1_TXEN_PUTYPESEL	CONF_MII1_TXEN_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_MII1_TXEN_MMODE			
R/W-0h				R/W-7h			

Table 7-126. CTRL_CONF_MII1_TXEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MII1_TXEN_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MII1_TXEN_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MII1_TXEN_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MII1_TXEN_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MII1_TXEN_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MII1_TXEN_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MII1_TXEN_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MII1_TXEN_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MII1_TXEN_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MII1_TXEN_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-126. CTRL_CONF_MII1_TXEN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MII1_TXEN_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MII1_TXEN_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.116 CTRL_CONF_MII1_RXDV Register (Offset = 918h) [reset = X]

CTRL_CONF_MII1_RXDV is shown in Figure 7-120 and described in Table 7-127.

Return to Summary Table.

Figure 7-120. CTRL_CONF_MII1_RXDV Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MII1_RXDV_WUEVT	CONF_MII1_RXDV_WUEN	CONF_MII1_RXDV_DSPULLTYPESELECT	CONF_MII1_RXDV_DSPULLUDEN	CONF_MII1_RXDV_DS0OUTVALUE	CONF_MII1_RXDV_DS0OUTEN	CONF_MII1_RXDV_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_MII1_RXDV_SLEWCTRL	CONF_MII1_RXDV_RXACTIVE	CONF_MII1_RXDV_PUTYPESEL	CONF_MII1_RXDV_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_MII1_RXDV_MMODE			
R/W-0h				R/W-7h			

Table 7-127. CTRL_CONF_MII1_RXDV Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MII1_RXDV_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MII1_RXDV_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MII1_RXDV_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MII1_RXDV_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MII1_RXDV_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MII1_RXDV_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MII1_RXDV_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MII1_RXDV_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MII1_RXDV_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MII1_RXDV_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-127. CTRL_CONF_MII1_RXDV Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MII1_RXDV_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MII1_RXDV_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.117 CTRL_CONF_MII1_TXD3 Register (Offset = 91Ch) [reset = X]

CTRL_CONF_MII1_TXD3 is shown in Figure 7-121 and described in Table 7-128.

Return to Summary Table.

Figure 7-121. CTRL_CONF_MII1_TXD3 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MII1_TXD3_WUEVT	CONF_MII1_TXD3_WUEN	CONF_MII1_TXD3_DSPULLTYPESELECT	CONF_MII1_TXD3_DSPULLUDEN	CONF_MII1_TXD3_DS0OUTVALUE	CONF_MII1_TXD3_DS0OUTEN	CONF_MII1_TXD3_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_MII1_TXD3_SLEWCTRL	CONF_MII1_TXD3_RXACTIVE	CONF_MII1_TXD3_PUTYPESEL	CONF_MII1_TXD3_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_MII1_TXD3_MMODE			
R/W-0h				R/W-7h			

Table 7-128. CTRL_CONF_MII1_TXD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MII1_TXD3_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MII1_TXD3_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MII1_TXD3_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MII1_TXD3_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MII1_TXD3_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MII1_TXD3_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MII1_TXD3_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MII1_TXD3_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MII1_TXD3_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MII1_TXD3_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-128. CTRL_CONF_MII1_TXD3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MII1_TXD3_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MII1_TXD3_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.118 CTRL_CONF_MII1_TXD2 Register (Offset = 920h) [reset = X]

CTRL_CONF_MII1_TXD2 is shown in Figure 7-122 and described in Table 7-129.

Return to Summary Table.

Figure 7-122. CTRL_CONF_MII1_TXD2 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MII1_TXD2_WUEVT	CONF_MII1_TXD2_WUEN	CONF_MII1_TXD2_DSPULLTYPESELECT	CONF_MII1_TXD2_DSPULLUDEN	CONF_MII1_TXD2_DS0OUTVALUE	CONF_MII1_TXD2_DS0OUTEN	CONF_MII1_TXD2_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_MII1_TXD2_SLEWCTRL	CONF_MII1_TXD2_RXACTIVE	CONF_MII1_TXD2_PUTYPESEL	CONF_MII1_TXD2_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_MII1_TXD2_MMODE			
R/W-0h				R/W-7h			

Table 7-129. CTRL_CONF_MII1_TXD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MII1_TXD2_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MII1_TXD2_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MII1_TXD2_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MII1_TXD2_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MII1_TXD2_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MII1_TXD2_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MII1_TXD2_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MII1_TXD2_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MII1_TXD2_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MII1_TXD2_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-129. CTRL_CONF_MII1_TXD2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MII1_TXD2_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MII1_TXD2_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.119 CTRL_CONF_MII1_TXD1 Register (Offset = 924h) [reset = X]

CTRL_CONF_MII1_TXD1 is shown in Figure 7-123 and described in Table 7-130.

Return to Summary Table.

Figure 7-123. CTRL_CONF_MII1_TXD1 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MII1_TXD1_WUEVT	CONF_MII1_TXD1_WUEN	CONF_MII1_TXD1_DSPULLTYPESELECT	CONF_MII1_TXD1_DSPULLUDEN	CONF_MII1_TXD1_DS0OUTVALUE	CONF_MII1_TXD1_DS0OUTEN	CONF_MII1_TXD1_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_MII1_TXD1_SLEWCTRL	CONF_MII1_TXD1_RXACTIVE	CONF_MII1_TXD1_PUTYPESEL	CONF_MII1_TXD1_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_MII1_TXD1_MMODE			
R/W-0h				R/W-7h			

Table 7-130. CTRL_CONF_MII1_TXD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MII1_TXD1_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MII1_TXD1_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MII1_TXD1_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MII1_TXD1_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MII1_TXD1_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MII1_TXD1_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MII1_TXD1_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MII1_TXD1_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MII1_TXD1_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MII1_TXD1_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-130. CTRL_CONF_MII1_TXD1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MII1_TXD1_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MII1_TXD1_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.120 CTRL_CONF_MII1_TXD0 Register (Offset = 928h) [reset = X]

CTRL_CONF_MII1_TXD0 is shown in Figure 7-124 and described in Table 7-131.

Return to Summary Table.

Figure 7-124. CTRL_CONF_MII1_TXD0 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MII1_TXD0_WUEVT	CONF_MII1_TXD0_WUEN	CONF_MII1_TXD0_DSPULLTYPESELECT	CONF_MII1_TXD0_DSPULLUDEN	CONF_MII1_TXD0_DS0OUTVALUE	CONF_MII1_TXD0_DS0OUTEN	CONF_MII1_TXD0_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_MII1_TXD0_SLEWCTRL	CONF_MII1_TXD0_RXACTIVE	CONF_MII1_TXD0_PUTYPESEL	CONF_MII1_TXD0_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_MII1_TXD0_MMODE			
R/W-0h				R/W-7h			

Table 7-131. CTRL_CONF_MII1_TXD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MII1_TXD0_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MII1_TXD0_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MII1_TXD0_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MII1_TXD0_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MII1_TXD0_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MII1_TXD0_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MII1_TXD0_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MII1_TXD0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MII1_TXD0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MII1_TXD0_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-131. CTRL_CONF_MII1_TXD0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MII1_TXD0_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MII1_TXD0_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.121 CTRL_CONF_MII1_TXCLK Register (Offset = 92Ch) [reset = X]

CTRL_CONF_MII1_TXCLK is shown in Figure 7-125 and described in Table 7-132.

Return to Summary Table.

Figure 7-125. CTRL_CONF_MII1_TXCLK Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MII1_TXCLK_WUEVT	CONF_MII1_TXCLK_WUEN	CONF_MII1_TXCLK_DSPULLTYPESELECT	CONF_MII1_TXCLK_DSPULLUDEN	CONF_MII1_TXCLK_DS0OUTVALUE	CONF_MII1_TXCLK_DS0OUTEN	CONF_MII1_TXCLK_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_MII1_TXCLK_SLEWCTRL	CONF_MII1_TXCLK_RXACTIVE	CONF_MII1_TXCLK_PUTYPESEL	CONF_MII1_TXCLK_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_MII1_TXCLK_MMODE			
R/W-0h				R/W-7h			

Table 7-132. CTRL_CONF_MII1_TXCLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MII1_TXCLK_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MII1_TXCLK_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MII1_TXCLK_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MII1_TXCLK_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MII1_TXCLK_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MII1_TXCLK_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MII1_TXCLK_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MII1_TXCLK_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MII1_TXCLK_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MII1_TXCLK_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-132. CTRL_CONF_MII1_TXCLK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MII1_TXCLK_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MII1_TXCLK_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.122 CTRL_CONF_MII1_RXCLK Register (Offset = 930h) [reset = X]

CTRL_CONF_MII1_RXCLK is shown in Figure 7-126 and described in Table 7-133.

Return to Summary Table.

Figure 7-126. CTRL_CONF_MII1_RXCLK Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MII1_RXCLK_WUEVT	CONF_MII1_RXCLK_WUEN	CONF_MII1_RXCLK_DSPULLTYPESELECT	CONF_MII1_RXCLK_DSPULLUDEN	CONF_MII1_RXCLK_DS0OUTVALUE	CONF_MII1_RXCLK_DS0OUTEN	CONF_MII1_RXCLK_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_MII1_RXCLK_SLEWCTRL	CONF_MII1_RXCLK_RXACTIVE	CONF_MII1_RXCLK_PUTYPESEL	CONF_MII1_RXCLK_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_MII1_RXCLK_MMODE			
R/W-0h				R/W-7h			

Table 7-133. CTRL_CONF_MII1_RXCLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MII1_RXCLK_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MII1_RXCLK_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MII1_RXCLK_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MII1_RXCLK_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MII1_RXCLK_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MII1_RXCLK_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MII1_RXCLK_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MII1_RXCLK_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MII1_RXCLK_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MII1_RXCLK_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-133. CTRL_CONF_MII1_RXCLK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MII1_RXCLK_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MII1_RXCLK_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.123 CTRL_CONF_MII1_RXD3 Register (Offset = 934h) [reset = X]

CTRL_CONF_MII1_RXD3 is shown in Figure 7-127 and described in Table 7-134.

Return to Summary Table.

Figure 7-127. CTRL_CONF_MII1_RXD3 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MII1_RXD3_WUEVT	CONF_MII1_RXD3_WUEN	CONF_MII1_RXD3_DSPULLTYPESELECT	CONF_MII1_RXD3_DSPULLUDEN	CONF_MII1_RXD3_DS0OUTVALUE	CONF_MII1_RXD3_DS0OUTEN	CONF_MII1_RXD3_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_MII1_RXD3_SLEWCTRL	CONF_MII1_RXD3_RXACTIVE	CONF_MII1_RXD3_PUTYPESEL	CONF_MII1_RXD3_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_MII1_RXD3_MMODE			
R/W-0h				R/W-7h			

Table 7-134. CTRL_CONF_MII1_RXD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MII1_RXD3_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MII1_RXD3_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MII1_RXD3_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MII1_RXD3_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MII1_RXD3_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MII1_RXD3_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MII1_RXD3_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MII1_RXD3_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MII1_RXD3_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MII1_RXD3_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-134. CTRL_CONF_MII1_RXD3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MII1_RXD3_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MII1_RXD3_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.124 CTRL_CONF_MII1_RXD2 Register (Offset = 938h) [reset = X]

CTRL_CONF_MII1_RXD2 is shown in Figure 7-128 and described in Table 7-135.

Return to Summary Table.

Figure 7-128. CTRL_CONF_MII1_RXD2 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MII1_RXD2_WUEVT	CONF_MII1_RXD2_WUEN	CONF_MII1_RXD2_DSPULLTYPESELECT	CONF_MII1_RXD2_DSPULLUDEN	CONF_MII1_RXD2_DS0OUTVALUE	CONF_MII1_RXD2_DS0OUTEN	CONF_MII1_RXD2_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_MII1_RXD2_SLEWCTRL	CONF_MII1_RXD2_RXACTIVE	CONF_MII1_RXD2_PUTYPESEL	CONF_MII1_RXD2_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_MII1_RXD2_MMODE			
R/W-0h				R/W-7h			

Table 7-135. CTRL_CONF_MII1_RXD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MII1_RXD2_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MII1_RXD2_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MII1_RXD2_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MII1_RXD2_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MII1_RXD2_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MII1_RXD2_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MII1_RXD2_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MII1_RXD2_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MII1_RXD2_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MII1_RXD2_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-135. CTRL_CONF_MII1_RXD2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MII1_RXD2_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MII1_RXD2_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.125 CTRL_CONF_MII1_RXD1 Register (Offset = 93Ch) [reset = X]

CTRL_CONF_MII1_RXD1 is shown in Figure 7-129 and described in Table 7-136.

Return to Summary Table.

Figure 7-129. CTRL_CONF_MII1_RXD1 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MII1_RXD1_WUEVT	CONF_MII1_RXD1_WUEN	CONF_MII1_RXD1_DSPULLTYPESELECT	CONF_MII1_RXD1_DSPULLUDEN	CONF_MII1_RXD1_DS0OUTVALUE	CONF_MII1_RXD1_DS0OUTEN	CONF_MII1_RXD1_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_MII1_RXD1_SLEWCTRL	CONF_MII1_RXD1_RXACTIVE	CONF_MII1_RXD1_PUTYPESEL	CONF_MII1_RXD1_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_MII1_RXD1_MMODE			
R/W-0h				R/W-7h			

Table 7-136. CTRL_CONF_MII1_RXD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MII1_RXD1_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MII1_RXD1_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MII1_RXD1_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MII1_RXD1_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MII1_RXD1_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MII1_RXD1_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MII1_RXD1_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MII1_RXD1_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MII1_RXD1_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MII1_RXD1_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-136. CTRL_CONF_MII1_RXD1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MII1_RXD1_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MII1_RXD1_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.126 CTRL_CONF_MII1_RXD0 Register (Offset = 940h) [reset = X]

CTRL_CONF_MII1_RXD0 is shown in Figure 7-130 and described in Table 7-137.

Return to Summary Table.

Figure 7-130. CTRL_CONF_MII1_RXD0 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MII1_RXD0_WUEVT	CONF_MII1_RXD0_WUEN	CONF_MII1_RXD0_DSPULLTYPESELECT	CONF_MII1_RXD0_DSPULLUDEN	CONF_MII1_RXD0_DS0OUTVALUE	CONF_MII1_RXD0_DS0OUTEN	CONF_MII1_RXD0_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_MII1_RXD0_SLEWCTRL	CONF_MII1_RXD0_RXACTIVE	CONF_MII1_RXD0_PUTYPESEL	CONF_MII1_RXD0_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_MII1_RXD0_MMODE			
R/W-0h				R/W-7h			

Table 7-137. CTRL_CONF_MII1_RXD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MII1_RXD0_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MII1_RXD0_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MII1_RXD0_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MII1_RXD0_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MII1_RXD0_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MII1_RXD0_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MII1_RXD0_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MII1_RXD0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MII1_RXD0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MII1_RXD0_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-137. CTRL_CONF_MII1_RXD0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MII1_RXD0_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MII1_RXD0_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.127 CTRL_CONF_RMII1_REFCLK Register (Offset = 944h) [reset = X]

CTRL_CONF_RMII1_REFCLK is shown in Figure 7-131 and described in Table 7-138.

Return to Summary Table.

Figure 7-131. CTRL_CONF_RMII1_REFCLK Register

31	30	29	28	27	26	25	24
RESERVED	CONF_RMII1_REFCLK_WUEVT	CONF_RMII1_REFCLK_WUEN	CONF_RMII1_REFCLK_DSPULLTYPESELECT	CONF_RMII1_REFCLK_DSPULLUDEN	CONF_RMII1_REFCLK_DS0OUTVALUE	CONF_RMII1_REFCLK_DS0OUTEN	CONF_RMII1_REFCLK_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_RMII1_REFCLK_SLEWCTRL	CONF_RMII1_REFCLK_RXACTIVE	CONF_RMII1_REFCLK_PUTYPESEL	CONF_RMII1_REFCLK_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_RMII1_REFCLK_MMODE			
R/W-0h				R/W-7h			

Table 7-138. CTRL_CONF_RMII1_REFCLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_RMII1_REFCLK_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_RMII1_REFCLK_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_RMII1_REFCLK_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_RMII1_REFCLK_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_RMII1_REFCLK_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_RMII1_REFCLK_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_RMII1_REFCLK_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_RMII1_REFCLK_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_RMII1_REFCLK_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_RMII1_REFCLK_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-138. CTRL_CONF_RMII1_REFCLK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_RMII1_REFCLK_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_RMII1_REFCLK_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.128 CTRL_CONF_MDIO_DATA Register (Offset = 948h) [reset = X]

CTRL_CONF_MDIO_DATA is shown in Figure 7-132 and described in Table 7-139.

Return to Summary Table.

Figure 7-132. CTRL_CONF_MDIO_DATA Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MDIO_DATA_WUEVT	CONF_MDIO_DATA_WUEN	CONF_MDIO_DATA_DSPULLTYPESELECT	CONF_MDIO_DATA_DSPULLUDEN	CONF_MDIO_DATA_DS0OUTVALUE	CONF_MDIO_DATA_DS0OUTEN	CONF_MDIO_DATA_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_MDIO_DATA_SLEWCTRL	CONF_MDIO_DATA_RXACTIVE	CONF_MDIO_DATA_PUTYPESEL	CONF_MDIO_DATA_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_MDIO_DATA_MMODE			
R/W-0h				R/W-7h			

Table 7-139. CTRL_CONF_MDIO_DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MDIO_DATA_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MDIO_DATA_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MDIO_DATA_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MDIO_DATA_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MDIO_DATA_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MDIO_DATA_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MDIO_DATA_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MDIO_DATA_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MDIO_DATA_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MDIO_DATA_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-139. CTRL_CONF_MDIO_DATA Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MDIO_DATA_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MDIO_DATA_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.129 CTRL_CONF_MDIO_CLK Register (Offset = 94Ch) [reset = X]

CTRL_CONF_MDIO_CLK is shown in Figure 7-133 and described in Table 7-140.

Return to Summary Table.

Figure 7-133. CTRL_CONF_MDIO_CLK Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MDIO_CLK_WUEVT	CONF_MDIO_CLK_WUEN	CONF_MDIO_CLK_DSPULLTYPESELECT	CONF_MDIO_CLK_DSPULLUDEN	CONF_MDIO_CLK_DS0OUTVALUE	CONF_MDIO_CLK_DS0OUTEN	CONF_MDIO_CLK_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_MDIO_CLK_SLEWCTRL	CONF_MDIO_CLK_RXACTIVE	CONF_MDIO_CLK_PUTYPESEL	CONF_MDIO_CLK_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_MDIO_CLK_MMODE			
R/W-0h				R/W-7h			

Table 7-140. CTRL_CONF_MDIO_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MDIO_CLK_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MDIO_CLK_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MDIO_CLK_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MDIO_CLK_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MDIO_CLK_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MDIO_CLK_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MDIO_CLK_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MDIO_CLK_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MDIO_CLK_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MDIO_CLK_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-140. CTRL_CONF_MDIO_CLK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MDIO_CLK_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MDIO_CLK_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.130 CTRL_CONF_SPI0_SCLK Register (Offset = 950h) [reset = X]

CTRL_CONF_SPI0_SCLK is shown in Figure 7-134 and described in Table 7-141.

Return to Summary Table.

Figure 7-134. CTRL_CONF_SPI0_SCLK Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>RESERVED</td><td>CONF_SPI0_SCLK_WUEVT</td><td>CONF_SPI0_SCLK_WUEN</td><td>CONF_SPI0_SCLK_DSPULLTYPESELECT</td><td>CONF_SPI0_SCLK_DSPULLUDEN</td><td>CONF_SPI0_SCLK_DS0OUTVALUE</td><td>CONF_SPI0_SCLK_DS0OUTEN</td><td>CONF_SPI0_SCLK_DS0EN</td></tr>
<tr><td>R/W-0h</td><td>R-X</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-1h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td>CONF_SPI0_SCLK_SLEWCTRL</td><td>CONF_SPI0_SCLK_RXACTIVE</td><td>CONF_SPI0_SCLK_PUTYPESEL</td><td>CONF_SPI0_SCLK_PUDEN</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R/W-0h</td><td>R/W-1h</td><td>R/W-1h</td><td>R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">CONF_SPI0_SCLK_MMODE</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-7h</td></tr>
</table>

Table 7-141. CTRL_CONF_SPI0_SCLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_SPI0_SCLK_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_SPI0_SCLK_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_SPI0_SCLK_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_SPI0_SCLK_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_SPI0_SCLK_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_SPI0_SCLK_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_SPI0_SCLK_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_SPI0_SCLK_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_SPI0_SCLK_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_SPI0_SCLK_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-141. CTRL_CONF_SPI0_SCLK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_SPI0_SCLK_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_SPI0_SCLK_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.131 CTRL_CONF_SPI0_D0 Register (Offset = 954h) [reset = X]

CTRL_CONF_SPI0_D0 is shown in Figure 7-135 and described in Table 7-142.

Return to Summary Table.

Figure 7-135. CTRL_CONF_SPI0_D0 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_SPI0_D0_WUEVT	CONF_SPI0_D0_WUEN	CONF_SPI0_D0_DSPULLTYPESELECT	CONF_SPI0_D0_DSPULLUDEN	CONF_SPI0_D0_DS0OUTVALUE	CONF_SPI0_D0_DS0OUTEN	CONF_SPI0_D0_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_SPI0_D0_SLEWCTRL	CONF_SPI0_D0_RXACTIVE	CONF_SPI0_D0_PUTYPESEL	CONF_SPI0_D0_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_SPI0_D0_MMODE			
R/W-0h				R/W-7h			

Table 7-142. CTRL_CONF_SPI0_D0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_SPI0_D0_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_SPI0_D0_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_SPI0_D0_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_SPI0_D0_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_SPI0_D0_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_SPI0_D0_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_SPI0_D0_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_SPI0_D0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_SPI0_D0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_SPI0_D0_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-142. CTRL_CONF_SPI0_D0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_SPI0_D0_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_SPI0_D0_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.132 CTRL_CONF_SPI0_D1 Register (Offset = 958h) [reset = X]

CTRL_CONF_SPI0_D1 is shown in Figure 7-136 and described in Table 7-143.

Return to Summary Table.

Figure 7-136. CTRL_CONF_SPI0_D1 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_SPI0_D1_WUEVT	CONF_SPI0_D1_WUEN	CONF_SPI0_D1_DSPULLTYPESELECT	CONF_SPI0_D1_DSPULLUDEN	CONF_SPI0_D1_DS0OUTVALUE	CONF_SPI0_D1_DS0OUTEN	CONF_SPI0_D1_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_SPI0_D1_SLEWCTRL	CONF_SPI0_D1_RXACTIVE	CONF_SPI0_D1_PUTYPESEL	CONF_SPI0_D1_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_SPI0_D1_MMODE			
R/W-0h				R/W-7h			

Table 7-143. CTRL_CONF_SPI0_D1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_SPI0_D1_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_SPI0_D1_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_SPI0_D1_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_SPI0_D1_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_SPI0_D1_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_SPI0_D1_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_SPI0_D1_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_SPI0_D1_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_SPI0_D1_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_SPI0_D1_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-143. CTRL_CONF_SPI0_D1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_SPI0_D1_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_SPI0_D1_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.133 CTRL_CONF_SPI0_CS0 Register (Offset = 95Ch) [reset = X]

CTRL_CONF_SPI0_CS0 is shown in Figure 7-137 and described in Table 7-144.

Return to Summary Table.

Figure 7-137. CTRL_CONF_SPI0_CS0 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_SPI0_CS0_WUEVT	CONF_SPI0_CS0_WUEN	CONF_SPI0_CS0_DSPULLTYPESELECT	CONF_SPI0_CS0_DSPULLUDEN	CONF_SPI0_CS0_DS0OUTVALUE	CONF_SPI0_CS0_DS0OUTEN	CONF_SPI0_CS0_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_SPI0_CS0_SLEWCTRL	CONF_SPI0_CS0_RXACTIVE	CONF_SPI0_CS0_PUTYPESEL	CONF_SPI0_CS0_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_SPI0_CS0_MMODE			
R/W-0h				R/W-7h			

Table 7-144. CTRL_CONF_SPI0_CS0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_SPI0_CS0_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_SPI0_CS0_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_SPI0_CS0_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_SPI0_CS0_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_SPI0_CS0_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_SPI0_CS0_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_SPI0_CS0_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_SPI0_CS0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_SPI0_CS0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_SPI0_CS0_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-144. CTRL_CONF_SPI0_CS0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_SPI0_CS0_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_SPI0_CS0_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.134 CTRL_CONF_SPI0_CS1 Register (Offset = 960h) [reset = 00060007h]

Register mask: FFFFFFFFh

CTRL_CONF_SPI0_CS1 is shown in Figure 7-138 and described in Table 7-145.

Return to Summary Table.

Figure 7-138. CTRL_CONF_SPI0_CS1 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED				CONF_SPI0_CS1_SLEWCTRL	CONF_SPI0_CS1_RXACTIVE	CONF_SPI0_CS1_PUTYPESEL	CONF_SPI0_CS1_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_SPI0_CS1_MMODE			
R/W-0h				R/W-7h			

Table 7-145. CTRL_CONF_SPI0_CS1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_SPI0_CS1_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_SPI0_CS1_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_SPI0_CS1_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_SPI0_CS1_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_SPI0_CS1_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.135 CTRL_CONF_ECAP0_IN_PWM0_OUT Register (Offset = 964h) [reset = 00040007h]

Register mask: FFFFFFFFh

CTRL_CONF_ECAP0_IN_PWM0_OUT is shown in Figure 7-139 and described in Table 7-146.

Return to Summary Table.

Figure 7-139. CTRL_CONF_ECAP0_IN_PWM0_OUT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td>CONF_ECAP0_IN_PWM0_OUT_SLEWCTRL</td><td>CONF_ECAP0_IN_PWM0_OUT_RXACTIVE</td><td>CONF_ECAP0_IN_PWM0_OUT_PUTYPESEL</td><td>CONF_ECAP0_IN_PWM0_OUT_PUDEN</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R/W-0h</td><td>R/W-1h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">CONF_ECAP0_IN_PWM0_OUT_MMODE</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-7h</td></tr>
</table>

Table 7-146. CTRL_CONF_ECAP0_IN_PWM0_OUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_ECAP0_IN_PWM0_OUT_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_ECAP0_IN_PWM0_OUT_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_ECAP0_IN_PWM0_OUT_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_ECAP0_IN_PWM0_OUT_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_ECAP0_IN_PWM0_OUT_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.136 CTRL_CONF_UART0_CTSN Register (Offset = 968h) [reset = X]

CTRL_CONF_UART0_CTSN is shown in Figure 7-140 and described in Table 7-147.

Return to Summary Table.

Figure 7-140. CTRL_CONF_UART0_CTSN Register

31	30	29	28	27	26	25	24
RESERVED	CONF_UART0_CTSN_WUEVT	CONF_UART0_CTSN_WUEN	CONF_UART0_CTSN_DSPULLTYPESELECT	CONF_UART0_CTSN_DSPULLUDEN	CONF_UART0_CTSN_DS0OUTVALUE	CONF_UART0_CTSN_DS0OUTEN	CONF_UART0_CTSN_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_UART0_CTSN_SLEWCTRL	CONF_UART0_CTSN_RXACTIVE	CONF_UART0_CTSN_PUTYPESEL	CONF_UART0_CTSN_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_UART0_CTSN_MMODE			
R/W-0h				R/W-7h			

Table 7-147. CTRL_CONF_UART0_CTSN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_UART0_CTSN_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_UART0_CTSN_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_UART0_CTSN_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_UART0_CTSN_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_UART0_CTSN_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_UART0_CTSN_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_UART0_CTSN_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_UART0_CTSN_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_UART0_CTSN_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_UART0_CTSN_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-147. CTRL_CONF_UART0_CTSN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_UART0_CTSN_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_UART0_CTSN_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.137 CTRL_CONF_UART0_RTSN Register (Offset = 96Ch) [reset = X]

CTRL_CONF_UART0_RTSN is shown in Figure 7-141 and described in Table 7-148.

Return to Summary Table.

Figure 7-141. CTRL_CONF_UART0_RTSN Register

31	30	29	28	27	26	25	24
RESERVED	CONF_UART0_RTSN_WUEVT	CONF_UART0_RTSN_WUEN	CONF_UART0_RTSN_DSPULLTYPESELECT	CONF_UART0_RTSN_DSPULLUDEN	CONF_UART0_RTSN_DS0OUTVALUE	CONF_UART0_RTSN_DS0OUTEN	CONF_UART0_RTSN_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_UART0_RTSN_SLEWCTRL	CONF_UART0_RTSN_RXACTIVE	CONF_UART0_RTSN_PUTYPESEL	CONF_UART0_RTSN_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_UART0_RTSN_MMODE			
R/W-0h				R/W-7h			

Table 7-148. CTRL_CONF_UART0_RTSN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_UART0_RTSN_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_UART0_RTSN_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_UART0_RTSN_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_UART0_RTSN_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_UART0_RTSN_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_UART0_RTSN_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_UART0_RTSN_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_UART0_RTSN_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_UART0_RTSN_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_UART0_RTSN_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-148. CTRL_CONF_UART0_RTSN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_UART0_RTSN_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_UART0_RTSN_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.138 CTRL_CONF_UART0_RXD Register (Offset = 970h) [reset = X]

CTRL_CONF_UART0_RXD is shown in Figure 7-142 and described in Table 7-149.

Return to Summary Table.

Figure 7-142. CTRL_CONF_UART0_RXD Register

31	30	29	28	27	26	25	24
RESERVED	CONF_UART0_RXD_WUEVT	CONF_UART0_RXD_WUEN	CONF_UART0_RXD_DSPULLTYPESELECT	CONF_UART0_RXD_DSPULLUDEN	CONF_UART0_RXD_DS0OUTVALUE	CONF_UART0_RXD_DS0OUTEN	CONF_UART0_RXD_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_UART0_RXD_SLEWCTRL	CONF_UART0_RXD_RXACTIVE	CONF_UART0_RXD_PUTYPESEL	CONF_UART0_RXD_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_UART0_RXD_MMODE			
R/W-0h				R/W-7h			

Table 7-149. CTRL_CONF_UART0_RXD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_UART0_RXD_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_UART0_RXD_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_UART0_RXD_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_UART0_RXD_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_UART0_RXD_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_UART0_RXD_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_UART0_RXD_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_UART0_RXD_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_UART0_RXD_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_UART0_RXD_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-149. CTRL_CONF_UART0_RXD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_UART0_RXD_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_UART0_RXD_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.139 CTRL_CONF_UART0_TXD Register (Offset = 974h) [reset = X]

CTRL_CONF_UART0_TXD is shown in Figure 7-143 and described in Table 7-150.

Return to Summary Table.

Figure 7-143. CTRL_CONF_UART0_TXD Register

31	30	29	28	27	26	25	24
RESERVED	CONF_UART0_TXD_WUEVT	CONF_UART0_TXD_WUEN	CONF_UART0_TXD_DSPULLTYPESELECT	CONF_UART0_TXD_DSPULLUDEN	CONF_UART0_TXD_DS0OUTVALUE	CONF_UART0_TXD_DS0OUTEN	CONF_UART0_TXD_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_UART0_TXD_SLEWCTRL	CONF_UART0_TXD_RXACTIVE	CONF_UART0_TXD_PUTYPESEL	CONF_UART0_TXD_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_UART0_TXD_MMODE			
R/W-0h				R/W-7h			

Table 7-150. CTRL_CONF_UART0_TXD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_UART0_TXD_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_UART0_TXD_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_UART0_TXD_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_UART0_TXD_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_UART0_TXD_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_UART0_TXD_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_UART0_TXD_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_UART0_TXD_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_UART0_TXD_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_UART0_TXD_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-150. CTRL_CONF_UART0_TXD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_UART0_TXD_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_UART0_TXD_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.140 CTRL_CONF_UART1_CTSN Register (Offset = 978h) [reset = X]

CTRL_CONF_UART1_CTSN is shown in Figure 7-144 and described in Table 7-151.

Return to Summary Table.

Figure 7-144. CTRL_CONF_UART1_CTSN Register

31	30	29	28	27	26	25	24
RESERVED	CONF_UART1_CTSN_WUEVT	CONF_UART1_CTSN_WUEN	CONF_UART1_CTSN_DSPULLTYPESELECT	CONF_UART1_CTSN_DSPULLUDEN	CONF_UART1_CTSN_DS0OUTVALUE	CONF_UART1_CTSN_DS0OUTEN	CONF_UART1_CTSN_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_UART1_CTSN_SLEWCTRL	CONF_UART1_CTSN_RXACTIVE	CONF_UART1_CTSN_PUTYPESEL	CONF_UART1_CTSN_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_UART1_CTSN_MMODE			
R/W-0h				R/W-7h			

Table 7-151. CTRL_CONF_UART1_CTSN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_UART1_CTSN_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_UART1_CTSN_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_UART1_CTSN_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_UART1_CTSN_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_UART1_CTSN_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_UART1_CTSN_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_UART1_CTSN_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_UART1_CTSN_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_UART1_CTSN_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_UART1_CTSN_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-151. CTRL_CONF_UART1_CTSN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_UART1_CTSN_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_UART1_CTSN_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.141 CTRL_CONF_UART1_RTSN Register (Offset = 97Ch) [reset = X]

CTRL_CONF_UART1_RTSN is shown in Figure 7-145 and described in Table 7-152.

Return to Summary Table.

Figure 7-145. CTRL_CONF_UART1_RTSN Register

31	30	29	28	27	26	25	24
RESERVED	CONF_UART1_RTSN_WUEVT	CONF_UART1_RTSN_WUEN	CONF_UART1_RTSN_DSPULLTYPESELECT	CONF_UART1_RTSN_DSPULLUDEN	CONF_UART1_RTSN_DS0OUTVALUE	CONF_UART1_RTSN_DS0OUTEN	CONF_UART1_RTSN_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_UART1_RTSN_SLEWCTRL	CONF_UART1_RTSN_RXACTIVE	CONF_UART1_RTSN_PUTYPESEL	CONF_UART1_RTSN_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_UART1_RTSN_MMODE			
R/W-0h				R/W-7h			

Table 7-152. CTRL_CONF_UART1_RTSN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_UART1_RTSN_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_UART1_RTSN_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_UART1_RTSN_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_UART1_RTSN_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_UART1_RTSN_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_UART1_RTSN_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_UART1_RTSN_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_UART1_RTSN_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_UART1_RTSN_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_UART1_RTSN_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-152. CTRL_CONF_UART1_RTSN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_UART1_RTSN_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_UART1_RTSN_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.142 CTRL_CONF_UART1_RXD Register (Offset = 980h) [reset = X]

CTRL_CONF_UART1_RXD is shown in Figure 7-146 and described in Table 7-153.

Return to Summary Table.

Figure 7-146. CTRL_CONF_UART1_RXD Register

31	30	29	28	27	26	25	24
RESERVED	CONF_UART1_RXD_WUEVT	CONF_UART1_RXD_WUEN	CONF_UART1_RXD_DSPULLTYPESELECT	CONF_UART1_RXD_DSPULLUDEN	CONF_UART1_RXD_DS0OUTVALUE	CONF_UART1_RXD_DS0OUTEN	CONF_UART1_RXD_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_UART1_RXD_SLEWCTRL	CONF_UART1_RXD_RXACTIVE	CONF_UART1_RXD_PUTYPESEL	CONF_UART1_RXD_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_UART1_RXD_MMODE			
R/W-0h				R/W-7h			

Table 7-153. CTRL_CONF_UART1_RXD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_UART1_RXD_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_UART1_RXD_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_UART1_RXD_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_UART1_RXD_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_UART1_RXD_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_UART1_RXD_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_UART1_RXD_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_UART1_RXD_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_UART1_RXD_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_UART1_RXD_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-153. CTRL_CONF_UART1_RXD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_UART1_RXD_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_UART1_RXD_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.143 CTRL_CONF_UART1_TXD Register (Offset = 984h) [reset = X]

CTRL_CONF_UART1_TXD is shown in Figure 7-147 and described in Table 7-154.

Return to Summary Table.

Figure 7-147. CTRL_CONF_UART1_TXD Register

31	30	29	28	27	26	25	24
RESERVED	CONF_UART1_TXD_WUEVT	CONF_UART1_TXD_WUEN	CONF_UART1_TXD_DSPULLTYPESELECT	CONF_UART1_TXD_DSPULLUDEN	CONF_UART1_TXD_DS0OUTVALUE	CONF_UART1_TXD_DS0OUTEN	CONF_UART1_TXD_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_UART1_TXD_SLEWCTRL	CONF_UART1_TXD_RXACTIVE	CONF_UART1_TXD_PUTYPESEL	CONF_UART1_TXD_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_UART1_TXD_MMODE			
R/W-0h				R/W-7h			

Table 7-154. CTRL_CONF_UART1_TXD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_UART1_TXD_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_UART1_TXD_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_UART1_TXD_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_UART1_TXD_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_UART1_TXD_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_UART1_TXD_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_UART1_TXD_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_UART1_TXD_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_UART1_TXD_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_UART1_TXD_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-154. CTRL_CONF_UART1_TXD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_UART1_TXD_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_UART1_TXD_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.144 CTRL_CONF_I2C0_SDA Register (Offset = 988h) [reset = 00060007h]

Register mask: FFFFFFFFh

CTRL_CONF_I2C0_SDA is shown in Figure 7-148 and described in Table 7-155.

Return to Summary Table.

Figure 7-148. CTRL_CONF_I2C0_SDA Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_I2C0_SDA_SLEWCTRL	CONF_I2C0_SDA_RXACTIVE	CONF_I2C0_SDA_PUTYPESEL	CONF_I2C0_SDA_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_I2C0_SDA_MMODE			
R/W-0h				R/W-7h			

Table 7-155. CTRL_CONF_I2C0_SDA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_I2C0_SDA_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_I2C0_SDA_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_I2C0_SDA_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_I2C0_SDA_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_I2C0_SDA_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.145 CTRL_CONF_I2C0_SCL Register (Offset = 98Ch) [reset = 00060007h]

Register mask: FFFFFFFFh

CTRL_CONF_I2C0_SCL is shown in Figure 7-149 and described in Table 7-156.

Return to Summary Table.

Figure 7-149. CTRL_CONF_I2C0_SCL Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							

23	22	21	20	19	18	17	16
RESERVED				CONF_I2C0_SCL_SLEWCTRL	CONF_I2C0_SCL_RXACTIVE	CONF_I2C0_SCL_PUTYPESEL	CONF_I2C0_SCL_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_I2C0_SCL_MMODE			
R/W-0h				R/W-7h			

Table 7-156. CTRL_CONF_I2C0_SCL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_I2C0_SCL_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_I2C0_SCL_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_I2C0_SCL_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_I2C0_SCL_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_I2C0_SCL_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.146 CTRL_CONF_MCASP0_ACLKX Register (Offset = 990h) [reset = X]

CTRL_CONF_MCASP0_ACLKX is shown in Figure 7-150 and described in Table 7-157.

Return to Summary Table.

Figure 7-150. CTRL_CONF_MCASP0_ACLKX Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MCASP0_ACLKX_WUEVT	CONF_MCASP0_ACLKX_WUEN	CONF_MCASP0_ACLKX_DSPULLTYPESELECT	CONF_MCASP0_ACLKX_DSPULLUDEN	CONF_MCASP0_ACLKX_DS0OUTVALUE	CONF_MCASP0_ACLKX_DS0OUTEN	CONF_MCASP0_ACLKX_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_MCASP0_ACLKX_SLEWCTRL	CONF_MCASP0_ACLKX_RXACTIVE	CONF_MCASP0_ACLKX_PUTYPESEL	CONF_MCASP0_ACLKX_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_MCASP0_ACLKX_MMODE			
R/W-0h				R/W-7h			

Table 7-157. CTRL_CONF_MCASP0_ACLKX Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MCASP0_ACLKX_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MCASP0_ACLKX_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MCASP0_ACLKX_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MCASP0_ACLKX_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MCASP0_ACLKX_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MCASP0_ACLKX_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MCASP0_ACLKX_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MCASP0_ACLKX_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MCASP0_ACLKX_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MCASP0_ACLKX_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-157. CTRL_CONF_MCASP0_ACLKX Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MCASP0_ACLKX_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MCASP0_ACLKX_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.147 CTRL_CONF_MCASP0_FSX Register (Offset = 994h) [reset = X]

CTRL_CONF_MCASP0_FSX is shown in Figure 7-151 and described in Table 7-158.

Return to Summary Table.

Figure 7-151. CTRL_CONF_MCASP0_FSX Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MCASP0_FSX_WUEVT	CONF_MCASP0_FSX_WUEN	CONF_MCASP0_FSX_DSPULLTYPESELECT	CONF_MCASP0_FSX_DSPULLUDEN	CONF_MCASP0_FSX_DS0OUTVALUE	CONF_MCASP0_FSX_DS0OUTEN	CONF_MCASP0_FSX_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_MCASP0_FSX_SLEWCTRL	CONF_MCASP0_FSX_RXACTIVE	CONF_MCASP0_FSX_PUTYPESEL	CONF_MCASP0_FSX_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_MCASP0_FSX_MMODE			
R/W-0h				R/W-7h			

Table 7-158. CTRL_CONF_MCASP0_FSX Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MCASP0_FSX_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MCASP0_FSX_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MCASP0_FSX_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MCASP0_FSX_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MCASP0_FSX_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MCASP0_FSX_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MCASP0_FSX_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MCASP0_FSX_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MCASP0_FSX_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MCASP0_FSX_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-158. CTRL_CONF_MCASP0_FSX Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MCASP0_FSX_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MCASP0_FSX_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.148 CTRL_CONF_MCASP0_AXR0 Register (Offset = 998h) [reset = X]

CTRL_CONF_MCASP0_AXR0 is shown in Figure 7-152 and described in Table 7-159.

Return to Summary Table.

Figure 7-152. CTRL_CONF_MCASP0_AXR0 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MCASP0_AXR0_WUEVT	CONF_MCASP0_AXR0_WUEN	CONF_MCASP0_AXR0_DSPULLTYPESELECT	CONF_MCASP0_AXR0_DSPULLUDEN	CONF_MCASP0_AXR0_DS0OUTVALUE	CONF_MCASP0_AXR0_DS0OUTEN	CONF_MCASP0_AXR0_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_MCASP0_AXR0_SLEWCTRL	CONF_MCASP0_AXR0_RXACTIVE	CONF_MCASP0_AXR0_PUTYPESEL	CONF_MCASP0_AXR0_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_MCASP0_AXR0_MMODE			
R/W-0h				R/W-7h			

Table 7-159. CTRL_CONF_MCASP0_AXR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MCASP0_AXR0_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MCASP0_AXR0_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MCASP0_AXR0_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MCASP0_AXR0_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MCASP0_AXR0_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MCASP0_AXR0_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MCASP0_AXR0_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MCASP0_AXR0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MCASP0_AXR0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MCASP0_AXR0_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-159. CTRL_CONF_MCASP0_AXR0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MCASP0_AXR0_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MCASP0_AXR0_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.149 CTRL_CONF_MCASP0_AHCLKR Register (Offset = 99Ch) [reset = X]

CTRL_CONF_MCASP0_AHCLKR is shown in Figure 7-153 and described in Table 7-160.

Return to Summary Table.

Figure 7-153. CTRL_CONF_MCASP0_AHCLKR Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MCASP0_AHCLKR_WUEVT	CONF_MCASP0_AHCLKR_WUEN	CONF_MCASP0_AHCLKR_DSPULLTYPESELECT	CONF_MCASP0_AHCLKR_DSPULLUDEN	CONF_MCASP0_AHCLKR_DS0OUTVALUE	CONF_MCASP0_AHCLKR_DS0OUTEN	CONF_MCASP0_AHCLKR_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_MCASP0_AHCLKR_SLEWCTRL	CONF_MCASP0_AHCLKR_RXACTIVE	CONF_MCASP0_AHCLKR_PUTYPESEL	CONF_MCASP0_AHCLKR_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_MCASP0_AHCLKR_MMODE			
R/W-0h				R/W-7h			

Table 7-160. CTRL_CONF_MCASP0_AHCLKR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MCASP0_AHCLKR_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MCASP0_AHCLKR_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MCASP0_AHCLKR_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MCASP0_AHCLKR_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MCASP0_AHCLKR_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MCASP0_AHCLKR_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MCASP0_AHCLKR_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MCASP0_AHCLKR_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MCASP0_AHCLKR_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MCASP0_AHCLKR_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-160. CTRL_CONF_MCASP0_AHCLKR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MCASP0_AHCLKR_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MCASP0_AHCLKR_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.150 CTRL_CONF_MCASP0_ACLKR Register (Offset = 9A0h) [reset = X]

CTRL_CONF_MCASP0_ACLKR is shown in Figure 7-154 and described in Table 7-161.

Return to Summary Table.

Figure 7-154. CTRL_CONF_MCASP0_ACLKR Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MCASP0_ACLKR_WUEVT	CONF_MCASP0_ACLKR_WUEN	CONF_MCASP0_ACLKR_DSPULLTYPESELECT	CONF_MCASP0_ACLKR_DSPULLUDEN	CONF_MCASP0_ACLKR_DS0OUTVALUE	CONF_MCASP0_ACLKR_DS0OUTEN	CONF_MCASP0_ACLKR_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_MCASP0_ACLKR_SLEWCTRL	CONF_MCASP0_ACLKR_RXACTIVE	CONF_MCASP0_ACLKR_PUTYPESEL	CONF_MCASP0_ACLKR_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_MCASP0_ACLKR_MMODE			
R/W-0h				R/W-7h			

Table 7-161. CTRL_CONF_MCASP0_ACLKR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MCASP0_ACLKR_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MCASP0_ACLKR_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MCASP0_ACLKR_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MCASP0_ACLKR_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MCASP0_ACLKR_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MCASP0_ACLKR_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MCASP0_ACLKR_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MCASP0_ACLKR_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MCASP0_ACLKR_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MCASP0_ACLKR_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-161. CTRL_CONF_MCASP0_ACLKR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MCASP0_ACLKR_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MCASP0_ACLKR_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.151 CTRL_CONF_MCASP0_FSR Register (Offset = 9A4h) [reset = X]

CTRL_CONF_MCASP0_FSR is shown in Figure 7-155 and described in Table 7-162.

Return to Summary Table.

Figure 7-155. CTRL_CONF_MCASP0_FSR Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MCASP0_FSR_WUEVT	CONF_MCASP0_FSR_WUEN	CONF_MCASP0_FSR_DSPULLTYPESELECT	CONF_MCASP0_FSR_DSPULLUDEN	CONF_MCASP0_FSR_DS0OUTVALUE	CONF_MCASP0_FSR_DS0OUTEN	CONF_MCASP0_FSR_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_MCASP0_FSR_SLEWCTRL	CONF_MCASP0_FSR_RXACTIVE	CONF_MCASP0_FSR_PUTYPESEL	CONF_MCASP0_FSR_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_MCASP0_FSR_MMODE			
R/W-0h				R/W-7h			

Table 7-162. CTRL_CONF_MCASP0_FSR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MCASP0_FSR_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MCASP0_FSR_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MCASP0_FSR_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MCASP0_FSR_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MCASP0_FSR_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MCASP0_FSR_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MCASP0_FSR_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MCASP0_FSR_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MCASP0_FSR_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MCASP0_FSR_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-162. CTRL_CONF_MCASP0_FSR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MCASP0_FSR_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MCASP0_FSR_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.152 CTRL_CONF_MCASP0_AXR1 Register (Offset = 9A8h) [reset = X]

CTRL_CONF_MCASP0_AXR1 is shown in Figure 7-156 and described in Table 7-163.

Return to Summary Table.

Figure 7-156. CTRL_CONF_MCASP0_AXR1 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MCASP0_AXR1_WUEVT	CONF_MCASP0_AXR1_WUEN	CONF_MCASP0_AXR1_DSPULLTYPESELECT	CONF_MCASP0_AXR1_DSPULLUDEN	CONF_MCASP0_AXR1_DS0OUTVALUE	CONF_MCASP0_AXR1_DS0OUTEN	CONF_MCASP0_AXR1_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_MCASP0_AXR1_SLEWCTRL	CONF_MCASP0_AXR1_RXACTIVE	CONF_MCASP0_AXR1_PUTYPESEL	CONF_MCASP0_AXR1_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_MCASP0_AXR1_MMODE			
R/W-0h				R/W-7h			

Table 7-163. CTRL_CONF_MCASP0_AXR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MCASP0_AXR1_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MCASP0_AXR1_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MCASP0_AXR1_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MCASP0_AXR1_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MCASP0_AXR1_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MCASP0_AXR1_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MCASP0_AXR1_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MCASP0_AXR1_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MCASP0_AXR1_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MCASP0_AXR1_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-163. CTRL_CONF_MCASP0_AXR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MCASP0_AXR1_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MCASP0_AXR1_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.153 CTRL_CONF_MCASP0_AHCLKX Register (Offset = 9ACh) [reset = X]

CTRL_CONF_MCASP0_AHCLKX is shown in Figure 7-157 and described in Table 7-164.

Return to Summary Table.

Figure 7-157. CTRL_CONF_MCASP0_AHCLKX Register

31	30	29	28	27	26	25	24
RESERVED	CONF_MCASP0_AHCLKX_WUEVT	CONF_MCASP0_AHCLKX_WUEN	CONF_MCASP0_AHCLKX_DSPULLTYPESELECT	CONF_MCASP0_AHCLKX_DSPULLUDEN	CONF_MCASP0_AHCLKX_DS0OUTVALUE	CONF_MCASP0_AHCLKX_DS0OUTEN	CONF_MCASP0_AHCLKX_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_MCASP0_AHCLKX_SLEWCTRL	CONF_MCASP0_AHCLKX_RXACTIVE	CONF_MCASP0_AHCLKX_PUTYPESEL	CONF_MCASP0_AHCLKX_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_MCASP0_AHCLKX_MMODE			
R/W-0h				R/W-7h			

Table 7-164. CTRL_CONF_MCASP0_AHCLKX Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_MCASP0_AHCLKX_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_MCASP0_AHCLKX_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_MCASP0_AHCLKX_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_MCASP0_AHCLKX_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_MCASP0_AHCLKX_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_MCASP0_AHCLKX_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_MCASP0_AHCLKX_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_MCASP0_AHCLKX_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_MCASP0_AHCLKX_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_MCASP0_AHCLKX_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-164. CTRL_CONF_MCASP0_AHCLKX Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_MCASP0_AHCLKX_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_MCASP0_AHCLKX_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.154 CTRL_CONF_CAM0_HD Register (Offset = 9B0h) [reset = X]

CTRL_CONF_CAM0_HD is shown in Figure 7-158 and described in Table 7-165.

Return to Summary Table.

Figure 7-158. CTRL_CONF_CAM0_HD Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM0_HD_WUEVT	CONF_CAM0_HD_WUEN	CONF_CAM0_HD_DSPULLTYPESELECT	CONF_CAM0_HD_DSPULLUDEN	CONF_CAM0_HD_DS0OUTVALUE	CONF_CAM0_HD_DS0OUTEN	CONF_CAM0_HD_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_CAM0_HD_SLEWCTRL	CONF_CAM0_HD_RXACTIVE	CONF_CAM0_HD_PUTYPESEL	CONF_CAM0_HD_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_CAM0_HD_MMODE			
R/W-0h				R/W-7h			

Table 7-165. CTRL_CONF_CAM0_HD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM0_HD_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM0_HD_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM0_HD_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM0_HD_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM0_HD_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM0_HD_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM0_HD_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM0_HD_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM0_HD_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM0_HD_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-165. CTRL_CONF_CAM0_HD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM0_HD_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM0_HD_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.155 CTRL_CONF_CAM0_VD Register (Offset = 9B4h) [reset = X]

CTRL_CONF_CAM0_VD is shown in Figure 7-159 and described in Table 7-166.

Return to Summary Table.

Figure 7-159. CTRL_CONF_CAM0_VD Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM0_VD_WUEVT	CONF_CAM0_VD_WUEN	CONF_CAM0_VD_DSPULLTYPESELECT	CONF_CAM0_VD_DSPULLUDEN	CONF_CAM0_VD_DS0OUTVALUE	CONF_CAM0_VD_DS0OUTEN	CONF_CAM0_VD_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_CAM0_VD_SLEWCTRL	CONF_CAM0_VD_RXACTIVE	CONF_CAM0_VD_PUTYPESEL	CONF_CAM0_VD_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_CAM0_VD_MMODE			
R/W-0h				R/W-7h			

Table 7-166. CTRL_CONF_CAM0_VD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM0_VD_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM0_VD_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM0_VD_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM0_VD_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM0_VD_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM0_VD_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM0_VD_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM0_VD_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM0_VD_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM0_VD_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-166. CTRL_CONF_CAM0_VD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM0_VD_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM0_VD_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.156 CTRL_CONF_CAM0_FIELD Register (Offset = 9B8h) [reset = X]

CTRL_CONF_CAM0_FIELD is shown in Figure 7-160 and described in Table 7-167.

Return to Summary Table.

Figure 7-160. CTRL_CONF_CAM0_FIELD Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM0_FIELD_WUEVT	CONF_CAM0_FIELD_WUEN	CONF_CAM0_FIELD_DSPULLTYPESELECT	CONF_CAM0_FIELD_DSPULLUDEN	CONF_CAM0_FIELD_DS0OUTVALUE	CONF_CAM0_FIELD_DS0OUTEN	CONF_CAM0_FIELD_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_CAM0_FIELD_SLEWCTRL	CONF_CAM0_FIELD_RXACTIVE	CONF_CAM0_FIELD_PUTYPESEL	CONF_CAM0_FIELD_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_CAM0_FIELD_MMODE			
R/W-0h				R/W-7h			

Table 7-167. CTRL_CONF_CAM0_FIELD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM0_FIELD_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM0_FIELD_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM0_FIELD_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM0_FIELD_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM0_FIELD_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM0_FIELD_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM0_FIELD_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM0_FIELD_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM0_FIELD_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM0_FIELD_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-167. CTRL_CONF_CAM0_FIELD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM0_FIELD_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM0_FIELD_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.157 CTRL_CONF_CAM0_WEN Register (Offset = 9BCh) [reset = X]

CTRL_CONF_CAM0_WEN is shown in Figure 7-161 and described in Table 7-168.

Return to Summary Table.

Figure 7-161. CTRL_CONF_CAM0_WEN Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM0_WEN_WUEVT	CONF_CAM0_WEN_WUEN	CONF_CAM0_WEN_DSPULLTYPESELECT	CONF_CAM0_WEN_DSPULLUDEN	CONF_CAM0_WEN_DS0OUTVALUE	CONF_CAM0_WEN_DS0OUTEN	CONF_CAM0_WEN_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_CAM0_WEN_SLEWCTRL	CONF_CAM0_WEN_RXACTIVE	CONF_CAM0_WEN_PUTYPESEL	CONF_CAM0_WEN_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_CAM0_WEN_MMODE			
R/W-0h				R/W-7h			

Table 7-168. CTRL_CONF_CAM0_WEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM0_WEN_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM0_WEN_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM0_WEN_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM0_WEN_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM0_WEN_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM0_WEN_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM0_WEN_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM0_WEN_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM0_WEN_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM0_WEN_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-168. CTRL_CONF_CAM0_WEN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM0_WEN_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM0_WEN_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.158 CTRL_CONF_CAM0_PCLK Register (Offset = 9C0h) [reset = X]

CTRL_CONF_CAM0_PCLK is shown in Figure 7-162 and described in Table 7-169.

Return to Summary Table.

Figure 7-162. CTRL_CONF_CAM0_PCLK Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM0_PCLK_WUEVT	CONF_CAM0_PCLK_WUEN	CONF_CAM0_PCLK_DSPULLTYPESELECT	CONF_CAM0_PCLK_DSPULLUDEN	CONF_CAM0_PCLK_DS0OUTVALUE	CONF_CAM0_PCLK_DS0OUTEN	CONF_CAM0_PCLK_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_CAM0_PCLK_SLEWCTRL	CONF_CAM0_PCLK_RXACTIVE	CONF_CAM0_PCLK_PUTYPESEL	CONF_CAM0_PCLK_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_CAM0_PCLK_MMODE			
R/W-0h				R/W-7h			

Table 7-169. CTRL_CONF_CAM0_PCLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM0_PCLK_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM0_PCLK_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM0_PCLK_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM0_PCLK_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM0_PCLK_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM0_PCLK_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM0_PCLK_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM0_PCLK_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM0_PCLK_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM0_PCLK_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-169. CTRL_CONF_CAM0_PCLK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM0_PCLK_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM0_PCLK_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.159 CTRL_CONF_CAM0_DATA8 Register (Offset = 9C4h) [reset = X]

CTRL_CONF_CAM0_DATA8 is shown in Figure 7-163 and described in Table 7-170.

Return to Summary Table.

Figure 7-163. CTRL_CONF_CAM0_DATA8 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM0_DATA8_WUEVT	CONF_CAM0_DATA8_WUEN	CONF_CAM0_DATA8_DSPULLTYPESELECT	CONF_CAM0_DATA8_DSPULLUDEN	CONF_CAM0_DATA8_DS0OUTVALUE	CONF_CAM0_DATA8_DS0OUTEN	CONF_CAM0_DATA8_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_CAM0_DATA8_SLEWCTRL	CONF_CAM0_DATA8_RXACTIVE	CONF_CAM0_DATA8_PUTYPESEL	CONF_CAM0_DATA8_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_CAM0_DATA8_MMODE			
R/W-0h				R/W-7h			

Table 7-170. CTRL_CONF_CAM0_DATA8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM0_DATA8_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM0_DATA8_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM0_DATA8_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM0_DATA8_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM0_DATA8_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM0_DATA8_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM0_DATA8_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM0_DATA8_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM0_DATA8_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM0_DATA8_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-170. CTRL_CONF_CAM0_DATA8 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM0_DATA8_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM0_DATA8_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.160 CTRL_CONF_CAM0_DATA9 Register (Offset = 9C8h) [reset = X]

CTRL_CONF_CAM0_DATA9 is shown in Figure 7-164 and described in Table 7-171.

Return to Summary Table.

Figure 7-164. CTRL_CONF_CAM0_DATA9 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM0_DATA9_WUEVT	CONF_CAM0_DATA9_WUEN	CONF_CAM0_DATA9_DSPULLTYPESELECT	CONF_CAM0_DATA9_DSPULLUDEN	CONF_CAM0_DATA9_DS0OUTVALUE	CONF_CAM0_DATA9_DS0OUTEN	CONF_CAM0_DATA9_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_CAM0_DATA9_SLEWCTRL	CONF_CAM0_DATA9_RXACTIVE	CONF_CAM0_DATA9_PUTYPESEL	CONF_CAM0_DATA9_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_CAM0_DATA9_MMODE			
R/W-0h				R/W-7h			

Table 7-171. CTRL_CONF_CAM0_DATA9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM0_DATA9_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM0_DATA9_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM0_DATA9_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM0_DATA9_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM0_DATA9_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM0_DATA9_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM0_DATA9_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM0_DATA9_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM0_DATA9_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM0_DATA9_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-171. CTRL_CONF_CAM0_DATA9 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM0_DATA9_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM0_DATA9_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.161 CTRL_CONF_CAM1_DATA9 Register (Offset = 9CCh) [reset = X]

CTRL_CONF_CAM1_DATA9 is shown in Figure 7-165 and described in Table 7-172.

Return to Summary Table.

Figure 7-165. CTRL_CONF_CAM1_DATA9 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM1_DATA9_WUEVT	CONF_CAM1_DATA9_WUEN	CONF_CAM1_DATA9_DSPULLTYPESELECT	CONF_CAM1_DATA9_DSPULLUDEN	CONF_CAM1_DATA9_DS0OUTVALUE	CONF_CAM1_DATA9_DS0OUTEN	CONF_CAM1_DATA9_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_CAM1_DATA9_SLEWCTRL	CONF_CAM1_DATA9_RXACTIVE	CONF_CAM1_DATA9_PUTYPESEL	CONF_CAM1_DATA9_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_CAM1_DATA9_MMODE			
R/W-0h				R/W-7h			

Table 7-172. CTRL_CONF_CAM1_DATA9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM1_DATA9_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM1_DATA9_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM1_DATA9_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM1_DATA9_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM1_DATA9_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM1_DATA9_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM1_DATA9_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM1_DATA9_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM1_DATA9_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM1_DATA9_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-172. CTRL_CONF_CAM1_DATA9 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM1_DATA9_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM1_DATA9_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.162 CTRL_CONF_CAM1_DATA8 Register (Offset = 9D0h) [reset = X]

CTRL_CONF_CAM1_DATA8 is shown in Figure 7-166 and described in Table 7-173.

Return to Summary Table.

Figure 7-166. CTRL_CONF_CAM1_DATA8 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM1_DATA8_WUEVT	CONF_CAM1_DATA8_WUEN	CONF_CAM1_DATA8_DSPULLTYPESELECT	CONF_CAM1_DATA8_DSPULLUDEN	CONF_CAM1_DATA8_DS0OUTVALUE	CONF_CAM1_DATA8_DS0OUTEN	CONF_CAM1_DATA8_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_CAM1_DATA8_SLEWCTRL	CONF_CAM1_DATA8_RXACTIVE	CONF_CAM1_DATA8_PUTYPESEL	CONF_CAM1_DATA8_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_CAM1_DATA8_MMODE			
R/W-0h				R/W-7h			

Table 7-173. CTRL_CONF_CAM1_DATA8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM1_DATA8_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM1_DATA8_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM1_DATA8_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM1_DATA8_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM1_DATA8_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM1_DATA8_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM1_DATA8_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM1_DATA8_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM1_DATA8_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM1_DATA8_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-173. CTRL_CONF_CAM1_DATA8 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM1_DATA8_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM1_DATA8_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.163 CTRL_CONF_CAM1_HD Register (Offset = 9D4h) [reset = X]

CTRL_CONF_CAM1_HD is shown in Figure 7-167 and described in Table 7-174.

Return to Summary Table.

Figure 7-167. CTRL_CONF_CAM1_HD Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM1_HD_WUEVT	CONF_CAM1_HD_WUEN	CONF_CAM1_HD_DSPULLTYPESELECT	CONF_CAM1_HD_DSPULLUDEN	CONF_CAM1_HD_DS0OUTVALUE	CONF_CAM1_HD_DS0OUTEN	CONF_CAM1_HD_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_CAM1_HD_SLEWCTRL	CONF_CAM1_HD_RXACTIVE	CONF_CAM1_HD_PUTYPESEL	CONF_CAM1_HD_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_CAM1_HD_MMODE			
R/W-0h				R/W-7h			

Table 7-174. CTRL_CONF_CAM1_HD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM1_HD_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM1_HD_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM1_HD_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM1_HD_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM1_HD_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM1_HD_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM1_HD_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM1_HD_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM1_HD_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM1_HD_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-174. CTRL_CONF_CAM1_HD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM1_HD_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM1_HD_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.164 CTRL_CONF_CAM1_VD Register (Offset = 9D8h) [reset = X]

CTRL_CONF_CAM1_VD is shown in Figure 7-168 and described in Table 7-175.

Return to Summary Table.

Figure 7-168. CTRL_CONF_CAM1_VD Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM1_VD_WUEVT	CONF_CAM1_VD_WUEN	CONF_CAM1_VD_DSPULLTYPESELECT	CONF_CAM1_VD_DSPULLUDEN	CONF_CAM1_VD_DS0OUTVALUE	CONF_CAM1_VD_DS0OUTEN	CONF_CAM1_VD_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_CAM1_VD_SLEWCTRL	CONF_CAM1_VD_RXACTIVE	CONF_CAM1_VD_PUTYPESEL	CONF_CAM1_VD_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_CAM1_VD_MMODE			
R/W-0h				R/W-7h			

Table 7-175. CTRL_CONF_CAM1_VD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM1_VD_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM1_VD_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM1_VD_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM1_VD_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM1_VD_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM1_VD_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM1_VD_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM1_VD_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM1_VD_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM1_VD_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-175. CTRL_CONF_CAM1_VD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM1_VD_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM1_VD_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.165 CTRL_CONF_CAM1_PCLK Register (Offset = 9DCh) [reset = X]

CTRL_CONF_CAM1_PCLK is shown in Figure 7-169 and described in Table 7-176.

Return to Summary Table.

Figure 7-169. CTRL_CONF_CAM1_PCLK Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM1_PCLK_WUEVT	CONF_CAM1_PCLK_WUEN	CONF_CAM1_PCLK_DSPULLTYPESELECT	CONF_CAM1_PCLK_DSPULLUDEN	CONF_CAM1_PCLK_DS0OUTVALUE	CONF_CAM1_PCLK_DS0OUTEN	CONF_CAM1_PCLK_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_CAM1_PCLK_SLEWCTRL	CONF_CAM1_PCLK_RXACTIVE	CONF_CAM1_PCLK_PUTYPESEL	CONF_CAM1_PCLK_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_CAM1_PCLK_MMODE			
R/W-0h				R/W-7h			

Table 7-176. CTRL_CONF_CAM1_PCLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM1_PCLK_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM1_PCLK_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM1_PCLK_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM1_PCLK_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM1_PCLK_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM1_PCLK_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM1_PCLK_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM1_PCLK_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM1_PCLK_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM1_PCLK_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-176. CTRL_CONF_CAM1_PCLK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM1_PCLK_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM1_PCLK_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.166 CTRL_CONF_CAM1_FIELD Register (Offset = 9E0h) [reset = X]

CTRL_CONF_CAM1_FIELD is shown in Figure 7-170 and described in Table 7-177.

Return to Summary Table.

Figure 7-170. CTRL_CONF_CAM1_FIELD Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM1_FIELD_WUEVT	CONF_CAM1_FIELD_WUEN	CONF_CAM1_FIELD_DSPULLTYPESELECT	CONF_CAM1_FIELD_DSPULLUDEN	CONF_CAM1_FIELD_DS0OUTVALUE	CONF_CAM1_FIELD_DS0OUTEN	CONF_CAM1_FIELD_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_CAM1_FIELD_SLEWCTRL	CONF_CAM1_FIELD_RXACTIVE	CONF_CAM1_FIELD_PUTYPESEL	CONF_CAM1_FIELD_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_CAM1_FIELD_MMODE			
R/W-0h				R/W-7h			

Table 7-177. CTRL_CONF_CAM1_FIELD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM1_FIELD_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM1_FIELD_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM1_FIELD_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM1_FIELD_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM1_FIELD_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM1_FIELD_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM1_FIELD_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM1_FIELD_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM1_FIELD_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM1_FIELD_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-177. CTRL_CONF_CAM1_FIELD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM1_FIELD_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM1_FIELD_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.167 CTRL_CONF_CAM1_WEN Register (Offset = 9E4h) [reset = X]

CTRL_CONF_CAM1_WEN is shown in Figure 7-171 and described in Table 7-178.

Return to Summary Table.

Figure 7-171. CTRL_CONF_CAM1_WEN Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM1_WEN_WUEVT	CONF_CAM1_WEN_WUEN	CONF_CAM1_WEN_DSPULLTYPESELECT	CONF_CAM1_WEN_DSPULLUDEN	CONF_CAM1_WEN_DS0OUTVALUE	CONF_CAM1_WEN_DS0OUTEN	CONF_CAM1_WEN_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_CAM1_WEN_SLEWCTRL	CONF_CAM1_WEN_RXACTIVE	CONF_CAM1_WEN_PUTYPESEL	CONF_CAM1_WEN_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_CAM1_WEN_MMODE			
R/W-0h				R/W-7h			

Table 7-178. CTRL_CONF_CAM1_WEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM1_WEN_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM1_WEN_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM1_WEN_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM1_WEN_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM1_WEN_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM1_WEN_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM1_WEN_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM1_WEN_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM1_WEN_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM1_WEN_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-178. CTRL_CONF_CAM1_WEN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM1_WEN_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM1_WEN_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.168 CTRL_CONF_CAM1_DATA0 Register (Offset = 9E8h) [reset = X]

CTRL_CONF_CAM1_DATA0 is shown in Figure 7-172 and described in Table 7-179.

Return to Summary Table.

Figure 7-172. CTRL_CONF_CAM1_DATA0 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM1_DATA0_WUEVT	CONF_CAM1_DATA0_WUEN	CONF_CAM1_DATA0_DSPULLTYPESELECT	CONF_CAM1_DATA0_DSPULLUDEN	CONF_CAM1_DATA0_DS0OUTVALUE	CONF_CAM1_DATA0_DS0OUTEN	CONF_CAM1_DATA0_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_CAM1_DATA0_SLEWCTRL	CONF_CAM1_DATA0_RXACTIVE	CONF_CAM1_DATA0_PUTYPESEL	CONF_CAM1_DATA0_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_CAM1_DATA0_MMODE			
R/W-0h				R/W-7h			

Table 7-179. CTRL_CONF_CAM1_DATA0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM1_DATA0_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM1_DATA0_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM1_DATA0_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM1_DATA0_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM1_DATA0_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM1_DATA0_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM1_DATA0_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM1_DATA0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM1_DATA0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM1_DATA0_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-179. CTRL_CONF_CAM1_DATA0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM1_DATA0_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM1_DATA0_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.169 CTRL_CONF_CAM1_DATA1 Register (Offset = 9ECh) [reset = X]

CTRL_CONF_CAM1_DATA1 is shown in Figure 7-173 and described in Table 7-180.

Return to Summary Table.

Figure 7-173. CTRL_CONF_CAM1_DATA1 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM1_DATA1_WUEVT	CONF_CAM1_DATA1_WUEN	CONF_CAM1_DATA1_DSPULLTYPESELECT	CONF_CAM1_DATA1_DSPULLUDEN	CONF_CAM1_DATA1_DS0OUTVALUE	CONF_CAM1_DATA1_DS0OUTEN	CONF_CAM1_DATA1_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_CAM1_DATA1_SLEWCTRL	CONF_CAM1_DATA1_RXACTIVE	CONF_CAM1_DATA1_PUTYPESEL	CONF_CAM1_DATA1_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_CAM1_DATA1_MMODE			
R/W-0h				R/W-7h			

Table 7-180. CTRL_CONF_CAM1_DATA1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM1_DATA1_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM1_DATA1_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM1_DATA1_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM1_DATA1_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM1_DATA1_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM1_DATA1_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM1_DATA1_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM1_DATA1_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM1_DATA1_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM1_DATA1_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-180. CTRL_CONF_CAM1_DATA1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM1_DATA1_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM1_DATA1_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.170 CTRL_CONF_CAM1_DATA2 Register (Offset = 9F0h) [reset = X]

CTRL_CONF_CAM1_DATA2 is shown in Figure 7-174 and described in Table 7-181.

Return to Summary Table.

Figure 7-174. CTRL_CONF_CAM1_DATA2 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM1_DATA2_WUEVT	CONF_CAM1_DATA2_WUEN	CONF_CAM1_DATA2_DSPULLTYPESELECT	CONF_CAM1_DATA2_DSPULLUDEN	CONF_CAM1_DATA2_DS0OUTVALUE	CONF_CAM1_DATA2_DS0OUTEN	CONF_CAM1_DATA2_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_CAM1_DATA2_SLEWCTRL	CONF_CAM1_DATA2_RXACTIVE	CONF_CAM1_DATA2_PUTYPESEL	CONF_CAM1_DATA2_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_CAM1_DATA2_MMODE			
R/W-0h				R/W-7h			

Table 7-181. CTRL_CONF_CAM1_DATA2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM1_DATA2_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM1_DATA2_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM1_DATA2_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM1_DATA2_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM1_DATA2_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM1_DATA2_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM1_DATA2_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM1_DATA2_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM1_DATA2_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM1_DATA2_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-181. CTRL_CONF_CAM1_DATA2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM1_DATA2_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM1_DATA2_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.171 CTRL_CONF_CAM1_DATA3 Register (Offset = 9F4h) [reset = X]

CTRL_CONF_CAM1_DATA3 is shown in Figure 7-175 and described in Table 7-182.

Return to Summary Table.

Figure 7-175. CTRL_CONF_CAM1_DATA3 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM1_DATA3_WUEVT	CONF_CAM1_DATA3_WUEN	CONF_CAM1_DATA3_DSPULLTYPESELECT	CONF_CAM1_DATA3_DSPULLUDEN	CONF_CAM1_DATA3_DS0OUTVALUE	CONF_CAM1_DATA3_DS0OUTEN	CONF_CAM1_DATA3_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_CAM1_DATA3_SLEWCTRL	CONF_CAM1_DATA3_RXACTIVE	CONF_CAM1_DATA3_PUTYPESEL	CONF_CAM1_DATA3_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_CAM1_DATA3_MMODE			
R/W-0h				R/W-7h			

Table 7-182. CTRL_CONF_CAM1_DATA3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM1_DATA3_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM1_DATA3_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM1_DATA3_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM1_DATA3_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM1_DATA3_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM1_DATA3_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM1_DATA3_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM1_DATA3_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM1_DATA3_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM1_DATA3_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-182. CTRL_CONF_CAM1_DATA3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM1_DATA3_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM1_DATA3_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.172 CTRL_CONF_CAM1_DATA4 Register (Offset = 9F8h) [reset = X]

CTRL_CONF_CAM1_DATA4 is shown in Figure 7-176 and described in Table 7-183.

Return to Summary Table.

Figure 7-176. CTRL_CONF_CAM1_DATA4 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM1_DATA4_WUEVT	CONF_CAM1_DATA4_WUEN	CONF_CAM1_DATA4_DSPULLTYPESELECT	CONF_CAM1_DATA4_DSPULLUDEN	CONF_CAM1_DATA4_DS0OUTVALUE	CONF_CAM1_DATA4_DS0OUTEN	CONF_CAM1_DATA4_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_CAM1_DATA4_SLEWCTRL	CONF_CAM1_DATA4_RXACTIVE	CONF_CAM1_DATA4_PUTYPESEL	CONF_CAM1_DATA4_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_CAM1_DATA4_MMODE			
R/W-0h				R/W-7h			

Table 7-183. CTRL_CONF_CAM1_DATA4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM1_DATA4_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM1_DATA4_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM1_DATA4_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM1_DATA4_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM1_DATA4_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM1_DATA4_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM1_DATA4_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM1_DATA4_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM1_DATA4_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM1_DATA4_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-183. CTRL_CONF_CAM1_DATA4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM1_DATA4_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM1_DATA4_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.173 CTRL_CONF_CAM1_DATA5 Register (Offset = 9FCh) [reset = X]

CTRL_CONF_CAM1_DATA5 is shown in Figure 7-177 and described in Table 7-184.

Return to Summary Table.

Figure 7-177. CTRL_CONF_CAM1_DATA5 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM1_DATA5_WUEVT	CONF_CAM1_DATA5_WUEN	CONF_CAM1_DATA5_DSPULLTYPESELECT	CONF_CAM1_DATA5_DSPULLUDEN	CONF_CAM1_DATA5_DS0OUTVALUE	CONF_CAM1_DATA5_DS0OUTEN	CONF_CAM1_DATA5_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_CAM1_DATA5_SLEWCTRL	CONF_CAM1_DATA5_RXACTIVE	CONF_CAM1_DATA5_PUTYPESEL	CONF_CAM1_DATA5_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_CAM1_DATA5_MMODE			
R/W-0h				R/W-7h			

Table 7-184. CTRL_CONF_CAM1_DATA5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM1_DATA5_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM1_DATA5_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM1_DATA5_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM1_DATA5_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM1_DATA5_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM1_DATA5_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM1_DATA5_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM1_DATA5_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM1_DATA5_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM1_DATA5_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-184. CTRL_CONF_CAM1_DATA5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM1_DATA5_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM1_DATA5_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.174 CTRL_CONF_CAM1_DATA6 Register (Offset = A00h) [reset = X]

CTRL_CONF_CAM1_DATA6 is shown in Figure 7-178 and described in Table 7-185.

Return to Summary Table.

Figure 7-178. CTRL_CONF_CAM1_DATA6 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM1_DATA6_WUEVT	CONF_CAM1_DATA6_WUEN	CONF_CAM1_DATA6_DSPULLTYPESELECT	CONF_CAM1_DATA6_DSPULLUDEN	CONF_CAM1_DATA6_DS0OUTVALUE	CONF_CAM1_DATA6_DS0OUTEN	CONF_CAM1_DATA6_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_CAM1_DATA6_SLEWCTRL	CONF_CAM1_DATA6_RXACTIVE	CONF_CAM1_DATA6_PUTYPESEL	CONF_CAM1_DATA6_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_CAM1_DATA6_MMODE			
R/W-0h				R/W-7h			

Table 7-185. CTRL_CONF_CAM1_DATA6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM1_DATA6_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM1_DATA6_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM1_DATA6_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM1_DATA6_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM1_DATA6_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM1_DATA6_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM1_DATA6_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM1_DATA6_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM1_DATA6_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM1_DATA6_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-185. CTRL_CONF_CAM1_DATA6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM1_DATA6_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM1_DATA6_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.175 CTRL_CONF_CAM1_DATA7 Register (Offset = A04h) [reset = X]

CTRL_CONF_CAM1_DATA7 is shown in Figure 7-179 and described in Table 7-186.

Return to Summary Table.

Figure 7-179. CTRL_CONF_CAM1_DATA7 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM1_DATA7_WUEVT	CONF_CAM1_DATA7_WUEN	CONF_CAM1_DATA7_DSPULLTYPESELECT	CONF_CAM1_DATA7_DSPULLUDEN	CONF_CAM1_DATA7_DS0OUTVALUE	CONF_CAM1_DATA7_DS0OUTEN	CONF_CAM1_DATA7_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_CAM1_DATA7_SLEWCTRL	CONF_CAM1_DATA7_RXACTIVE	CONF_CAM1_DATA7_PUTYPESEL	CONF_CAM1_DATA7_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_CAM1_DATA7_MMODE			
R/W-0h				R/W-7h			

Table 7-186. CTRL_CONF_CAM1_DATA7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM1_DATA7_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM1_DATA7_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM1_DATA7_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM1_DATA7_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM1_DATA7_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM1_DATA7_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM1_DATA7_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM1_DATA7_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM1_DATA7_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM1_DATA7_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-186. CTRL_CONF_CAM1_DATA7 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM1_DATA7_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM1_DATA7_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.176 CTRL_CONF_CAM0_DATA0 Register (Offset = A08h) [reset = X]

CTRL_CONF_CAM0_DATA0 is shown in Figure 7-180 and described in Table 7-187.

Return to Summary Table.

Figure 7-180. CTRL_CONF_CAM0_DATA0 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM0_DATA0_WUEVT	CONF_CAM0_DATA0_WUEN	CONF_CAM0_DATA0_DSPULLTYPESELECT	CONF_CAM0_DATA0_DSPULLUDEN	CONF_CAM0_DATA0_DS0OUTVALUE	CONF_CAM0_DATA0_DS0OUTEN	CONF_CAM0_DATA0_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_CAM0_DATA0_SLEWCTRL	CONF_CAM0_DATA0_RXACTIVE	CONF_CAM0_DATA0_PUTYPESEL	CONF_CAM0_DATA0_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_CAM0_DATA0_MMODE			
R/W-0h				R/W-7h			

Table 7-187. CTRL_CONF_CAM0_DATA0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM0_DATA0_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM0_DATA0_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM0_DATA0_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM0_DATA0_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM0_DATA0_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM0_DATA0_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM0_DATA0_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM0_DATA0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM0_DATA0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM0_DATA0_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-187. CTRL_CONF_CAM0_DATA0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM0_DATA0_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM0_DATA0_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.177 CTRL_CONF_CAM0_DATA1 Register (Offset = A0Ch) [reset = X]

CTRL_CONF_CAM0_DATA1 is shown in Figure 7-181 and described in Table 7-188.

Return to Summary Table.

Figure 7-181. CTRL_CONF_CAM0_DATA1 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM0_DATA1_WUEVT	CONF_CAM0_DATA1_WUEN	CONF_CAM0_DATA1_DSPULLTYPESELECT	CONF_CAM0_DATA1_DSPULLUDEN	CONF_CAM0_DATA1_DS0OUTVALUE	CONF_CAM0_DATA1_DS0OUTEN	CONF_CAM0_DATA1_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_CAM0_DATA1_SLEWCTRL	CONF_CAM0_DATA1_RXACTIVE	CONF_CAM0_DATA1_PUTYPESEL	CONF_CAM0_DATA1_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_CAM0_DATA1_MMODE			
R/W-0h				R/W-7h			

Table 7-188. CTRL_CONF_CAM0_DATA1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM0_DATA1_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM0_DATA1_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM0_DATA1_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM0_DATA1_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM0_DATA1_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM0_DATA1_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM0_DATA1_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM0_DATA1_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM0_DATA1_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM0_DATA1_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-188. CTRL_CONF_CAM0_DATA1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM0_DATA1_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM0_DATA1_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.178 CTRL_CONF_CAM0_DATA2 Register (Offset = A10h) [reset = X]

CTRL_CONF_CAM0_DATA2 is shown in Figure 7-182 and described in Table 7-189.

Return to Summary Table.

Figure 7-182. CTRL_CONF_CAM0_DATA2 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM0_DATA2_WUEVT	CONF_CAM0_DATA2_WUEN	CONF_CAM0_DATA2_DSPULLTYPESELECT	CONF_CAM0_DATA2_DSPULLUDEN	CONF_CAM0_DATA2_DS0OUTVALUE	CONF_CAM0_DATA2_DS0OUTEN	CONF_CAM0_DATA2_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_CAM0_DATA2_SLEWCTRL	CONF_CAM0_DATA2_RXACTIVE	CONF_CAM0_DATA2_PUTYPESEL	CONF_CAM0_DATA2_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_CAM0_DATA2_MMODE			
R/W-0h				R/W-7h			

Table 7-189. CTRL_CONF_CAM0_DATA2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM0_DATA2_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM0_DATA2_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM0_DATA2_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM0_DATA2_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM0_DATA2_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM0_DATA2_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM0_DATA2_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM0_DATA2_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM0_DATA2_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM0_DATA2_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-189. CTRL_CONF_CAM0_DATA2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM0_DATA2_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM0_DATA2_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.179 CTRL_CONF_CAM0_DATA3 Register (Offset = A14h) [reset = X]

CTRL_CONF_CAM0_DATA3 is shown in Figure 7-183 and described in Table 7-190.

Return to Summary Table.

Figure 7-183. CTRL_CONF_CAM0_DATA3 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM0_DATA3_WUEVT	CONF_CAM0_DATA3_WUEN	CONF_CAM0_DATA3_DSPULLTYPESELECT	CONF_CAM0_DATA3_DSPULLUDEN	CONF_CAM0_DATA3_DS0OUTVALUE	CONF_CAM0_DATA3_DS0OUTEN	CONF_CAM0_DATA3_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_CAM0_DATA3_SLEWCTRL	CONF_CAM0_DATA3_RXACTIVE	CONF_CAM0_DATA3_PUTYPESEL	CONF_CAM0_DATA3_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_CAM0_DATA3_MMODE			
R/W-0h				R/W-7h			

Table 7-190. CTRL_CONF_CAM0_DATA3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM0_DATA3_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM0_DATA3_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM0_DATA3_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM0_DATA3_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM0_DATA3_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM0_DATA3_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM0_DATA3_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM0_DATA3_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM0_DATA3_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM0_DATA3_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-190. CTRL_CONF_CAM0_DATA3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM0_DATA3_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM0_DATA3_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.180 CTRL_CONF_CAM0_DATA4 Register (Offset = A18h) [reset = X]

CTRL_CONF_CAM0_DATA4 is shown in Figure 7-184 and described in Table 7-191.

Return to Summary Table.

Figure 7-184. CTRL_CONF_CAM0_DATA4 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM0_DATA4_WUEVT	CONF_CAM0_DATA4_WUEN	CONF_CAM0_DATA4_DSPULLTYPESELECT	CONF_CAM0_DATA4_DSPULLUDEN	CONF_CAM0_DATA4_DS0OUTVALUE	CONF_CAM0_DATA4_DS0OUTEN	CONF_CAM0_DATA4_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_CAM0_DATA4_SLEWCTRL	CONF_CAM0_DATA4_RXACTIVE	CONF_CAM0_DATA4_PUTYPESEL	CONF_CAM0_DATA4_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_CAM0_DATA4_MMODE			
R/W-0h				R/W-7h			

Table 7-191. CTRL_CONF_CAM0_DATA4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM0_DATA4_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM0_DATA4_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM0_DATA4_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM0_DATA4_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM0_DATA4_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM0_DATA4_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM0_DATA4_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM0_DATA4_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM0_DATA4_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM0_DATA4_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-191. CTRL_CONF_CAM0_DATA4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM0_DATA4_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM0_DATA4_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.181 CTRL_CONF_CAM0_DATA5 Register (Offset = A1Ch) [reset = X]

CTRL_CONF_CAM0_DATA5 is shown in Figure 7-185 and described in Table 7-192.

Return to Summary Table.

Figure 7-185. CTRL_CONF_CAM0_DATA5 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM0_DATA5_WUEVT	CONF_CAM0_DATA5_WUEN	CONF_CAM0_DATA5_DSPULLTYPESELECT	CONF_CAM0_DATA5_DSPULLUDEN	CONF_CAM0_DATA5_DS0OUTVALUE	CONF_CAM0_DATA5_DS0OUTEN	CONF_CAM0_DATA5_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_CAM0_DATA5_SLEWCTRL	CONF_CAM0_DATA5_RXACTIVE	CONF_CAM0_DATA5_PUTYPESEL	CONF_CAM0_DATA5_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_CAM0_DATA5_MMODE			
R/W-0h				R/W-7h			

Table 7-192. CTRL_CONF_CAM0_DATA5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM0_DATA5_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM0_DATA5_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM0_DATA5_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM0_DATA5_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM0_DATA5_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM0_DATA5_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM0_DATA5_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM0_DATA5_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM0_DATA5_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM0_DATA5_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-192. CTRL_CONF_CAM0_DATA5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM0_DATA5_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM0_DATA5_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.182 CTRL_CONF_CAM0_DATA6 Register (Offset = A20h) [reset = X]

CTRL_CONF_CAM0_DATA6 is shown in Figure 7-186 and described in Table 7-193.

Return to Summary Table.

Figure 7-186. CTRL_CONF_CAM0_DATA6 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM0_DATA6_WUEVT	CONF_CAM0_DATA6_WUEN	CONF_CAM0_DATA6_DSPULLTYPESELECT	CONF_CAM0_DATA6_DSPULLUDEN	CONF_CAM0_DATA6_DS0OUTVALUE	CONF_CAM0_DATA6_DS0OUTEN	CONF_CAM0_DATA6_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_CAM0_DATA6_SLEWCTRL	CONF_CAM0_DATA6_RXACTIVE	CONF_CAM0_DATA6_PUTYPESEL	CONF_CAM0_DATA6_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_CAM0_DATA6_MMODE			
R/W-0h				R/W-7h			

Table 7-193. CTRL_CONF_CAM0_DATA6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM0_DATA6_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM0_DATA6_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM0_DATA6_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM0_DATA6_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM0_DATA6_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM0_DATA6_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM0_DATA6_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM0_DATA6_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM0_DATA6_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM0_DATA6_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-193. CTRL_CONF_CAM0_DATA6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM0_DATA6_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM0_DATA6_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.183 CTRL_CONF_CAM0_DATA7 Register (Offset = A24h) [reset = X]

CTRL_CONF_CAM0_DATA7 is shown in Figure 7-187 and described in Table 7-194.

Return to Summary Table.

Figure 7-187. CTRL_CONF_CAM0_DATA7 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_CAM0_DATA7_WUEVT	CONF_CAM0_DATA7_WUEN	CONF_CAM0_DATA7_DSPULLTYPESELECT	CONF_CAM0_DATA7_DSPULLUDEN	CONF_CAM0_DATA7_DS0OUTVALUE	CONF_CAM0_DATA7_DS0OUTEN	CONF_CAM0_DATA7_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_CAM0_DATA7_SLEWCTRL	CONF_CAM0_DATA7_RXACTIVE	CONF_CAM0_DATA7_PUTYPESEL	CONF_CAM0_DATA7_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_CAM0_DATA7_MMODE			
R/W-0h				R/W-7h			

Table 7-194. CTRL_CONF_CAM0_DATA7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_CAM0_DATA7_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_CAM0_DATA7_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_CAM0_DATA7_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_CAM0_DATA7_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_CAM0_DATA7_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_CAM0_DATA7_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_CAM0_DATA7_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_CAM0_DATA7_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CAM0_DATA7_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CAM0_DATA7_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-194. CTRL_CONF_CAM0_DATA7 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_CAM0_DATA7_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CAM0_DATA7_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.184 CTRL_CONF_UART3_RXD Register (Offset = A28h) [reset = X]

CTRL_CONF_UART3_RXD is shown in Figure 7-188 and described in Table 7-195.

Return to Summary Table.

Figure 7-188. CTRL_CONF_UART3_RXD Register

31	30	29	28	27	26	25	24
RESERVED	CONF_UART3_RXD_WUEVT	CONF_UART3_RXD_WUEN	CONF_UART3_RXD_DSPULLTYPESELECT	CONF_UART3_RXD_DSPULLUDEN	CONF_UART3_RXD_DS0OUTVALUE	CONF_UART3_RXD_DS0OUTEN	CONF_UART3_RXD_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_UART3_RXD_SLEWCTRL	CONF_UART3_RXD_RXACTIVE	CONF_UART3_RXD_PUTYPESEL	CONF_UART3_RXD_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_UART3_RXD_MMODE			
R/W-0h				R/W-7h			

Table 7-195. CTRL_CONF_UART3_RXD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_UART3_RXD_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_UART3_RXD_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_UART3_RXD_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_UART3_RXD_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_UART3_RXD_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_UART3_RXD_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_UART3_RXD_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_UART3_RXD_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_UART3_RXD_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_UART3_RXD_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-195. CTRL_CONF_UART3_RXD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_UART3_RXD_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_UART3_RXD_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.185 CTRL_CONF_UART3_TXD Register (Offset = A2Ch) [reset = X]

CTRL_CONF_UART3_TXD is shown in Figure 7-189 and described in Table 7-196.

Return to Summary Table.

Figure 7-189. CTRL_CONF_UART3_TXD Register

31	30	29	28	27	26	25	24
RESERVED	CONF_UART3_TXD_WUEVT	CONF_UART3_TXD_WUEN	CONF_UART3_TXD_DSPULLTYPESELECT	CONF_UART3_TXD_DSPULLUDEN	CONF_UART3_TXD_DS0OUTVALUE	CONF_UART3_TXD_DS0OUTEN	CONF_UART3_TXD_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_UART3_TXD_SLEWCTRL	CONF_UART3_TXD_RXACTIVE	CONF_UART3_TXD_PUTYPESEL	CONF_UART3_TXD_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_UART3_TXD_MMODE			
R/W-0h				R/W-7h			

Table 7-196. CTRL_CONF_UART3_TXD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_UART3_TXD_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_UART3_TXD_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_UART3_TXD_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_UART3_TXD_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_UART3_TXD_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_UART3_TXD_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_UART3_TXD_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_UART3_TXD_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_UART3_TXD_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_UART3_TXD_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-196. CTRL_CONF_UART3_TXD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_UART3_TXD_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_UART3_TXD_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.186 CTRL_CONF_UART3_CTSN Register (Offset = A30h) [reset = X]

CTRL_CONF_UART3_CTSN is shown in Figure 7-190 and described in Table 7-197.

Return to Summary Table.

Figure 7-190. CTRL_CONF_UART3_CTSN Register

31	30	29	28	27	26	25	24
RESERVED	CONF_UART3_CTSN_WUEVT	CONF_UART3_CTSN_WUEN	CONF_UART3_CTSN_DSPULLTYPESELECT	CONF_UART3_CTSN_DSPULLUDEN	CONF_UART3_CTSN_DS0OUTVALUE	CONF_UART3_CTSN_DS0OUTEN	CONF_UART3_CTSN_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_UART3_CTSN_SLEWCTRL	CONF_UART3_CTSN_RXACTIVE	CONF_UART3_CTSN_PUTYPESEL	CONF_UART3_CTSN_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_UART3_CTSN_MMODE			
R/W-0h				R/W-7h			

Table 7-197. CTRL_CONF_UART3_CTSN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_UART3_CTSN_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_UART3_CTSN_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_UART3_CTSN_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_UART3_CTSN_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_UART3_CTSN_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_UART3_CTSN_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_UART3_CTSN_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_UART3_CTSN_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_UART3_CTSN_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_UART3_CTSN_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-197. CTRL_CONF_UART3_CTSN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_UART3_CTSN_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_UART3_CTSN_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.187 CTRL_CONF_UART3_RTSN Register (Offset = A34h) [reset = X]

CTRL_CONF_UART3_RTSN is shown in Figure 7-191 and described in Table 7-198.

Return to Summary Table.

Figure 7-191. CTRL_CONF_UART3_RTSN Register

31	30	29	28	27	26	25	24
RESERVED	CONF_UART3_RTSN_WUEVT	CONF_UART3_RTSN_WUEN	CONF_UART3_RTSN_DSPULLTYPESELECT	CONF_UART3_RTSN_DSPULLUDEN	CONF_UART3_RTSN_DS0OUTVALUE	CONF_UART3_RTSN_DS0OUTEN	CONF_UART3_RTSN_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_UART3_RTSN_SLEWCTRL	CONF_UART3_RTSN_RXACTIVE	CONF_UART3_RTSN_PUTYPESEL	CONF_UART3_RTSN_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_UART3_RTSN_MMODE			
R/W-0h				R/W-7h			

Table 7-198. CTRL_CONF_UART3_RTSN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_UART3_RTSN_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_UART3_RTSN_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_UART3_RTSN_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_UART3_RTSN_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_UART3_RTSN_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_UART3_RTSN_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_UART3_RTSN_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_UART3_RTSN_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_UART3_RTSN_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_UART3_RTSN_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-198. CTRL_CONF_UART3_RTSN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_UART3_RTSN_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_UART3_RTSN_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.188 CTRL_CONF_GPIO5_8 Register (Offset = A38h) [reset = X]

CTRL_CONF_GPIO5_8 is shown in Figure 7-192 and described in Table 7-199.

Return to Summary Table.

Figure 7-192. CTRL_CONF_GPIO5_8 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPIO5_8_WUEVT	CONF_GPIO5_8_WUEN	CONF_GPIO5_8_DSPULLTYPESELECT	CONF_GPIO5_8_DSPULLUDEN	CONF_GPIO5_8_DS0OUTVALUE	CONF_GPIO5_8_DS0OUTEN	CONF_GPIO5_8_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPIO5_8_SLEWCTRL	CONF_GPIO5_8_RXACTIVE	CONF_GPIO5_8_PUTYPESEL	CONF_GPIO5_8_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPIO5_8_MMODE			
R/W-0h				R/W-7h			

Table 7-199. CTRL_CONF_GPIO5_8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPIO5_8_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPIO5_8_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPIO5_8_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPIO5_8_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPIO5_8_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPIO5_8_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPIO5_8_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPIO5_8_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPIO5_8_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPIO5_8_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-199. CTRL_CONF_GPIO5_8 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPIO5_8_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPIO5_8_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.189 CTRL_CONF_GPIO5_9 Register (Offset = A3Ch) [reset = X]

CTRL_CONF_GPIO5_9 is shown in Figure 7-193 and described in Table 7-200.

Return to Summary Table.

Figure 7-193. CTRL_CONF_GPIO5_9 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPIO5_9_WUEVT	CONF_GPIO5_9_WUEN	CONF_GPIO5_9_DSPULLTYPESELECT	CONF_GPIO5_9_DSPULLUDEN	CONF_GPIO5_9_DS0OUTVALUE	CONF_GPIO5_9_DS0OUTEN	CONF_GPIO5_9_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_GPIO5_9_SLEWCTRL	CONF_GPIO5_9_RXACTIVE	CONF_GPIO5_9_PUTYPESEL	CONF_GPIO5_9_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_GPIO5_9_MMODE			
R/W-0h				R/W-7h			

Table 7-200. CTRL_CONF_GPIO5_9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPIO5_9_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPIO5_9_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPIO5_9_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPIO5_9_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPIO5_9_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPIO5_9_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPIO5_9_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPIO5_9_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPIO5_9_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPIO5_9_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-200. CTRL_CONF_GPIO5_9 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPIO5_9_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPIO5_9_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.190 CTRL_CONF_GPIO5_10 Register (Offset = A40h) [reset = X]

CTRL_CONF_GPIO5_10 is shown in Figure 7-194 and described in Table 7-201.

Return to Summary Table.

Figure 7-194. CTRL_CONF_GPIO5_10 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPIO5_10_WUEVT	CONF_GPIO5_10_WUEN	CONF_GPIO5_10_DSPULLTYPESELECT	CONF_GPIO5_10_DSPULLUDEN	CONF_GPIO5_10_DS0OUTVALUE	CONF_GPIO5_10_DS0OUTEN	CONF_GPIO5_10_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPIO5_10_SLEWCTRL	CONF_GPIO5_10_RXACTIVE	CONF_GPIO5_10_PUTYPESEL	CONF_GPIO5_10_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPIO5_10_MMODE			
R/W-0h				R/W-7h			

Table 7-201. CTRL_CONF_GPIO5_10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPIO5_10_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPIO5_10_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPIO5_10_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPIO5_10_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPIO5_10_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPIO5_10_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPIO5_10_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPIO5_10_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPIO5_10_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPIO5_10_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-201. CTRL_CONF_GPIO5_10 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPIO5_10_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPIO5_10_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.191 CTRL_CONF_GPIO5_11 Register (Offset = A44h) [reset = X]

CTRL_CONF_GPIO5_11 is shown in Figure 7-195 and described in Table 7-202.

Return to Summary Table.

Figure 7-195. CTRL_CONF_GPIO5_11 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPIO5_11_WUEVT	CONF_GPIO5_11_WUEN	CONF_GPIO5_11_DSPULLTYPESELECT	CONF_GPIO5_11_DSPULLUDEN	CONF_GPIO5_11_DS0OUTVALUE	CONF_GPIO5_11_DS0OUTEN	CONF_GPIO5_11_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPIO5_11_SLEWCTRL	CONF_GPIO5_11_RXACTIVE	CONF_GPIO5_11_PUTYPESEL	CONF_GPIO5_11_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPIO5_11_MMODE			
R/W-0h				R/W-7h			

Table 7-202. CTRL_CONF_GPIO5_11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPIO5_11_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPIO5_11_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPIO5_11_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPIO5_11_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPIO5_11_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPIO5_11_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPIO5_11_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPIO5_11_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPIO5_11_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPIO5_11_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-202. CTRL_CONF_GPIO5_11 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPIO5_11_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPIO5_11_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.192 CTRL_CONF_GPIO5_12 Register (Offset = A48h) [reset = X]

CTRL_CONF_GPIO5_12 is shown in Figure 7-196 and described in Table 7-203.

Return to Summary Table.

Figure 7-196. CTRL_CONF_GPIO5_12 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPIO5_12_WUEVT	CONF_GPIO5_12_WUEN	CONF_GPIO5_12_DSPULLTYPESELECT	CONF_GPIO5_12_DSPULLUDEN	CONF_GPIO5_12_DS0OUTVALUE	CONF_GPIO5_12_DS0OUTEN	CONF_GPIO5_12_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_GPIO5_12_SLEWCTRL	CONF_GPIO5_12_RXACTIVE	CONF_GPIO5_12_PUTYPESEL	CONF_GPIO5_12_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_GPIO5_12_MMODE			
R/W-0h				R/W-7h			

Table 7-203. CTRL_CONF_GPIO5_12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPIO5_12_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPIO5_12_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPIO5_12_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPIO5_12_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPIO5_12_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPIO5_12_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPIO5_12_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPIO5_12_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPIO5_12_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPIO5_12_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-203. CTRL_CONF_GPIO5_12 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPIO5_12_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPIO5_12_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.193 CTRL_CONF_GPIO5_13 Register (Offset = A4Ch) [reset = X]

CTRL_CONF_GPIO5_13 is shown in Figure 7-197 and described in Table 7-204.

Return to Summary Table.

Figure 7-197. CTRL_CONF_GPIO5_13 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_GPIO5_13_WUEVT	CONF_GPIO5_13_WUEN	CONF_GPIO5_13_DSPULLTYPESELECT	CONF_GPIO5_13_DSPULLUDEN	CONF_GPIO5_13_DS0OUTVALUE	CONF_GPIO5_13_DS0OUTEN	CONF_GPIO5_13_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_GPIO5_13_SLEWCTRL	CONF_GPIO5_13_RXACTIVE	CONF_GPIO5_13_PUTYPESEL	CONF_GPIO5_13_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_GPIO5_13_MMODE			
R/W-0h				R/W-7h			

Table 7-204. CTRL_CONF_GPIO5_13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_GPIO5_13_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_GPIO5_13_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_GPIO5_13_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_GPIO5_13_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_GPIO5_13_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_GPIO5_13_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_GPIO5_13_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_GPIO5_13_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_GPIO5_13_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_GPIO5_13_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-204. CTRL_CONF_GPIO5_13 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_GPIO5_13_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_GPIO5_13_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.194 CTRL_CONF_SPI4_SCLK Register (Offset = A50h) [reset = X]

CTRL_CONF_SPI4_SCLK is shown in Figure 7-198 and described in Table 7-205.

Return to Summary Table.

Figure 7-198. CTRL_CONF_SPI4_SCLK Register

31	30	29	28	27	26	25	24
RESERVED	CONF_SPI4_SCLK_WUEVT	CONF_SPI4_SCLK_WUEN	CONF_SPI4_SCLK_DSPULLTYPESELECT	CONF_SPI4_SCLK_DSPULLUDEN	CONF_SPI4_SCLK_DS0OUTVALUE	CONF_SPI4_SCLK_DS0OUTEN	CONF_SPI4_SCLK_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_SPI4_SCLK_SLEWCTRL	CONF_SPI4_SCLK_RXACTIVE	CONF_SPI4_SCLK_PUTYPESEL	CONF_SPI4_SCLK_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_SPI4_SCLK_MMODE			
R/W-0h				R/W-7h			

Table 7-205. CTRL_CONF_SPI4_SCLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_SPI4_SCLK_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_SPI4_SCLK_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_SPI4_SCLK_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_SPI4_SCLK_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_SPI4_SCLK_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_SPI4_SCLK_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_SPI4_SCLK_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_SPI4_SCLK_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_SPI4_SCLK_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_SPI4_SCLK_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-205. CTRL_CONF_SPI4_SCLK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_SPI4_SCLK_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_SPI4_SCLK_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.195 CTRL_CONF_SPI4_D0 Register (Offset = A54h) [reset = X]

CTRL_CONF_SPI4_D0 is shown in Figure 7-199 and described in Table 7-206.

Return to Summary Table.

Figure 7-199. CTRL_CONF_SPI4_D0 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_SPI4_D0_WUEVT	CONF_SPI4_D0_WUEN	CONF_SPI4_D0_DSPULLTYPESELECT	CONF_SPI4_D0_DSPULLUDEN	CONF_SPI4_D0_DS0OUTVALUE	CONF_SPI4_D0_DS0OUTEN	CONF_SPI4_D0_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_SPI4_D0_SLEWCTRL	CONF_SPI4_D0_RXACTIVE	CONF_SPI4_D0_PUTYPESEL	CONF_SPI4_D0_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_SPI4_D0_MMODE			
R/W-0h				R/W-7h			

Table 7-206. CTRL_CONF_SPI4_D0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_SPI4_D0_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_SPI4_D0_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_SPI4_D0_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_SPI4_D0_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_SPI4_D0_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_SPI4_D0_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_SPI4_D0_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_SPI4_D0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_SPI4_D0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_SPI4_D0_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-206. CTRL_CONF_SPI4_D0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_SPI4_D0_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_SPI4_D0_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.196 CTRL_CONF_SPI4_D1 Register (Offset = A58h) [reset = X]

CTRL_CONF_SPI4_D1 is shown in Figure 7-200 and described in Table 7-207.

Return to Summary Table.

Figure 7-200. CTRL_CONF_SPI4_D1 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_SPI4_D1_WUEVT	CONF_SPI4_D1_WUEN	CONF_SPI4_D1_DSPULLTYPESELECT	CONF_SPI4_D1_DSPULLUDEN	CONF_SPI4_D1_DS0OUTVALUE	CONF_SPI4_D1_DS0OUTEN	CONF_SPI4_D1_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_SPI4_D1_SLEWCTRL	CONF_SPI4_D1_RXACTIVE	CONF_SPI4_D1_PUTYPESEL	CONF_SPI4_D1_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_SPI4_D1_MMODE			
R/W-0h				R/W-7h			

Table 7-207. CTRL_CONF_SPI4_D1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_SPI4_D1_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_SPI4_D1_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_SPI4_D1_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_SPI4_D1_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_SPI4_D1_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_SPI4_D1_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_SPI4_D1_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_SPI4_D1_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_SPI4_D1_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_SPI4_D1_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-207. CTRL_CONF_SPI4_D1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_SPI4_D1_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_SPI4_D1_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.197 CTRL_CONF_SPI4_CS0 Register (Offset = A5Ch) [reset = X]

CTRL_CONF_SPI4_CS0 is shown in Figure 7-201 and described in Table 7-208.

Return to Summary Table.

Figure 7-201. CTRL_CONF_SPI4_CS0 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_SPI4_CS0_WUEVT	CONF_SPI4_CS0_WUEN	CONF_SPI4_CS0_DSPULLTYPESELECT	CONF_SPI4_CS0_DSPULLUDEN	CONF_SPI4_CS0_DS0OUTVALUE	CONF_SPI4_CS0_DS0OUTEN	CONF_SPI4_CS0_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_SPI4_CS0_SLEWCTRL	CONF_SPI4_CS0_RXACTIVE	CONF_SPI4_CS0_PUTYPESEL	CONF_SPI4_CS0_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_SPI4_CS0_MMODE			
R/W-0h				R/W-7h			

Table 7-208. CTRL_CONF_SPI4_CS0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_SPI4_CS0_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_SPI4_CS0_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_SPI4_CS0_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_SPI4_CS0_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_SPI4_CS0_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_SPI4_CS0_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_SPI4_CS0_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_SPI4_CS0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_SPI4_CS0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_SPI4_CS0_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-208. CTRL_CONF_SPI4_CS0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_SPI4_CS0_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_SPI4_CS0_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.198 CTRL_CONF_SPI2_SCLK Register (Offset = A60h) [reset = X]

CTRL_CONF_SPI2_SCLK is shown in Figure 7-202 and described in Table 7-209.

Return to Summary Table.

Figure 7-202. CTRL_CONF_SPI2_SCLK Register

31	30	29	28	27	26	25	24
RESERVED	CONF_SPI2_SCLK_WUEVT	CONF_SPI2_SCLK_WUEN	CONF_SPI2_SCLK_DSPULLTYPESELECT	CONF_SPI2_SCLK_DSPULLUDEN	CONF_SPI2_SCLK_DS0OUTVALUE	CONF_SPI2_SCLK_DS0OUTEN	CONF_SPI2_SCLK_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_SPI2_SCLK_SLEWCTRL	CONF_SPI2_SCLK_RXACTIVE	CONF_SPI2_SCLK_PUTYPESEL	CONF_SPI2_SCLK_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_SPI2_SCLK_MMODE			
R/W-0h				R/W-7h			

Table 7-209. CTRL_CONF_SPI2_SCLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_SPI2_SCLK_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_SPI2_SCLK_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_SPI2_SCLK_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_SPI2_SCLK_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_SPI2_SCLK_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_SPI2_SCLK_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_SPI2_SCLK_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_SPI2_SCLK_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_SPI2_SCLK_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_SPI2_SCLK_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-209. CTRL_CONF_SPI2_SCLK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_SPI2_SCLK_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_SPI2_SCLK_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.199 CTRL_CONF_SPI2_D0 Register (Offset = A64h) [reset = X]

CTRL_CONF_SPI2_D0 is shown in Figure 7-203 and described in Table 7-210.

Return to Summary Table.

Figure 7-203. CTRL_CONF_SPI2_D0 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_SPI2_D0_WUEVT	CONF_SPI2_D0_WUEN	CONF_SPI2_D0_DSPULLTYPESELECT	CONF_SPI2_D0_DSPULLUDEN	CONF_SPI2_D0_DS0OUTVALUE	CONF_SPI2_D0_DS0OUTEN	CONF_SPI2_D0_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_SPI2_D0_SLEWCTRL	CONF_SPI2_D0_RXACTIVE	CONF_SPI2_D0_PUTYPESEL	CONF_SPI2_D0_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_SPI2_D0_MMODE			
R/W-0h				R/W-7h			

Table 7-210. CTRL_CONF_SPI2_D0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_SPI2_D0_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_SPI2_D0_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_SPI2_D0_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_SPI2_D0_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_SPI2_D0_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_SPI2_D0_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_SPI2_D0_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_SPI2_D0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_SPI2_D0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_SPI2_D0_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-210. CTRL_CONF_SPI2_D0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_SPI2_D0_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_SPI2_D0_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.200 CTRL_CONF_SPI2_D1 Register (Offset = A68h) [reset = X]

CTRL_CONF_SPI2_D1 is shown in Figure 7-204 and described in Table 7-211.

Return to Summary Table.

Figure 7-204. CTRL_CONF_SPI2_D1 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_SPI2_D1_WUEVT	CONF_SPI2_D1_WUEN	CONF_SPI2_D1_DSPULLTYPESELECT	CONF_SPI2_D1_DSPULLUDEN	CONF_SPI2_D1_DS0OUTVALUE	CONF_SPI2_D1_DS0OUTEN	CONF_SPI2_D1_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_SPI2_D1_SLEWCTRL	CONF_SPI2_D1_RXACTIVE	CONF_SPI2_D1_PUTYPESEL	CONF_SPI2_D1_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_SPI2_D1_MMODE			
R/W-0h				R/W-7h			

Table 7-211. CTRL_CONF_SPI2_D1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_SPI2_D1_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_SPI2_D1_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_SPI2_D1_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_SPI2_D1_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_SPI2_D1_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_SPI2_D1_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_SPI2_D1_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_SPI2_D1_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_SPI2_D1_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_SPI2_D1_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-211. CTRL_CONF_SPI2_D1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_SPI2_D1_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_SPI2_D1_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.201 CTRL_CONF_SPI2_CS0 Register (Offset = A6Ch) [reset = X]

CTRL_CONF_SPI2_CS0 is shown in Figure 7-205 and described in Table 7-212.

Return to Summary Table.

Figure 7-205. CTRL_CONF_SPI2_CS0 Register

31	30	29	28	27	26	25	24
RESERVED	CONF_SPI2_CS0_WUEVT	CONF_SPI2_CS0_WUEN	CONF_SPI2_CS0_DSPULLTYPESELECT	CONF_SPI2_CS0_DSPULLUDEN	CONF_SPI2_CS0_DS0OUTVALUE	CONF_SPI2_CS0_DS0OUTEN	CONF_SPI2_CS0_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_SPI2_CS0_SLEWCTRL	CONF_SPI2_CS0_RXACTIVE	CONF_SPI2_CS0_PUTYPESEL	CONF_SPI2_CS0_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_SPI2_CS0_MMODE			
R/W-0h				R/W-7h			

Table 7-212. CTRL_CONF_SPI2_CS0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_SPI2_CS0_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_SPI2_CS0_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_SPI2_CS0_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_SPI2_CS0_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_SPI2_CS0_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_SPI2_CS0_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_SPI2_CS0_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_SPI2_CS0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_SPI2_CS0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_SPI2_CS0_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-212. CTRL_CONF_SPI2_CS0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_SPI2_CS0_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_SPI2_CS0_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.202 CTRL_CONF_XDMA_EVT_INTR0 Register (Offset = A70h) [reset = 00040007h]

Register mask: FFFFFFFFh

CTRL_CONF_XDMA_EVT_INTR0 is shown in Figure 7-206 and described in Table 7-213.

Return to Summary Table.

Figure 7-206. CTRL_CONF_XDMA_EVT_INTR0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td>CONF_XDMA_EVT_INTR0_SLEWCTRL</td><td>CONF_XDMA_EVT_INTR0_RXACTIVE</td><td>CONF_XDMA_EVT_INTR0_PUTYPESEL</td><td>CONF_XDMA_EVT_INTR0_PUDEN</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R/W-0h</td><td>R/W-1h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">CONF_XDMA_EVT_INTR0_MMODE</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-7h</td></tr>
</table>

Table 7-213. CTRL_CONF_XDMA_EVT_INTR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_XDMA_EVT_INTR0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_XDMA_EVT_INTR0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_XDMA_EVT_INTR0_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_XDMA_EVT_INTR0_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_XDMA_EVT_INTR0_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.203 CTRL_CONF_XDMA_EVT_INTR1 Register (Offset = A74h) [reset = 00040007h]

Register mask: FFFFFFFFh

CTRL_CONF_XDMA_EVT_INTR1 is shown in Figure 7-207 and described in Table 7-214.

Return to Summary Table.

Figure 7-207. CTRL_CONF_XDMA_EVT_INTR1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td>CONF_XDMA_EVT_INTR1_SLEWCTRL</td><td>CONF_XDMA_EVT_INTR1_RXACTIVE</td><td>CONF_XDMA_EVT_INTR1_PUTYPESEL</td><td>CONF_XDMA_EVT_INTR1_PUDEN</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R/W-0h</td><td>R/W-1h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">CONF_XDMA_EVT_INTR1_MMODE</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-7h</td></tr>
</table>

Table 7-214. CTRL_CONF_XDMA_EVT_INTR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_XDMA_EVT_INTR1_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_XDMA_EVT_INTR1_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_XDMA_EVT_INTR1_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_XDMA_EVT_INTR1_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_XDMA_EVT_INTR1_MMODE	R/W	7h	Pad Functional Signal Mux Select

7.3.1.204 CTRL_CONF_CLKREQ Register (Offset = A78h) [reset = 00060000h]

Register mask: FFFFFFFFh

CTRL_CONF_CLKREQ is shown in Figure 7-208 and described in Table 7-215.

Return to Summary Table.

Figure 7-208. CTRL_CONF_CLKREQ Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED				CONF_CLKREQ_SLEWCTRL	CONF_CLKREQ_RXACTIVE	CONF_CLKREQ_PUTYPESEL	CONF_CLKREQ_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_CLKREQ_MMODE			
R/W-0h				R/W-0h			

Table 7-215. CTRL_CONF_CLKREQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_CLKREQ_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_CLKREQ_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_CLKREQ_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_CLKREQ_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_CLKREQ_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.205 CTRL_CONF_NRESETIN_OUT Register (Offset = A7Ch) [reset = 00060000h]

Register mask: FFFFFFFFh

CTRL_CONF_NRESETIN_OUT is shown in Figure 7-209 and described in Table 7-216.

Return to Summary Table.

Figure 7-209. CTRL_CONF_NRESETIN_OUT Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED				CONF_NRESETIN_OUT_SLEWCTRL	CONF_NRESETIN_OUT_RXACTIVE	CONF_NRESETIN_OUT_PUTYPESEL	CONF_NRESETIN_OUT_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_NRESETIN_OUT_MMODE			
R/W-0h				R/W-0h			

Table 7-216. CTRL_CONF_NRESETIN_OUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_NRESETIN_OUT_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_NRESETIN_OUT_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_NRESETIN_OUT_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_NRESETIN_OUT_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_NRESETIN_OUT_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.206 CTRL_CONF_NNMI Register (Offset = A84h) [reset = 00060000h]

Register mask: FFFFFFFFh

CTRL_CONF_NNMI is shown in Figure 7-210 and described in Table 7-217.

Return to Summary Table.

Figure 7-210. CTRL_CONF_NNMI Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							

23	22	21	20	19	18	17	16
RESERVED				CONF_NNMI_ SLEWCTRL	CONF_NNMI_ RXACTIVE	CONF_NNMI_ PUTYPESEL	CONF_NNMI_ PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_NNMI_MMODE			
R/W-0h				R/W-0h			

Table 7-217. CTRL_CONF_NNMI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_NNMI_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_NNMI_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_NNMI_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_NNMI_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_NNMI_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.207 CTRL_CONF_TMS Register (Offset = A90h) [reset = 00060000h]

Register mask: FFFFFFFFh

CTRL_CONF_TMS is shown in Figure 7-211 and described in Table 7-218.

Return to Summary Table.

Figure 7-211. CTRL_CONF_TMS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td>CONF_TMS_SLEWCTRL</td><td>CONF_TMS_RXACTIVE</td><td>CONF_TMS_PUTYPESEL</td><td>CONF_TMS_PUDEN</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R/W-0h</td><td>R/W-1h</td><td>R/W-1h</td><td>R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">CONF_TMS_MMODE</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 7-218. CTRL_CONF_TMS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_TMS_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_TMS_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_TMS_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_TMS_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_TMS_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.208 CTRL_CONF_TDI Register (Offset = A94h) [reset = 00060000h]

Register mask: FFFFFFFFh

CTRL_CONF_TDI is shown in Figure 7-212 and described in Table 7-219.

Return to Summary Table.

Figure 7-212. CTRL_CONF_TDI Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td>CONF_TDI_SLEWCTRL</td><td>CONF_TDI_RXACTIVE</td><td>CONF_TDI_PUTYPESEL</td><td>CONF_TDI_PUDEN</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R/W-0h</td><td>R/W-1h</td><td>R/W-1h</td><td>R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">CONF_TDI_MMODE</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 7-219. CTRL_CONF_TDI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_TDI_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_TDI_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_TDI_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_TDI_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_TDI_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.209 CTRL_CONF_TDO Register (Offset = A98h) [reset = 00060000h]

Register mask: FFFFFFFFh

CTRL_CONF_TDO is shown in Figure 7-213 and described in Table 7-220.

Return to Summary Table.

Figure 7-213. CTRL_CONF_TDO Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED				CONF_TDO_SLEWCTRL	CONF_TDO_RXACTIVE	CONF_TDO_PUTYPESEL	CONF_TDO_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_TDO_MMODE			
R/W-0h				R/W-0h			

Table 7-220. CTRL_CONF_TDO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_TDO_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_TDO_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_TDO_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_TDO_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_TDO_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.210 CTRL_CONF_TCK Register (Offset = A9Ch) [reset = 00060000h]

Register mask: FFFFFFFFh

CTRL_CONF_TCK is shown in Figure 7-214 and described in Table 7-221.

Return to Summary Table.

Figure 7-214. CTRL_CONF_TCK Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_TCK_SLEWCTRL	CONF_TCK_RXACTIVE	CONF_TCK_PUTYPESEL	CONF_TCK_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_TCK_MMODE			
R/W-0h				R/W-0h			

Table 7-221. CTRL_CONF_TCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_TCK_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_TCK_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_TCK_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_TCK_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_TCK_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.211 CTRL_CONF_NTRST Register (Offset = AA0h) [reset = 00040000h]

Register mask: FFFFFFFFh

CTRL_CONF_NTRST is shown in Figure 7-215 and described in Table 7-222.

Return to Summary Table.

Figure 7-215. CTRL_CONF_NTRST Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED				CONF_NTRST_SLEWCTRL	CONF_NTRST_RXACTIVE	CONF_NTRST_PUTYPESEL	CONF_NTRST_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_NTRST_MMODE			
R/W-0h				R/W-0h			

Table 7-222. CTRL_CONF_NTRST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_NTRST_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_NTRST_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_NTRST_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_NTRST_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_NTRST_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.212 CTRL_CONF_EMU0 Register (Offset = AA4h) [reset = 00060000h]

Register mask: FFFFFFFFh

CTRL_CONF_EMU0 is shown in Figure 7-216 and described in Table 7-223.

Return to Summary Table.

Figure 7-216. CTRL_CONF_EMU0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td>CONF_EMU0_SLEWCTRL</td><td>CONF_EMU0_RXACTIVE</td><td>CONF_EMU0_PUTYPESEL</td><td>CONF_EMU0_PUDEN</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R/W-0h</td><td>R/W-1h</td><td>R/W-1h</td><td>R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">CONF_EMU0_MMODE</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 7-223. CTRL_CONF_EMU0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_EMU0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_EMU0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_EMU0_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_EMU0_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_EMU0_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.213 CTRL_CONF_EMU1 Register (Offset = AA8h) [reset = 00060000h]

Register mask: FFFFFFFFh

CTRL_CONF_EMU1 is shown in Figure 7-217 and described in Table 7-224.

Return to Summary Table.

Figure 7-217. CTRL_CONF_EMU1 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED				CONF_EMU1_SLEWCTRL	CONF_EMU1_RXACTIVE	CONF_EMU1_PUTYPESEL	CONF_EMU1_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_EMU1_MMODE			
R/W-0h				R/W-0h			

Table 7-224. CTRL_CONF_EMU1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_EMU1_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_EMU1_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_EMU1_PUTYPESEL	R/W	1h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_EMU1_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_EMU1_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.214 CTRL_CONF_OSC1_IN Register (Offset = AACh) [reset = 00050000h]

Register mask: FFFFFFFFh

CTRL_CONF_OSC1_IN is shown in Figure 7-218 and described in Table 7-225.

Return to Summary Table.

Figure 7-218. CTRL_CONF_OSC1_IN Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_OSC1_IN_SLEWCTRL	CONF_OSC1_IN_RXACTIVE	CONF_OSC1_IN_PUTYPESEL	CONF_OSC1_IN_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_OSC1_IN_MMODE			
R/W-0h				R/W-0h			

Table 7-225. CTRL_CONF_OSC1_IN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_OSC1_IN_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_OSC1_IN_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_OSC1_IN_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_OSC1_IN_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_OSC1_IN_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.215 CTRL_CONF_OSC1_OUT Register (Offset = AB0h) [reset = 00050000h]

Register mask: FFFFFFFFh

CTRL_CONF_OSC1_OUT is shown in Figure 7-219 and described in Table 7-226.

Return to Summary Table.

Figure 7-219. CTRL_CONF_OSC1_OUT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td>CONF_OSC1_OUT_SLEWCTRL</td><td>CONF_OSC1_OUT_RXACTIVE</td><td>CONF_OSC1_OUT_PUTYPESEL</td><td>CONF_OSC1_OUT_PUDEN</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R/W-0h</td><td>R/W-1h</td><td>R/W-0h</td><td>R/W-1h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">CONF_OSC1_OUT_MMODE</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 7-226. CTRL_CONF_OSC1_OUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_OSC1_OUT_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_OSC1_OUT_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_OSC1_OUT_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_OSC1_OUT_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_OSC1_OUT_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.216 CTRL_CONF_RTC_PORZ Register (Offset = AB4h) [reset = 00050000h]

Register mask: FFFFFFFFh

CTRL_CONF_RTC_PORZ is shown in Figure 7-220 and described in Table 7-227.

Return to Summary Table.

Figure 7-220. CTRL_CONF_RTC_PORZ Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_RTC_PORZ_SLEWCTRL	CONF_RTC_PORZ_RXACTIVE	CONF_RTC_PORZ_PUTYPESEL	CONF_RTC_PORZ_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_RTC_PORZ_MMODE			
R/W-0h				R/W-0h			

Table 7-227. CTRL_CONF_RTC_PORZ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_RTC_PORZ_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_RTC_PORZ_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_RTC_PORZ_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_RTC_PORZ_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_RTC_PORZ_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.217 CTRL_CONF_EXT_WAKEUP0 Register (Offset = AB8h) [reset = 00050000h]

Register mask: FFFFFFFFh

CTRL_CONF_EXT_WAKEUP0 is shown in Figure 7-221 and described in Table 7-228.

Return to Summary Table.

Figure 7-221. CTRL_CONF_EXT_WAKEUP0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td>CONF_EXT_WAKEUP0_SLEWCTRL</td><td>CONF_EXT_WAKEUP0_RXACTIVE</td><td>CONF_EXT_WAKEUP0_PUTYPESEL</td><td>CONF_EXT_WAKEUP0_PUDEN</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R/W-0h</td><td>R/W-1h</td><td>R/W-0h</td><td>R/W-1h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">CONF_EXT_WAKEUP0_MMODE</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 7-228. CTRL_CONF_EXT_WAKEUP0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_EXT_WAKEUP0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_EXT_WAKEUP0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_EXT_WAKEUP0_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_EXT_WAKEUP0_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_EXT_WAKEUP0_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.218 CTRL_CONF_PMIC_POWER_EN0 Register (Offset = ABCh) [reset = 00050000h]

Register mask: FFFFFFFFh

CTRL_CONF_PMIC_POWER_EN0 is shown in Figure 7-222 and described in Table 7-229.

Return to Summary Table.

Figure 7-222. CTRL_CONF_PMIC_POWER_EN0 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				CONF_PMIC_POWER_EN0_SLEWCTRL	CONF_PMIC_POWER_EN0_RXACTIVE	CONF_PMIC_POWER_EN0_PUTYPESEL	CONF_PMIC_POWER_EN0_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-1h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CONF_PMIC_POWER_EN0_MMODE			
R/W-0h				R/W-0h			

Table 7-229. CTRL_CONF_PMIC_POWER_EN0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	CONF_PMIC_POWER_EN0_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_PMIC_POWER_EN0_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_PMIC_POWER_EN0_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected
16	CONF_PMIC_POWER_EN0_PUDEN	R/W	1h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_PMIC_POWER_EN0_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.219 CTRL_CONF_USB0_DRVVBUS Register (Offset = AC0h) [reset = X]

CTRL_CONF_USB0_DRVVBUS is shown in Figure 7-223 and described in Table 7-230.

Return to Summary Table.

Figure 7-223. CTRL_CONF_USB0_DRVVBUS Register

31	30	29	28	27	26	25	24
RESERVED	CONF_USB0_DRVVBUS_WUEVT	CONF_USB0_DRVVBUS_WUEN	CONF_USB0_DRVVBUS_DSPULLTYPESELECT	CONF_USB0_DRVVBUS_DSPULLUDEN	CONF_USB0_DRVVBUS_DS0OUTVALUE	CONF_USB0_DRVVBUS_DS0OUTEN	CONF_USB0_DRVVBUS_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED				CONF_USB0_DRVVBUS_SLEWCTRL	CONF_USB0_DRVVBUS_RXACTIVE	CONF_USB0_DRVVBUS_PUTYPESEL	CONF_USB0_DRVVBUS_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				CONF_USB0_DRVVBUS_MMODE			
R/W-0h				R/W-0h			

Table 7-230. CTRL_CONF_USB0_DRVVBUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_USB0_DRVVBUS_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_USB0_DRVVBUS_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_USB0_DRVVBUS_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_USB0_DRVVBUS_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_USB0_DRVVBUS_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_USB0_DRVVBUS_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_USB0_DRVVBUS_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_USB0_DRVVBUS_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_USB0_DRVVBUS_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_USB0_DRVVBUS_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-230. CTRL_CONF_USB0_DRVVBUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_USB0_DRVVBUS_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_USB0_DRVVBUS_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.220 CTRL_CONF_USB1_DRVVBUS Register (Offset = AC4h) [reset = X]

CTRL_CONF_USB1_DRVVBUS is shown in Figure 7-224 and described in Table 7-231.

Return to Summary Table.

Figure 7-224. CTRL_CONF_USB1_DRVVBUS Register

31	30	29	28	27	26	25	24
RESERVED	CONF_USB1_DRVVBUS_WUEVT	CONF_USB1_DRVVBUS_WUEN	CONF_USB1_DRVVBUS_DSPULLTYPESELECT	CONF_USB1_DRVVBUS_DSPULLUDEN	CONF_USB1_DRVVBUS_DS0OUTVALUE	CONF_USB1_DRVVBUS_DS0OUTEN	CONF_USB1_DRVVBUS_DS0EN
R/W-0h	R-X	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				CONF_USB1_DRVVBUS_SLEWCTRL	CONF_USB1_DRVVBUS_RXACTIVE	CONF_USB1_DRVVBUS_PUTYPESEL	CONF_USB1_DRVVBUS_PUDEN
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CONF_USB1_DRVVBUS_MMODE			
R/W-0h				R/W-0h			

Table 7-231. CTRL_CONF_USB1_DRVVBUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	CONF_USB1_DRVVBUS_WUEVT	R	X	Wakeup Event 0 = No event 1 = Event occurred
29	CONF_USB1_DRVVBUS_WUEN	R/W	0h	Wakeup enable 0 = disable 1 = enable
28	CONF_USB1_DRVVBUS_DSPULLTYPESELECT	R/W	0h	DS0 mode Pull-Up/Down selection 0 = Offmode Pull-Down selected 1 = Offmode Pull-Up selected
27	CONF_USB1_DRVVBUS_DSPULLUDEN	R/W	1h	DS0 mode Pull-Up/Down enable. This is an active low signal. 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
26	CONF_USB1_DRVVBUS_DS0OUTVALUE	R/W	0h	DS0 mode output value 0 = Set value at 0 1 = Set value at 1
25	CONF_USB1_DRVVBUS_DS0OUTEN	R/W	0h	DS0 mode output enable This is an active low signal 0 = Output Enable 1 = Output Disable
24	CONF_USB1_DRVVBUS_DS0EN	R/W	0h	DS0 mode override control 0: IO state keeps its previous state when DS0 mode is active 1: IO state is forced to OFF mode value when DS0 mode is active
23-20	RESERVED	R/W	0h	
19	CONF_USB1_DRVVBUS_SLEWCTRL	R/W	0h	Select between Faster or Slower Slew rate 0 = fast 1 = slow
18	CONF_USB1_DRVVBUS_RXACTIVE	R/W	1h	Input Enable Value for the PAD 0 = receiver disabled 1 = receiver enabled
17	CONF_USB1_DRVVBUS_PUTYPESEL	R/W	0h	Pad Pullup / Pulldown Type Selection 0 = Pulldown selected 1 = Pullup selected

Table 7-231. CTRL_CONF_USB1_DRVVBUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CONF_USB1_DRVVBUS_PUDEN	R/W	0h	Pad Pullup / Pulldown Enable. This is an active low signal 1 = Pullup / Pulldown disabled 0 = Pullup / Pulldown enabled
15-4	RESERVED	R/W	0h	
3-0	CONF_USB1_DRVVBUS_MMODE	R/W	0h	Pad Functional Signal Mux Select

7.3.1.221 CTRL_CQDETECT_STS Register (Offset = E00h) [reset = X]

CTRL_CQDETECT_STS is shown in Figure 7-225 and described in Table 7-232.

Return to Summary Table.

Figure 7-225. CTRL_CQDETECT_STS Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CQMODE_CAMERA	CQMODE_LCDC	CQMODE_GENERAL	CQMODE_GEMAC_B	CQMODE_GEMAC_A	CQMODE_MMCSD_B	CQMODE_MMCSD_A	CQMODE_GPMC
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CQERR_CAMERA	CQERR_LCDC	CQERR_GENERAL	CQERR_GEMAC_B	CQERR_GEMAC_A	CQERR_MMCSD_B	CQERR_MMCSD_A	CQERR_GPMC
R-X	R-X	R-X	R-X	R-X	R-X	R-X	R-X
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CQSTAT_CAMERA	CQSTAT_LCDC	CQSTAT_GENERAL	CQSTAT_GEMAC_B	CQSTAT_GEMAC_A	CQSTAT_MMCSD_B	CQSTAT_MMCSD_A	CQSTAT_GPMC
R-X	R-X	R-X	R-X	R-X	R-X	R-X	R-X

Table 7-232. CTRL_CQDETECT_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23	CQMODE_CAMERA	R/W	0h	0 : Set IO to 1.8V 1 : Set IO to 3.3V
22	CQMODE_LCDC	R/W	0h	0 : Set IO to 1.8V 1 : Set IO to 3.3V
21	CQMODE_GENERAL	R/W	0h	0 : Set IO to 1.8V 1 : Set IO to 3.3V
20	CQMODE_GEMAC_B	R/W	0h	0 : Set IO to 1.8V 1 : Set IO to 3.3V
19	CQMODE_GEMAC_A	R/W	0h	0 : Set IO to 1.8V 1 : Set IO to 3.3V
18	CQMODE_MMCSD_B	R/W	0h	0 : Set IO to 1.8V 1 : Set IO to 3.3V
17	CQMODE_MMCSD_A	R/W	0h	0 : Set IO to 1.8V 1 : Set IO to 3.3V
16	CQMODE_GPMC	R/W	0h	0 : Set IO to 1.8V 1 : Set IO to 3.3V
15	CQERR_CAMERA	R	X	CQDetect Mode Error Status
14	CQERR_LCDC	R	X	CQDetect Mode Error Status
13	CQERR_GENERAL	R	X	CQDetect Mode Error Status
12	CQERR_GEMAC_B	R	X	CQDetect Mode Error Status
11	CQERR_GEMAC_A	R	X	CQDetect Mode Error Status
10	CQERR_MMCSD_B	R	X	CQDetect Mode Error Status
9	CQERR_MMCSD_A	R	X	CQDetect Mode Error Status
8	CQERR_GPMC	R	X	CQDetect Mode Error Status
7	CQSTAT_CAMERA	R	X	1 : IOs are 3.3V mode 0 : IOs are 1.8V mode
6	CQSTAT_LCDC	R	X	1 : IOs are 3.3V mode 0 : IOs are 1.8V mode

Table 7-232. CTRL_CQDETECT_STS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	CQSTAT_GENERAL	R	X	1 : IOs are 3.3V mode 0 : IOs are 1.8V mode
4	CQSTAT_GEMAC_B	R	X	1 : IOs are 3.3V mode 0 : IOs are 1.8V mode
3	CQSTAT_GEMAC_A	R	X	1 : IOs are 3.3V mode 0 : IOs are 1.8V mode
2	CQSTAT_MMCSD_B	R	X	1 : IOs are 3.3V mode 0 : IOs are 1.8V mode
1	CQSTAT_MMCSD_A	R	X	1 : IOs are 3.3V mode 0 : IOs are 1.8V mode
0	CQSTAT_GPMC	R	X	1 : IOs are 3.3V mode 0 : IOs are 1.8V mode

7.3.1.222 CTRL_DDR_IO Register (Offset = E04h) [reset = 80000000h]

Register mask: FFFFFFFFh

CTRL_DDR_IO is shown in Figure 7-226 and described in Table 7-233.

Return to Summary Table.

Figure 7-226. CTRL_DDR_IO Register

31	30	29	28	27	26	25	24
DDR3_RST_DEF_VAL	RESERVED						
R/W-1h	R/W-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							
R/W-0h							

Table 7-233. CTRL_DDR_IO Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DDR3_RST_DEF_VAL	R/W	1h	0 = DDR3 RESET controlled via controller/PHY 1 = DDR RESET is overridden to value HIGH As part of boot initialization (& before DDR/EMIF init) this bit must be written to value zero.
30-0	RESERVED	R/W	0h	

7.3.1.223 CTRL_CQDETECT_STS2 Register (Offset = E08h) [reset = X]

CTRL_CQDETECT_STS2 is shown in Figure 7-227 and described in Table 7-234.

Return to Summary Table.

Figure 7-227. CTRL_CQDETECT_STS2 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED						CQMODE_MDIO	CQMODE_CLKOUT
R/W-0h						R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED						CQERR_MDIO	CQERR_CLKOUT
R/W-0h						R-X	R-X
7	6	5	4	3	2	1	0
RESERVED						CQSTAT_MDIO	CQSTAT_CLKOUT
R/W-0h						R-X	R-X

Table 7-234. CTRL_CQDETECT_STS2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17	CQMODE_MDIO	R/W	0h	0 : Set IO to 1.8V 1 : Set IO to 3.3V
16	CQMODE_CLKOUT	R/W	0h	0 : Set IO to 1.8V 1 : Set IO to 3.3V
15-10	RESERVED	R/W	0h	
9	CQERR_MDIO	R	X	CQDetect Mode Error Status
8	CQERR_CLKOUT	R	X	CQDetect Mode Error Status
7-2	RESERVED	R/W	0h	
1	CQSTAT_MDIO	R	X	1 : IOs are 3.3V mode 0 : IOs are 1.8V mode
0	CQSTAT_CLKOUT	R	X	1 : IOs are 3.3V mode 0 : IOs are 1.8V mode

7.3.1.224 CTRL_VTP Register (Offset = E0Ch) [reset = X]

CTRL_VTP is shown in Figure 7-228 and described in Table 7-235.

Return to Summary Table.

Figure 7-228. CTRL_VTP Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>RESERVED</td><td colspan="7">PCIN</td></tr>
<tr><td>R/W-0h</td><td colspan="7">R/W-X</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>RESERVED</td><td colspan="7">NCIN</td></tr>
<tr><td>R/W-0h</td><td colspan="7">R/W-X</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>RESERVED</td><td>EN</td><td>READY</td><td>LOCK</td><td colspan="3">FILTER</td><td>CLRZ</td></tr>
<tr><td>R/W-0h</td><td>R/W-0h</td><td>R-X</td><td>R/W-0h</td><td colspan="3">R/W-3h</td><td>R/W-1h</td></tr>
</table>

Table 7-235. CTRL_VTP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	0h	
22-16	PCIN	R/W	X	Default/reset values of 'P' for the VTP controller.
15	RESERVED	R/W	0h	
14-8	NCIN	R/W	X	Default/reset values of 'N' for the VTP controller.
7	RESERVED	R/W	0h	
6	EN	R/W	0h	active high enable
5	READY	R	X	0: Training sequence is not complete. 1: Training sequence is complete.
4	LOCK	R/W	0h	0: Normal operation dynamic update 1: freeze dynamic update pwrdn controller
3-1	FILTER	R/W	3h	Digital filter bits
0	CLRZ	R/W	1h	clears flops start count again after low going pulse

7.3.1.225 CTRL_VREF Register (Offset = E14h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_VREF is shown in Figure 7-229 and described in Table 7-236.

Return to Summary Table.

Figure 7-229. CTRL_VREF Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED			DDR_VREF_CCAP		DDR_VREF_TAP		DDR_VREF_EN
R/W-0h			R/W-0h		R/W-0h		R/W-0h

Table 7-236. CTRL_VREF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4-3	DDR_VREF_CCAP	R/W	0h	select for coupling cap for DDR 00 : No capacitor connected 01 : Capacitor between BIAS2 and VSS 10 : Capacitor between BIAS2 and VDDS 11: Capacitor between BIAS2 and VSS & Capacitor between BIAS2 and VDDS
2-1	DDR_VREF_TAP	R/W	0h	select for int ref for DDR 00 : Pad/Bias2 connected to internal reference VDDS/2 for 2uA current load 01 : Pad/Bias2 connected to internal reference VDDS/2 for 4uA current load 10 : Pad/Bias2 connected to internal reference VDDS/2 for 6uA current load 11 : Pad/Bias2 connected to internal reference VDDS/2 for 8uA current load
0	DDR_VREF_EN	R/W	0h	active high internal reference enable for DDR

7.3.1.226 CTRL_TPCC_EVT_MUX_0_3 Register (Offset = F90h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_TPCC_EVT_MUX_0_3 is shown in Figure 7-230 and described in Table 7-237.

Return to Summary Table.

Figure 7-230. CTRL_TPCC_EVT_MUX_0_3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		EVT_MUX_3						RESERVED		EVT_MUX_2					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED		EVT_MUX_1						RESERVED		EVT_MUX_0					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					

Table 7-237. CTRL_TPCC_EVT_MUX_0_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	EVT_MUX_3	R/W	0h	Selects 1 of 64 inputs for DMA event 3
23-22	RESERVED	R/W	0h	
21-16	EVT_MUX_2	R/W	0h	Selects 1 of 64 inputs for DMA event 2
15-14	RESERVED	R/W	0h	
13-8	EVT_MUX_1	R/W	0h	Selects 1 of 64 inputs for DMA event 1
7-6	RESERVED	R/W	0h	
5-0	EVT_MUX_0	R/W	0h	Selects 1 of 64 inputs for DMA event 0

7.3.1.227 CTRL_TPCC_EVT_MUX_4_7 Register (Offset = F94h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_TPCC_EVT_MUX_4_7 is shown in Figure 7-231 and described in Table 7-238.

Return to Summary Table.

Figure 7-231. CTRL_TPCC_EVT_MUX_4_7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		EVT_MUX_7						RESERVED		EVT_MUX_6					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		EVT_MUX_5						RESERVED		EVT_MUX_4					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					

Table 7-238. CTRL_TPCC_EVT_MUX_4_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	EVT_MUX_7	R/W	0h	Selects 1 of 64 inputs for DMA event 7
23-22	RESERVED	R/W	0h	
21-16	EVT_MUX_6	R/W	0h	Selects 1 of 64 inputs for DMA event 6
15-14	RESERVED	R/W	0h	
13-8	EVT_MUX_5	R/W	0h	Selects 1 of 64 inputs for DMA event 5
7-6	RESERVED	R/W	0h	
5-0	EVT_MUX_4	R/W	0h	Selects 1 of 64 inputs for DMA event 4

7.3.1.228 CTRL_TPCC_EVT_MUX_8_11 Register (Offset = F98h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_TPCC_EVT_MUX_8_11 is shown in Figure 7-232 and described in Table 7-239.

Return to Summary Table.

Figure 7-232. CTRL_TPCC_EVT_MUX_8_11 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		EVT_MUX_11						RESERVED		EVT_MUX_10					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		EVT_MUX_9						RESERVED		EVT_MUX_8					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					

Table 7-239. CTRL_TPCC_EVT_MUX_8_11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	EVT_MUX_11	R/W	0h	Selects 1 of 64 inputs for DMA event 11
23-22	RESERVED	R/W	0h	
21-16	EVT_MUX_10	R/W	0h	Selects 1 of 64 inputs for DMA event 10
15-14	RESERVED	R/W	0h	
13-8	EVT_MUX_9	R/W	0h	Selects 1 of 64 inputs for DMA event 9
7-6	RESERVED	R/W	0h	
5-0	EVT_MUX_8	R/W	0h	Selects 1 of 64 inputs for DMA event 8

7.3.1.229 CTRL_TPCC_EVT_MUX_12_15 Register (Offset = F9Ch) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_TPCC_EVT_MUX_12_15 is shown in Figure 7-233 and described in Table 7-240.

Return to Summary Table.

Figure 7-233. CTRL_TPCC_EVT_MUX_12_15 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_15</td><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_14</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_13</td><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_12</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td></tr>
</table>

Table 7-240. CTRL_TPCC_EVT_MUX_12_15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	EVT_MUX_15	R/W	0h	Selects 1 of 64 inputs for DMA event 15
23-22	RESERVED	R/W	0h	
21-16	EVT_MUX_14	R/W	0h	Selects 1 of 64 inputs for DMA event 14
15-14	RESERVED	R/W	0h	
13-8	EVT_MUX_13	R/W	0h	Selects 1 of 64 inputs for DMA event 13
7-6	RESERVED	R/W	0h	
5-0	EVT_MUX_12	R/W	0h	Selects 1 of 64 inputs for DMA event 12

7.3.1.230 CTRL_TPCC_EVT_MUX_16_19 Register (Offset = FA0h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_TPCC_EVT_MUX_16_19 is shown in Figure 7-234 and described in Table 7-241.

Return to Summary Table.

Figure 7-234. CTRL_TPCC_EVT_MUX_16_19 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_19</td><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_18</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_17</td><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_16</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td></tr>
</table>

Table 7-241. CTRL_TPCC_EVT_MUX_16_19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	EVT_MUX_19	R/W	0h	Selects 1 of 64 inputs for DMA event 19
23-22	RESERVED	R/W	0h	
21-16	EVT_MUX_18	R/W	0h	Selects 1 of 64 inputs for DMA event 18
15-14	RESERVED	R/W	0h	
13-8	EVT_MUX_17	R/W	0h	Selects 1 of 64 inputs for DMA event 17
7-6	RESERVED	R/W	0h	
5-0	EVT_MUX_16	R/W	0h	Selects 1 of 64 inputs for DMA event 16

7.3.1.231 CTRL_TPCC_EVT_MUX_20_23 Register (Offset = FA4h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_TPCC_EVT_MUX_20_23 is shown in Figure 7-235 and described in Table 7-242.

Return to Summary Table.

Figure 7-235. CTRL_TPCC_EVT_MUX_20_23 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		EVT_MUX_23						RESERVED		EVT_MUX_22					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED		EVT_MUX_21						RESERVED		EVT_MUX_20					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					

Table 7-242. CTRL_TPCC_EVT_MUX_20_23 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	EVT_MUX_23	R/W	0h	Selects 1 of 64 inputs for DMA event 23
23-22	RESERVED	R/W	0h	
21-16	EVT_MUX_22	R/W	0h	Selects 1 of 64 inputs for DMA event 22
15-14	RESERVED	R/W	0h	
13-8	EVT_MUX_21	R/W	0h	Selects 1 of 64 inputs for DMA event 21
7-6	RESERVED	R/W	0h	
5-0	EVT_MUX_20	R/W	0h	Selects 1 of 64 inputs for DMA event 20

7.3.1.232 CTRL_TPCC_EVT_MUX_24_27 Register (Offset = FA8h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_TPCC_EVT_MUX_24_27 is shown in Figure 7-236 and described in Table 7-243.

Return to Summary Table.

Figure 7-236. CTRL_TPCC_EVT_MUX_24_27 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_27</td><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_26</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_25</td><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_24</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td></tr>
</table>

Table 7-243. CTRL_TPCC_EVT_MUX_24_27 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	EVT_MUX_27	R/W	0h	Selects 1 of 64 inputs for DMA event 27
23-22	RESERVED	R/W	0h	
21-16	EVT_MUX_26	R/W	0h	Selects 1 of 64 inputs for DMA event 26
15-14	RESERVED	R/W	0h	
13-8	EVT_MUX_25	R/W	0h	Selects 1 of 64 inputs for DMA event 25
7-6	RESERVED	R/W	0h	
5-0	EVT_MUX_24	R/W	0h	Selects 1 of 64 inputs for DMA event 24

7.3.1.233 CTRL_TPCC_EVT_MUX_28_31 Register (Offset = FACh) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_TPCC_EVT_MUX_28_31 is shown in Figure 7-237 and described in Table 7-244.

Return to Summary Table.

Figure 7-237. CTRL_TPCC_EVT_MUX_28_31 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		EVT_MUX_31						RESERVED		EVT_MUX_30					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		EVT_MUX_29						RESERVED		EVT_MUX_28					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					

Table 7-244. CTRL_TPCC_EVT_MUX_28_31 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	EVT_MUX_31	R/W	0h	Selects 1 of 64 inputs for DMA event 31
23-22	RESERVED	R/W	0h	
21-16	EVT_MUX_30	R/W	0h	Selects 1 of 64 inputs for DMA event 30
15-14	RESERVED	R/W	0h	
13-8	EVT_MUX_29	R/W	0h	Selects 1 of 64 inputs for DMA event 29
7-6	RESERVED	R/W	0h	
5-0	EVT_MUX_28	R/W	0h	Selects 1 of 64 inputs for DMA event 28

7.3.1.234 CTRL_TPCC_EVT_MUX_32_35 Register (Offset = FB0h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_TPCC_EVT_MUX_32_35 is shown in Figure 7-238 and described in Table 7-245.

Return to Summary Table.

Figure 7-238. CTRL_TPCC_EVT_MUX_32_35 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		EVT_MUX_35						RESERVED		EVT_MUX_34					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		EVT_MUX_33						RESERVED		EVT_MUX_32					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					

Table 7-245. CTRL_TPCC_EVT_MUX_32_35 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	EVT_MUX_35	R/W	0h	Selects 1 of 64 inputs for DMA event 35
23-22	RESERVED	R/W	0h	
21-16	EVT_MUX_34	R/W	0h	Selects 1 of 64 inputs for DMA event 34
15-14	RESERVED	R/W	0h	
13-8	EVT_MUX_33	R/W	0h	Selects 1 of 64 inputs for DMA event 33
7-6	RESERVED	R/W	0h	
5-0	EVT_MUX_32	R/W	0h	Selects 1 of 64 inputs for DMA event 32

7.3.1.235 CTRL_TPCC_EVT_MUX_36_39 Register (Offset = FB4h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_TPCC_EVT_MUX_36_39 is shown in Figure 7-239 and described in Table 7-246.

Return to Summary Table.

Figure 7-239. CTRL_TPCC_EVT_MUX_36_39 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_39</td><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_38</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_37</td><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_36</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td></tr>
</table>

Table 7-246. CTRL_TPCC_EVT_MUX_36_39 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	EVT_MUX_39	R/W	0h	Selects 1 of 64 inputs for DMA event 39
23-22	RESERVED	R/W	0h	
21-16	EVT_MUX_38	R/W	0h	Selects 1 of 64 inputs for DMA event 38
15-14	RESERVED	R/W	0h	
13-8	EVT_MUX_37	R/W	0h	Selects 1 of 64 inputs for DMA event 37
7-6	RESERVED	R/W	0h	
5-0	EVT_MUX_36	R/W	0h	Selects 1 of 64 inputs for DMA event 36

7.3.1.236 CTRL_TPCC_EVT_MUX_40_43 Register (Offset = FB8h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_TPCC_EVT_MUX_40_43 is shown in Figure 7-240 and described in Table 7-247.

Return to Summary Table.

Figure 7-240. CTRL_TPCC_EVT_MUX_40_43 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_43</td><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_42</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_41</td><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_40</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td></tr>
</table>

Table 7-247. CTRL_TPCC_EVT_MUX_40_43 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	EVT_MUX_43	R/W	0h	Selects 1 of 64 inputs for DMA event 43
23-22	RESERVED	R/W	0h	
21-16	EVT_MUX_42	R/W	0h	Selects 1 of 64 inputs for DMA event 42
15-14	RESERVED	R/W	0h	
13-8	EVT_MUX_41	R/W	0h	Selects 1 of 64 inputs for DMA event 41
7-6	RESERVED	R/W	0h	
5-0	EVT_MUX_40	R/W	0h	Selects 1 of 64 inputs for DMA event 40

7.3.1.237 CTRL_TPCC_EVT_MUX_44_47 Register (Offset = FBCh) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_TPCC_EVT_MUX_44_47 is shown in Figure 7-241 and described in Table 7-248.

Return to Summary Table.

Figure 7-241. CTRL_TPCC_EVT_MUX_44_47 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		EVT_MUX_47						RESERVED		EVT_MUX_46					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED		EVT_MUX_45						RESERVED		EVT_MUX_44					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					

Table 7-248. CTRL_TPCC_EVT_MUX_44_47 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	EVT_MUX_47	R/W	0h	Selects 1 of 64 inputs for DMA event 47
23-22	RESERVED	R/W	0h	
21-16	EVT_MUX_46	R/W	0h	Selects 1 of 64 inputs for DMA event 46
15-14	RESERVED	R/W	0h	
13-8	EVT_MUX_45	R/W	0h	Selects 1 of 64 inputs for DMA event 45
7-6	RESERVED	R/W	0h	
5-0	EVT_MUX_44	R/W	0h	Selects 1 of 64 inputs for DMA event 44

7.3.1.238 CTRL_TPCC_EVT_MUX_48_51 Register (Offset = FC0h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_TPCC_EVT_MUX_48_51 is shown in Figure 7-242 and described in Table 7-249.

Return to Summary Table.

Figure 7-242. CTRL_TPCC_EVT_MUX_48_51 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		EVT_MUX_51						RESERVED		EVT_MUX_50					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		EVT_MUX_49						RESERVED		EVT_MUX_48					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					

Table 7-249. CTRL_TPCC_EVT_MUX_48_51 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	EVT_MUX_51	R/W	0h	Selects 1 of 64 inputs for DMA event 51
23-22	RESERVED	R/W	0h	
21-16	EVT_MUX_50	R/W	0h	Selects 1 of 64 inputs for DMA event 50
15-14	RESERVED	R/W	0h	
13-8	EVT_MUX_49	R/W	0h	Selects 1 of 64 inputs for DMA event 49
7-6	RESERVED	R/W	0h	
5-0	EVT_MUX_48	R/W	0h	Selects 1 of 64 inputs for DMA event 48

7.3.1.239 CTRL_TPCC_EVT_MUX_52_55 Register (Offset = FC4h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_TPCC_EVT_MUX_52_55 is shown in Figure 7-243 and described in Table 7-250.

Return to Summary Table.

Figure 7-243. CTRL_TPCC_EVT_MUX_52_55 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		EVT_MUX_55						RESERVED		EVT_MUX_54					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED		EVT_MUX_53						RESERVED		EVT_MUX_52					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					

Table 7-250. CTRL_TPCC_EVT_MUX_52_55 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	EVT_MUX_55	R/W	0h	Selects 1 of 64 inputs for DMA event 55
23-22	RESERVED	R/W	0h	
21-16	EVT_MUX_54	R/W	0h	Selects 1 of 64 inputs for DMA event 54
15-14	RESERVED	R/W	0h	
13-8	EVT_MUX_53	R/W	0h	Selects 1 of 64 inputs for DMA event 53
7-6	RESERVED	R/W	0h	
5-0	EVT_MUX_52	R/W	0h	Selects 1 of 64 inputs for DMA event 52

7.3.1.240 CTRL_TPCC_EVT_MUX_56_59 Register (Offset = FC8h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_TPCC_EVT_MUX_56_59 is shown in Figure 7-244 and described in Table 7-251.

Return to Summary Table.

Figure 7-244. CTRL_TPCC_EVT_MUX_56_59 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_59</td><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_58</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_57</td><td colspan="2">RESERVED</td><td colspan="6">EVT_MUX_56</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td></tr>
</table>

Table 7-251. CTRL_TPCC_EVT_MUX_56_59 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	EVT_MUX_59	R/W	0h	Selects 1 of 64 inputs for DMA event 59
23-22	RESERVED	R/W	0h	
21-16	EVT_MUX_58	R/W	0h	Selects 1 of 64 inputs for DMA event 58
15-14	RESERVED	R/W	0h	
13-8	EVT_MUX_57	R/W	0h	Selects 1 of 64 inputs for DMA event 57
7-6	RESERVED	R/W	0h	
5-0	EVT_MUX_56	R/W	0h	Selects 1 of 64 inputs for DMA event 56

7.3.1.241 CTRL_TPCC_EVT_MUX_60_63 Register (Offset = FCCh) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_TPCC_EVT_MUX_60_63 is shown in Figure 7-245 and described in Table 7-252.

Return to Summary Table.

Figure 7-245. CTRL_TPCC_EVT_MUX_60_63 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		EVT_MUX_60						RESERVED		EVT_MUX_61					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED		EVT_MUX_62						RESERVED		EVT_MUX_63					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					

Table 7-252. CTRL_TPCC_EVT_MUX_60_63 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	EVT_MUX_60	R/W	0h	Selects 1 of 64 inputs for DMA event 60
23-22	RESERVED	R/W	0h	
21-16	EVT_MUX_61	R/W	0h	Selects 1 of 64 inputs for DMA event 61
15-14	RESERVED	R/W	0h	
13-8	EVT_MUX_62	R/W	0h	Selects 1 of 64 inputs for DMA event 62
7-6	RESERVED	R/W	0h	
5-0	EVT_MUX_63	R/W	0h	Selects 1 of 64 inputs for DMA event 63

7.3.1.242 CTRL_TIMER_EVT_CAPT Register (Offset = FD0h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_TIMER_EVT_CAPT is shown in Figure 7-246 and described in Table 7-253.

Return to Summary Table.

Figure 7-246. CTRL_TIMER_EVT_CAPT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="11">RESERVED</td><td colspan="5">TIMER7_EVTCAPT</td></tr>
<tr><td colspan="11">R/W-0h</td><td colspan="5">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">RESERVED</td><td colspan="5">TIMER6_EVTCAPT</td><td colspan="3">RESERVED</td><td colspan="5">TIMER5_EVTCAPT</td></tr>
<tr><td colspan="3">R/W-0h</td><td colspan="5">R/W-0h</td><td colspan="3">R/W-0h</td><td colspan="5">R/W-0h</td></tr>
</table>

Table 7-253. CTRL_TIMER_EVT_CAPT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R/W	0h	
20-16	TIMER7_EVTCAPT	R/W	0h	Timer 7 event capture mux
15-13	RESERVED	R/W	0h	
12-8	TIMER6_EVTCAPT	R/W	0h	Timer 6 event capture mux
7-5	RESERVED	R/W	0h	
4-0	TIMER5_EVTCAPT	R/W	0h	Timer 5 event capture mux

7.3.1.243 CTRL_ECAP_EVT_CAPT Register (Offset = FD4h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_ECAP_EVT_CAPT is shown in Figure 7-247 and described in Table 7-254.

Return to Summary Table.

Figure 7-247. CTRL_ECAP_EVT_CAPT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED											ECAP2_EVTCAPT				
R/W-0h											R/W-0h				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			ECAP1_EVTCAPT					RESERVED			ECAP0_EVTCAPT				
R/W-0h			R/W-0h					R/W-0h			R/W-0h				

Table 7-254. CTRL_ECAP_EVT_CAPT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R/W	0h	
20-16	ECAP2_EVTCAPT	R/W	0h	ECAP2 event capture mux
15-13	RESERVED	R/W	0h	
12-8	ECAP1_EVTCAPT	R/W	0h	ECAP1 event capture mux
7-5	RESERVED	R/W	0h	
4-0	ECAP0_EVTCAPT	R/W	0h	ECAP0 event capture mux

7.3.1.244 CTRL_ADC0_EVT_CAPT Register (Offset = FD8h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_ADC0_EVT_CAPT is shown in Figure 7-248 and described in Table 7-255.

Return to Summary Table.

Figure 7-248. CTRL_ADC0_EVT_CAPT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="12">RESERVED</td><td colspan="4">ADC0_EVTCAPT</td></tr>
<tr><td colspan="12">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 7-255. CTRL_ADC0_EVT_CAPT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	ADC0_EVTCAPT	R/W	0h	ADC0 event capture mux

7.3.1.245 CTRL_ADC1_EVT_CAPT Register (Offset = FDCh) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_ADC1_EVT_CAPT is shown in Figure 7-249 and described in Table 7-256.

Return to Summary Table.

Figure 7-249. CTRL_ADC1_EVT_CAPT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="12">RESERVED</td><td colspan="4">ADC1_EVTCAPT</td></tr>
<tr><td colspan="12">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 7-256. CTRL_ADC1_EVT_CAPT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	ADC1_EVTCAPT	R/W	0h	ADC1 event capture mux

7.3.1.246 CTRL_RESET_ISO Register (Offset = 1000h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_RESET_ISO is shown in Figure 7-250 and described in Table 7-257.

Return to Summary Table.

Figure 7-250. CTRL_RESET_ISO Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED						JTAG_ISO_CTRL	CPSW_ISO_CTRL
R/W-0h						R/W-0h	R/W-0h

Table 7-257. CTRL_RESET_ISO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	JTAG_ISO_CTRL	R/W	0h	0 : JTAG IOs are not isolated from warm reset 1 : JTAG IOs are isolated from warm reset
0	CPSW_ISO_CTRL	R/W	0h	0 : Ethernet Switch is not isolated from warm reset 1 : Ethernet Switch is isolated from warm reset Please refer to Ethernet Switch reset isolation microarch for details on which registers are isolated in the SOC.

7.3.1.247 CTRL_DPLL_PWR_SW Register (Offset = 1318h) [reset = 03030303h]

Register mask: FFFFFFFFh

CTRL_DPLL_PWR_SW is shown in Figure 7-251 and described in Table 7-258.

Return to Summary Table.

Figure 7-251. CTRL_DPLL_PWR_SW Register

31	30	29	28	27	26	25	24
SW_CTRL_DDR_PLL	RESERVED	ISOSCAN_DDR	RET_DDR	RESET_DDR	ISO_DDR	PGOODIN_DDR	PONIN_DDR
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
SW_CTRL_DISP_PLL	RESERVED	ISOSCAN_DISP	RET_DISP	RESET_DISP	ISO_DISP	PGOODIN_DISP	PONIN_DISP
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
SW_CTRL_PER_PLL	RESERVED	ISOSCAN_PER	RET_PER	RESET_PER	ISO_PER	PGOODIN_PER	PONIN_PER
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
SW_CTRL_MPU_PLL	RESERVED	ISOSCAN_MPU	RET_MPU	RESET_MPU	ISO_MPU	PGOODIN_MPU	PONIN_MPU
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h

Table 7-258. CTRL_DPLL_PWR_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SW_CTRL_DDR_PLL	R/W	0h	Enable software control over DDR DPLL RET RESET ISO PGOODIN PONIN for power savings. 0: PRCM controls the DPLL reset RET = 0 ISO = 0 PGOODIN = 1 PONIN = 1. 1: Controlled by corresponding bits in this register.
30	RESERVED	R/W	0h	
29	ISOSCAN_DDR	R/W	0h	Drives ISOSCAN of DDR PLL
28	RET_DDR	R/W	0h	Drives RET signal of DDR PLL
27	RESET_DDR	R/W	0h	Drives RESET of DDR DPLL
26	ISO_DDR	R/W	0h	Drives ISO of DDR DPLL
25	PGOODIN_DDR	R/W	1h	Drives PGOODIN of DDR DPLL
24	PONIN_DDR	R/W	1h	Drives PONIN of DDR DPLL
23	SW_CTRL_DISP_PLL	R/W	0h	Enable software control over DISP DPLL RET RESET ISO PGOODIN PONIN for power savings. 0: PRCM controls the DPLL reset RET = 0 ISO = 0 PGOODIN = 1 PONIN = 1. 1: Controlled by corresponding bits in this register.
22	RESERVED	R/W	0h	
21	ISOSCAN_DISP	R/W	0h	Drives ISOSCAN of DISP PLL
20	RET_DISP	R/W	0h	Drives RET of DISP DPLL
19	RESET_DISP	R/W	0h	Drives RESET of DISP DPLL
18	ISO_DISP	R/W	0h	Drives ISO of DISP DPLL
17	PGOODIN_DISP	R/W	1h	Drives PGOODIN of DISP DPLL
16	PONIN_DISP	R/W	1h	Drives PONIN of DISP DPLL

Table 7-258. CTRL_DPLL_PWR_SW Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	SW_CTRL_PER_PLL	R/W	0h	Enable software control over PER DPLL RET RESET ISO PGOODIN PONIN for power savings. 0: PRCM controls the DPLL reset RET = 0 ISO = 0 PGOODIN = 1 PONIN = 1. 1: Controlled by corresponding bits in this register.
14	RESERVED	R/W	0h	
13	ISOSCAN_PER	R/W	0h	Drives ISOSCAN of PER PLL
12	RET_PER	R/W	0h	Drives RET of PER DPLL
11	RESET_PER	R/W	0h	Drives RESET signal of PER DPLL
10	ISO_PER	R/W	0h	Drives ISO signal of PER DPLL
9	PGOODIN_PER	R/W	1h	Drives PGOODIN signal of PER DPLL
8	PONIN_PER	R/W	1h	Drives PONIN signal of PER DPLL
7	SW_CTRL_MPU_PLL	R/W	0h	Enable S/W control over MPU DPLL RET RESET ISO PGOODIN PONIN for power savings. 0: PRCM controls the DPLL reset RET = 0 ISO = 0 PGOODIN = 1 PONIN = 1. 1: Controlled by corresponding bits in this register.
6	RESERVED	R/W	0h	
5	ISOSCAN_MPU	R/W	0h	Drives ISOSCAN of MPU PLL
4	RET_MPU	R/W	0h	Drives RET of MPU DPLL
3	RESET_MPU	R/W	0h	Drives RESET signal of MPU DPLL
2	ISO_MPU	R/W	0h	Drives ISO signal of MPU DPLL
1	PGOODIN_MPU	R/W	1h	Drives PGOODIN signal of MPU DPLL
0	PONIN_MPU	R/W	1h	Drives PONIN signal of MPU DPLL

7.3.1.248 CTRL_DDR_CKE Register (Offset = 131Ch) [reset = X]

CTRL_DDR_CKE is shown in Figure 7-252 and described in Table 7-259.

Return to Summary Table.

Figure 7-252. CTRL_DDR_CKE Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							

23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				DDR_CKE1_ST	DDR_CKE0_ST	DDR_CKE1_CTRL	DDR_CKE0_CTRL
R/W-0h				R-X	R-X	R/W-0h	R/W-0h

Table 7-259. CTRL_DDR_CKE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	DDR_CKE1_ST	R	X	CKE status bit
2	DDR_CKE0_ST	R	X	CKE status bit
1	DDR_CKE1_CTRL	R/W	0h	CKE from EMIF/DDRPHY is ANDed with this bit. 0: CKE to memories gated off to zero. External DRAM memories will not be able to register DDR commands from the controller. 1: Normal operation. CKE is now controlled by EMIF/DDR PHY.
0	DDR_CKE0_CTRL	R/W	0h	CKE from EMIF/DDRPHY is ANDed with this bit. 0: CKE to memories gated off to zero. External DRAM memories will not be able to register DDR commands from this device. 1: Normal operation. CKE is now controlled by EMIF/DDR PHY.

7.3.1.249 CTRL_VSLDO Register (Offset = 1320h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_VSLDO is shown in Figure 7-253 and described in Table 7-260.

Return to Summary Table.

Figure 7-253. CTRL_VSLDO Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">CTRL_VSLDO</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">CTRL_VSLDO</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">CTRL_VSLDO</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">CTRL_VSLDO</td><td>VSLDO_CORE_AUTO_RAMP_EN</td><td>RESERVED</td></tr>
<tr><td colspan="6">R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 7-260. CTRL_VSLDO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	CTRL_VSLDO	R/W	0h	Selectively Modifiable Attribute Bits (Spare Register)
1	VSLDO_CORE_AUTO_RAMP_EN	R/W	0h	0 : PRCM controls VSLDO 1 : Allows H/W to bring VSLDO out of retention on wakeup from deep-sleep
0	RESERVED	R/W	0h	

7.3.1.250 CTRL_WAKEPROC_TXEV_EOI Register (Offset = 1324h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_WAKEPROC_TXEV_EOI is shown in Figure 7-254 and described in Table 7-261.

Return to Summary Table.

Figure 7-254. CTRL_WAKEPROC_TXEV_EOI Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							CTRL_WAKEPROC_TXEV_EOI
R/W-0h							R/W-0h

Table 7-261. CTRL_WAKEPROC_TXEV_EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	CTRL_WAKEPROC_TXEV_EOI	R/W	0h	TXEV (Event) from M3 processor is a pulse signal connected as interrupt to MPUSS IRQ(110) Since MPUSS expects level signals. The TXEV pulse from WAKEM3 is converted to a level in glue logic. The logic works as follows On a 0-1 transition on TXEV, the IRQ[110] is set For clearing the interrupt, S/W must do the following S/W must clear the IRQ[110] by writing a ‘1’ to CTRL_WAKEPROC_TXEV_EOI bit in this register This bit is sticky and for re-arming the IRQ[110], S/W must write a ‘0’ to this field in the ISR

7.3.1.251 CTRL_IPC_MSG_REG0 Register (Offset = 1328h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_IPC_MSG_REG0 is shown in Figure 7-255 and described in Table 7-262.

Return to Summary Table.

Figure 7-255. CTRL_IPC_MSG_REG0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">IPC_MSG_REG0</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 7-262. CTRL_IPC_MSG_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IPC_MSG_REG0	R/W	0h	Inter Processor Messaging Register

7.3.1.252 CTRL_IPC_MSG_REG1 Register (Offset = 132Ch) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_IPC_MSG_REG1 is shown in Figure 7-256 and described in Table 7-263.

Return to Summary Table.

Figure 7-256. CTRL_IPC_MSG_REG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPC_MSG_REG1																															
R/W-0h																															

Table 7-263. CTRL_IPC_MSG_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IPC_MSG_REG1	R/W	0h	Inter Processor Messaging Register

7.3.1.253 CTRL_IPC_MSG_REG2 Register (Offset = 1330h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_IPC_MSG_REG2 is shown in Figure 7-257 and described in Table 7-264.

Return to Summary Table.

Figure 7-257. CTRL_IPC_MSG_REG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPC_MSG_REG2																															
R/W-0h																															

Table 7-264. CTRL_IPC_MSG_REG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IPC_MSG_REG2	R/W	0h	Inter Processor Messaging Register

7.3.1.254 CTRL_IPC_MSG_REG3 Register (Offset = 1334h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_IPC_MSG_REG3 is shown in Figure 7-258 and described in Table 7-265.

Return to Summary Table.

Figure 7-258. CTRL_IPC_MSG_REG3 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">IPC_MSG_REG3</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 7-265. CTRL_IPC_MSG_REG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IPC_MSG_REG3	R/W	0h	Inter Processor Messaging Register

7.3.1.255 CTRL_IPC_MSG_REG4 Register (Offset = 1338h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_IPC_MSG_REG4 is shown in Figure 7-259 and described in Table 7-266.

Return to Summary Table.

Figure 7-259. CTRL_IPC_MSG_REG4 Register

31-0
IPC_MSG_REG4
R/W-0h

Table 7-266. CTRL_IPC_MSG_REG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IPC_MSG_REG4	R/W	0h	Inter Processor Messaging Register

7.3.1.256 CTRL_IPC_MSG_REG5 Register (Offset = 133Ch) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_IPC_MSG_REG5 is shown in Figure 7-260 and described in Table 7-267.

Return to Summary Table.

Figure 7-260. CTRL_IPC_MSG_REG5 Register

31-0
IPC_MSG_REG5
R/W-0h

Table 7-267. CTRL_IPC_MSG_REG5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IPC_MSG_REG5	R/W	0h	Inter Processor Messaging Register

7.3.1.257 CTRL_IPC_MSG_REG6 Register (Offset = 1340h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_IPC_MSG_REG6 is shown in Figure 7-261 and described in Table 7-268.

Return to Summary Table.

Figure 7-261. CTRL_IPC_MSG_REG6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPC_MSG_REG6																															
R/W-0h																															

Table 7-268. CTRL_IPC_MSG_REG6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IPC_MSG_REG6	R/W	0h	Inter Processor Messaging Register

7.3.1.258 CTRL_IPC_MSG_REG7 Register (Offset = 1344h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_IPC_MSG_REG7 is shown in Figure 7-262 and described in Table 7-269.

Return to Summary Table.

Figure 7-262. CTRL_IPC_MSG_REG7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPC_MSG_REG7																															
R/W-0h																															

Table 7-269. CTRL_IPC_MSG_REG7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IPC_MSG_REG7	R/W	0h	Inter Processor Messaging Register

7.3.1.259 CTRL_IPC_MSG_REG8 Register (Offset = 1348h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_IPC_MSG_REG8 is shown in Figure 7-263 and described in Table 7-270.

Return to Summary Table.

Figure 7-263. CTRL_IPC_MSG_REG8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPC_MSG_REG8																															
R/W-0h																															

Table 7-270. CTRL_IPC_MSG_REG8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IPC_MSG_REG8	R/W	0h	Inter Processor Messaging Register

7.3.1.260 CTRL_IPC_MSG_REG9 Register (Offset = 134Ch) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_IPC_MSG_REG9 is shown in Figure 7-264 and described in Table 7-271.

Return to Summary Table.

Figure 7-264. CTRL_IPC_MSG_REG9 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">IPC_MSG_REG9</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 7-271. CTRL_IPC_MSG_REG9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IPC_MSG_REG9	R/W	0h	Inter Processor Messaging Register

7.3.1.261 CTRL_IPC_MSG_REG10 Register (Offset = 1350h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_IPC_MSG_REG10 is shown in Figure 7-265 and described in Table 7-272.

Return to Summary Table.

Figure 7-265. CTRL_IPC_MSG_REG10 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPC_MSG_REG10																															
R/W-0h																															

Table 7-272. CTRL_IPC_MSG_REG10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IPC_MSG_REG10	R/W	0h	Inter Processor Messaging Register

7.3.1.262 CTRL_IPC_MSG_REG11 Register (Offset = 1354h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_IPC_MSG_REG11 is shown in Figure 7-266 and described in Table 7-273.

Return to Summary Table.

Figure 7-266. CTRL_IPC_MSG_REG11 Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
IPC_MSG_REG11
R/W-0h

Table 7-273. CTRL_IPC_MSG_REG11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IPC_MSG_REG11	R/W	0h	Inter Processor Messaging Register

7.3.1.263 CTRL_IPC_MSG_REG12 Register (Offset = 1358h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_IPC_MSG_REG12 is shown in Figure 7-267 and described in Table 7-274.

Return to Summary Table.

Figure 7-267. CTRL_IPC_MSG_REG12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPC_MSG_REG12																															
R/W-0h																															

Table 7-274. CTRL_IPC_MSG_REG12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IPC_MSG_REG12	R/W	0h	Inter Processor Messaging Register

7.3.1.264 CTRL_IPC_MSG_REG13 Register (Offset = 135Ch) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_IPC_MSG_REG13 is shown in Figure 7-268 and described in Table 7-275.

Return to Summary Table.

Figure 7-268. CTRL_IPC_MSG_REG13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPC_MSG_REG13																															
R/W-0h																															

Table 7-275. CTRL_IPC_MSG_REG13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IPC_MSG_REG13	R/W	0h	Inter Processor Messaging Register

7.3.1.265 CTRL_IPC_MSG_REG14 Register (Offset = 1360h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_IPC_MSG_REG14 is shown in Figure 7-269 and described in Table 7-276.

Return to Summary Table.

Figure 7-269. CTRL_IPC_MSG_REG14 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPC_MSG_REG14																															
R/W-0h																															

Table 7-276. CTRL_IPC_MSG_REG14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IPC_MSG_REG14	R/W	0h	Inter Processor Messaging Register

7.3.1.266 CTRL_IPC_INTR Register (Offset = 1364h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_IPC_INTR is shown in Figure 7-270 and described in Table 7-277.

Return to Summary Table.

Figure 7-270. CTRL_IPC_INTR Register

31	30	29	28	27	26	25	24
INTR2WAKEPROC	IPC_MSG_REG15						
R/W-0h	R/W-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
IPC_MSG_REG15							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
IPC_MSG_REG15							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IPC_MSG_REG15							
R/W-0h							

Table 7-277. CTRL_IPC_INTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INTR2WAKEPROC	R/W	0h	Interrupt to M3
30-0	IPC_MSG_REG15	R/W	0h	Inter Processor Messaging Register

7.3.1.267 CTRL_DPLL_PWR_SW_CTRL2 Register (Offset = 138Ch) [reset = 3h]

Register mask: FFFFFFFFh

CTRL_DPLL_PWR_SW_CTRL2 is shown in Figure 7-271 and described in Table 7-278.

Return to Summary Table.

Figure 7-271. CTRL_DPLL_PWR_SW_CTRL2 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
SW_CTRL_EXTCLK_PLL	RESERVED	ISOSCAN_EXTCLK	RET_EXTCLK	RESET_EXTCLK	ISO_EXTCLK	PGOODIN_EXTCLK	PONIN_EXTCLK
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h

Table 7-278. CTRL_DPLL_PWR_SW_CTRL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7	SW_CTRL_EXTCLK_PLL	R/W	0h	Enable S/W control over EXTCLK DPLL RET RESET ISO PGOODIN PONIN for power savings. 0: PRCM controls the DPLL reset RET = 0 ISO = 0 GOODIN = 1 PONIN = 1. 1: Controlled by corresponding bits in this register.
6	RESERVED	R/W	0h	
5	ISOSCAN_EXTCLK	R/W	0h	Drives ISOSCAN of EXTCLK PLL
4	RET_EXTCLK	R/W	0h	Drives RET of EXTCLK DPLL
3	RESET_EXTCLK	R/W	0h	Drives RESET signal of EXTCLK DPLL
2	ISO_EXTCLK	R/W	0h	Drives ISO signal of EXTCLK DPLL
1	PGOODIN_EXTCLK	R/W	1h	Drives PGOODIN signal of EXTCLK DPLL
0	PONIN_EXTCLK	R/W	1h	Drives PONIN signal of EXTCLK DPLL

7.3.1.268 CTRL_DPLL_PWR_SW_STS2 Register (Offset = 1390h) [reset = X]

CTRL_DPLL_PWR_SW_STS2 is shown in Figure 7-272 and described in Table 7-279.

Return to Summary Table.

Figure 7-272. CTRL_DPLL_PWR_SW_STS2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						PGOODOUT_EXTCLK	PONOUT_EXTCLK
R-0h						R-X	R-X

Table 7-279. CTRL_DPLL_PWR_SW_STS2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	PGOODOUT_EXTCLK	R	X	PGOODOUT signal from EXTCLK DPLL
0	PONOUT_EXTCLK	R	X	PONOUT signal from EXTCLK DPLL

7.3.1.269 CTRL_RESET_MISC Register (Offset = 1394h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_RESET_MISC is shown in Figure 7-273 and described in Table 7-280.

Return to Summary Table.

Figure 7-273. CTRL_RESET_MISC Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">CTRL_RESET_MISC</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">CTRL_RESET_MISC</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">CTRL_RESET_MISC</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">CTRL_RESET_MISC</td><td>RESERVED</td><td>NRESETIN_OUT_CTRL</td></tr>
<tr><td colspan="6">R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 7-280. CTRL_RESET_MISC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	CTRL_RESET_MISC	R/W	0h	Selectively Modifiable Attribute Bits (Spare Register)
1	RESERVED	R/W	0h	
0	NRESETIN_OUT_CTRL	R/W	0h	When 0 nRESETIN_OUT input is usable as an external warm reset input source. When 1 nRESETIN_OUT is not usable as warm reset input source (any input transitions on this pin or if the IO buffer is changed to disabled state will not cause a warm reset).

7.3.1.270 CTRL_DDR_ADDRCTRL_IOCTRL Register (Offset = 1404h) [reset = X]

CTRL_DDR_ADDRCTRL_IOCTRL is shown in Figure 7-274 and described in Table 7-281.

Return to Summary Table.

Figure 7-274. CTRL_DDR_ADDRCTRL_IOCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED						IO_CONFIG_SR_CLK	
R/W-0h						R/W-X	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IO_CONFIG_I_CLK			IO_CONFIG_SR		IO_CONFIG_I		
R/W-0h			R/W-X		R/W-4h		

Table 7-281. CTRL_DDR_ADDRCTRL_IOCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	0h	
9-8	IO_CONFIG_SR_CLK	R/W	X	2 bit to program clock IO Pads (CK/CK#) output Slew Rate
7-5	IO_CONFIG_I_CLK	R/W	0h	3 bit configuration input to program clock IO Pads (CK/CK#) output Impedance
4-3	IO_CONFIG_SR	R/W	X	2 bit to program addr/cmd IO Pads output Slew Rate
2-0	IO_CONFIG_I	R/W	4h	3 bit configuration input to program addr/cmd IO Pad output Impedance

7.3.1.271 CTRL_DDR_ADDRCTRL_WD0_IOCTRL Register (Offset = 1408h) [reset = 08000000h]

Register mask: FFFFFFFFh

CTRL_DDR_ADDRCTRL_WD0_IOCTRL is shown in Figure 7-275 and described in Table 7-282.

Return to Summary Table.

Figure 7-275. CTRL_DDR_ADDRCTRL_WD0_IOCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG																															
R/W-08000000h																															

Table 7-282. CTRL_DDR_ADDRCTRL_WD0_IOCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REG	R/W	08000000h	WD0 - IO control Bit mapping is as below: 31:30: Reserved 29: ddr_odt0 28: ddr_odt1 27: ddr_resetn 26: ddr_csn0 25: ddr_csn1 24: ddr_cke 23: ddr_ck 22: ddr_nck 21: ddr_casn 20: ddr_rasn 19: ddr_wen 18: ddr_ba0 17: ddr_ba1 16: ddr_ba2 (15:0) : Addr(15:0)

7.3.1.272 CTRL_DDR_ADDRCTRL_WD1_IOCTRL Register (Offset = 140Ch) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_DDR_ADDRCTRL_WD1_IOCTRL is shown in Figure 7-276 and described in Table 7-283.

Return to Summary Table.

Figure 7-276. CTRL_DDR_ADDRCTRL_WD1_IOCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG																															
R/W-0h																															

Table 7-283. CTRL_DDR_ADDRCTRL_WD1_IOCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REG	R/W	0h	WD1 - IO control Bit mapping is as below: 31:30: Reserved 29: ddr_odt0 28: ddr_odt1 27: ddr_resetn 26: ddr_csn0 25: ddr_csn1 24: ddr_cke 23: ddr_ck 22: ddr_nck 21: ddr_casn 20: ddr_rasn 19: ddr_wen 18: ddr_ba0 17: ddr_ba1 16: ddr_ba2 (15:0) : Addr(15:0)

7.3.1.273 CTRL_DDR_DATA0_IOCTRL Register (Offset = 1440h) [reset = X]

CTRL_DDR_DATA0_IOCTRL is shown in Figure 7-277 and described in Table 7-284.

Return to Summary Table.

Figure 7-277. CTRL_DDR_DATA0_IOCTRL Register

31	30	29	28	27	26	25	24
RESERVED		IO_CONFIG_WD1_DQS	IO_CONFIG_WD1_DM	IO_CONFIG_WD1_DQ			
R/W-0h		R/W-1h	R/W-1h	R/W-FFh			

23	22	21	20	19	18	17	16
IO_CONFIG_WD1_DQ				IO_CONFIG_WD0_DQS	IO_CONFIG_WD0_DM	IO_CONFIG_WD0_DQ	
R/W-FFh				R/W-0h	R/W-0h	R/W-0h	

15	14	13	12	11	10	9	8
IO_CONFIG_WD0_DQ						IO_CONFIG_SR_CLK	
R/W-0h						R/W-X	

7	6	5	4	3	2	1	0
IO_CONFIG_I_CLK			IO_CONFIG_SR		IO_CONFIG_I		
R/W-0h			R/W-X		R/W-4h		

Table 7-284. CTRL_DDR_DATA0_IOCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29	IO_CONFIG_WD1_DQS	R/W	1h	This register bit controls WD1 of DQS0 WD0 WD1 00 : Pullup/Pulldown disabled 01 : Weak pulldown enabled 10 : Weak pullup enabled 11 : Weak keeper enabled
28	IO_CONFIG_WD1_DM	R/W	1h	This register bit controls WD1 of DM0 WD0 WD1 00 : Pullup/Pulldown disabled 01 : Weak pulldown enabled 10 : Weak pullup enabled 11 : Weak keeper enabled
27-20	IO_CONFIG_WD1_DQ	R/W	FFh	This register controls WD1 of DDR data pins Bit mapping is as below: 20: D(0) WD1 control 21: D(1) WD1 control 22: D(2) WD1 control 23: D(3) WD1 control 24: D(4) WD1 control 25: D(5) WD1 control 26: D(6) WD1 control 27: D(7) WD1 control
19	IO_CONFIG_WD0_DQS	R/W	0h	This register bit controls WD0 of DQS0 WD0 WD1 00 : Pullup/Pulldown disabled 01 : Weak pulldown enabled 10 : Weak pullup enabled 11 : Weak keeper enabled
18	IO_CONFIG_WD0_DM	R/W	0h	This register bit controls WD0 of DM0 WD0 WD1 00 : Pullup/Pulldown disabled 01 : Weak pulldown enabled 10 : Weak pullup enabled 11 : Weak keeper enabled

Table 7-284. CTRL_DDR_DATA0_IOCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17-10	IO_CONFIG_WD0_DQ	R/W	0h	This register controls WD0 of DDR data pins Bit mapping is as below: 10: D(0) WD0 control 11: D(1) WD0 control 12: D(2) WD0 control 13: D(3) WD0 control 14: D(4) WD0 control 15: D(5) WD0 control 16: D(6) WD0 control 17: D(7) WD0 control
9-8	IO_CONFIG_SR_CLK	R/W	X	2 bit to program clock IO Pads (DQS/DQS#) output Slew Rate. Refer to Table 7-8 "DDR Slew Rate Control Settings" for a description of these settings.
7-5	IO_CONFIG_I_CLK	R/W	0h	3 bit configuration input to program clock IO Pads (DQS/DQS#) output Impedance. Refer to Table 7-9 "DDR Impedance Control Settings" for a description of these settings.
4-3	IO_CONFIG_SR	R/W	X	2 bit to program data IO Pads output Slew Rate for D(7 to 0). Refer to Table 7-8 "DDR Slew Rate Control Settings" for a description of these settings.
2-0	IO_CONFIG_I	R/W	4h	3 bit configuration input to program data IO Pad output Impedance for D(7 to 0). Refer to Table 7-9 "DDR Impedance Control Settings" for a description of these settings.

7.3.1.274 CTRL_DDR_DATA1_IOCTRL Register (Offset = 1444h) [reset = X]

CTRL_DDR_DATA1_IOCTRL is shown in Figure 7-278 and described in Table 7-285.

Return to Summary Table.

Figure 7-278. CTRL_DDR_DATA1_IOCTRL Register

31	30	29	28	27	26	25	24
RESERVED		IO_CONFIG_WD1_DQS	IO_CONFIG_WD1_DM	IO_CONFIG_WD1_DQ			
R/W-0h		R/W-1h	R/W-1h	R/W-FFh			

23	22	21	20	19	18	17	16
IO_CONFIG_WD1_DQ				IO_CONFIG_WD0_DQS	IO_CONFIG_WD0_DM	IO_CONFIG_WD0_DQ	
R/W-FFh				R/W-0h	R/W-0h	R/W-0h	

15	14	13	12	11	10	9	8
IO_CONFIG_WD0_DQ						IO_CONFIG_SR_CLK	
R/W-0h						R/W-X	

7	6	5	4	3	2	1	0
IO_CONFIG_I_CLK			IO_CONFIG_SR		IO_CONFIG_I		
R/W-0h			R/W-X		R/W-4h		

Table 7-285. CTRL_DDR_DATA1_IOCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29	IO_CONFIG_WD1_DQS	R/W	1h	This register bit controls WD1 of DQS1 WD0 WD1 00 : Pullup/Pulldown disabled 01 : Weak pulldown enabled 10 : Weak pullup enabled 11 : Weak keeper enabled
28	IO_CONFIG_WD1_DM	R/W	1h	This register bit controls WD1 of DM1 WD0 WD1 00 : Pullup/Pulldown disabled 01 : Weak pulldown enabled 10 : Weak pullup enabled 11 : Weak keeper enabled
27-20	IO_CONFIG_WD1_DQ	R/W	FFh	This register controls WD1 of DDR data pins Bit mapping is as below: 20: D(8) WD1 control 21: D(9) WD1 control 22: D(10) WD1 control 23: D(11) WD1 control 24: D(12) WD1 control 25: D(13) WD1 control 26: D(14) WD1 control 27: D(15) WD1 control
19	IO_CONFIG_WD0_DQS	R/W	0h	This register bit controls WD0 of DQS1 WD0 WD1 00 : Pullup/Pulldown disabled 01 : Weak pulldown enabled 10 : Weak pullup enabled 11 : Weak keeper enabled
18	IO_CONFIG_WD0_DM	R/W	0h	This register bit controls WD0 of DM1 WD0 WD1 00 : Pullup/Pulldown disabled 01 : Weak pulldown enabled 10 : Weak pullup enabled 11 : Weak keeper enabled

Table 7-285. CTRL_DDR_DATA1_IOCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17-10	IO_CONFIG_WD0_DQ	R/W	0h	This register controls WD0 of DDR data pins Bit mapping is as below: 10: D(8) WD0 control 11: D(9) WD0 control 12: D(10) WD0 control 13: D(11) WD0 control 14: D(12) WD0 control 15: D(13) WD0 control 16: D(14) WD0 control 17: D(15) WD0 control
9-8	IO_CONFIG_SR_CLK	R/W	X	2 bit to program clock IO Pads (DQS1/DQ1S#) output Slew Rate. Refer to Table 7-8 "DDR Slew Rate Control Settings" for a description of these settings.
7-5	IO_CONFIG_I_CLK	R/W	0h	3 bit configuration input to program clock IO Pads (DQS1/DQS1#) output Impedance. Refer to Table 7-9 "DDR Impedance Control Settings" for a description of these settings.
4-3	IO_CONFIG_SR	R/W	X	2 bit to program data IO Pads output Slew Rate for D(15 to 8). Refer to Table 7-8 "DDR Slew Rate Control Settings" for a description of these settings.
2-0	IO_CONFIG_I	R/W	4h	3 bit configuration input to program data IO Pad output Impedance for D(15 to 8). Refer to Table 7-9 "DDR Impedance Control Settings" for a description of these settings.

7.3.1.275 CTRL_DDR_DATA2_IOCTRL Register (Offset = 1448h) [reset = X]

CTRL_DDR_DATA2_IOCTRL is shown in Figure 7-279 and described in Table 7-286.

Return to Summary Table.

Figure 7-279. CTRL_DDR_DATA2_IOCTRL Register

31	30	29	28	27	26	25	24
RESERVED		IO_CONFIG_WD1_DQS	IO_CONFIG_WD1_DM	IO_CONFIG_WD1_DQ			
R/W-0h		R/W-1h	R/W-1h	R/W-FFh			
23	22	21	20	19	18	17	16
IO_CONFIG_WD1_DQ				IO_CONFIG_WD0_DQS	IO_CONFIG_WD0_DM	IO_CONFIG_WD0_DQ	
R/W-FFh				R/W-0h	R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
IO_CONFIG_WD0_DQ						IO_CONFIG_SR_CLK	
R/W-0h						R/W-X	
7	6	5	4	3	2	1	0
IO_CONFIG_I_CLK			IO_CONFIG_SR		IO_CONFIG_I		
R/W-0h			R/W-X		R/W-4h		

Table 7-286. CTRL_DDR_DATA2_IOCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29	IO_CONFIG_WD1_DQS	R/W	1h	This register bit controls WD1 of DQS2 WD0 WD1 00 : Pullup/Pulldown disabled 01 : Weak pulldown enabled 10 : Weak pullup enabled 11 : Weak keeper enabled
28	IO_CONFIG_WD1_DM	R/W	1h	This register bit controls WD1 of DM2 WD0 WD1 00 : Pullup/Pulldown disabled 01 : Weak pulldown enabled 10 : Weak pullup enabled 11 : Weak keeper enabled
27-20	IO_CONFIG_WD1_DQ	R/W	FFh	This register controls WD1 of DDR data pins Bit mapping is as below: 20: D(16) WD1 control 21: D(17) WD1 control 22: D(18) WD1 control 23: D(19) WD1 control 24: D(20) WD1 control 25: D(21) WD1 control 26: D(22) WD1 control 27: D(23) WD1 control
19	IO_CONFIG_WD0_DQS	R/W	0h	This register bit controls WD0 of DQS2 WD0 WD1 00 : Pullup/Pulldown disabled 01 : Weak pulldown enabled 10 : Weak pullup enabled 11 : Weak keeper enabled
18	IO_CONFIG_WD0_DM	R/W	0h	This register bit controls WD0 of DM2 WD0 WD1 00 : Pullup/Pulldown disabled 01 : Weak pulldown enabled 10 : Weak pullup enabled 11 : Weak keeper enabled

Submit Documentation Feedback

Table 7-286. CTRL_DDR_DATA2_IOCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17-10	IO_CONFIG_WD0_DQ	R/W	0h	This register controls WD0 of DDR data pins Bit mapping is as below: 10: D(16) WD0 control 11: D(17) WD0 control 12: D(18) WD0 control 13: D(19) WD0 control 14: D(20) WD0 control 15: D(21) WD0 control 16: D(22) WD0 control 17: D(23) WD0 control
9-8	IO_CONFIG_SR_CLK	R/W	X	2 bit to program clock IO Pads (DQS2/DQS2#) output Slew Rate. Refer to Table 7-8 "DDR Slew Rate Control Settings" for a description of these settings.
7-5	IO_CONFIG_I_CLK	R/W	0h	3 bit configuration input to program clock IO Pads (DQS2/DQS2#) output Impedance. Refer to Table 7-9 "DDR Impedance Control Settings" for a description of these settings.
4-3	IO_CONFIG_SR	R/W	X	2 bit to program data IO Pads output Slew Rate for D(16 to 23). Refer to Table 7-8 "DDR Slew Rate Control Settings" for a description of these settings.
2-0	IO_CONFIG_I	R/W	4h	3 bit configuration input to program data IO Pad Impedance for D(23 to 16). Refer to Table 7-9 "DDR Impedance Control Settings" for a description of these settings.

7.3.1.276 CTRL_DDR_DATA3_IOCTRL Register (Offset = 144Ch) [reset = X]

CTRL_DDR_DATA3_IOCTRL is shown in Figure 7-280 and described in Table 7-287.

Return to Summary Table.

Figure 7-280. CTRL_DDR_DATA3_IOCTRL Register

31	30	29	28	27	26	25	24
RESERVED		IO_CONFIG_WD1_DQS	IO_CONFIG_WD1_DM	IO_CONFIG_WD1_DQ			
R/W-0h		R/W-1h	R/W-1h	R/W-FFh			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
IO_CONFIG_WD1_DQ				IO_CONFIG_WD0_DQS	IO_CONFIG_WD0_DM	IO_CONFIG_WD0_DQ	
R/W-FFh				R/W-0h	R/W-0h	R/W-0h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
IO_CONFIG_WD0_DQ						IO_CONFIG_SR_CLK	
R/W-0h						R/W-X	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IO_CONFIG_I_CLK			IO_CONFIG_SR		IO_CONFIG_I		
R/W-0h			R/W-X		R/W-4h		

Table 7-287. CTRL_DDR_DATA3_IOCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29	IO_CONFIG_WD1_DQS	R/W	1h	This register bit controls WD1 of DQS3 WD0 WD1 00 : Pullup/Pulldown disabled 01 : Weak pulldown enabled 10 : Weak pullup enabled 11 : Weak keeper enabled
28	IO_CONFIG_WD1_DM	R/W	1h	This register bit controls WD1 of DM3 WD0 WD1 00 : Pullup/Pulldown disabled 01 : Weak pulldown enabled 10 : Weak pullup enabled 11 : Weak keeper enabled
27-20	IO_CONFIG_WD1_DQ	R/W	FFh	This register controls WD1 of DDR data pins Bit mapping is as below: 20: D(24) WD1 control 21: D(25) WD1 control 22: D(26) WD1 control 23: D(27) WD1 control 24: D(28) WD1 control 25: D(29) WD1 control 26: D(30) WD1 control 27: D(31) WD1 control
19	IO_CONFIG_WD0_DQS	R/W	0h	This register bit controls WD0 of DQS3 WD0 WD1 00 : Pullup/Pulldown disabled 01 : Weak pulldown enabled 10 : Weak pullup enabled 11 : Weak keeper enabled
18	IO_CONFIG_WD0_DM	R/W	0h	This register bit controls WD0 of DM3 WD0 WD1 00 : Pullup/Pulldown disabled 01 : Weak pulldown enabled 10 : Weak pullup enabled 11 : Weak keeper enabled

Table 7-287. CTRL_DDR_DATA3_IOCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17-10	IO_CONFIG_WD0_DQ	R/W	0h	This register controls WD0 of DDR data pins Bit mapping is as below: 10: D(24) WD0 control 11: D(25) WD0 control 12: D(26) WD0 control 13: D(27) WD0 control 14: D(28) WD0 control 15: D(29) WD0 control 16: D(30) WD0 control 17: D(31) WD0 control
9-8	IO_CONFIG_SR_CLK	R/W	X	2 bit to program clock IO Pads (DQS3/DQS3#) output Slew Rate. Refer to Table 7-8 "DDR Slew Rate Control Settings" for a description of these settings.
7-5	IO_CONFIG_I_CLK	R/W	0h	3 bit configuration input to program clock IO Pads (DQS3/DQS3#) output Impedance. Refer to Table 7-9 "DDR Impedance Control Settings" for a description of these settings.
4-3	IO_CONFIG_SR	R/W	X	2 bit to program data IO Pads output Slew Rate for D(31 to 24). Refer to Table 7-8 "DDR Slew Rate Control Settings" for a description of these settings.
2-0	IO_CONFIG_I	R/W	4h	3 bit configuration input to program data IO Pad Impedance for D(31 to 24). Refer to Table 7-9 "DDR Impedance Control Settings" for a description of these settings.

7.3.1.277 CTRL_EMIF_SDRAM_CONFIG_EXT Register (Offset = 1460h) [reset = 3h]

Register mask: FFFFFFFFh

CTRL_EMIF_SDRAM_CONFIG_EXT is shown in Figure 7-281 and described in Table 7-288.

Return to Summary Table.

Figure 7-281. CTRL_EMIF_SDRAM_CONFIG_EXT Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED						NARROW_ONLY	EN_ECC
R/W-0h						R/W-0h	R-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PHY_NUM_OF_SAMPLES		PHY_SEL_LOGIC	PHY_ALL_DQ_MPR_RD_RESP	PHY_OUTPUT_STS_SELECT			DYNAMIC_PWRDN_EN
R/W-0h		R/W-0h	R/W-0h	R/W-0h			R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	PHY_RD_LOCAL_ODT		RESERVED	DFI_CLOCK_PHASE_CTRL	RESERVED	EN_SLICE_1	EN_SLICE_0
R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h

Table 7-288. CTRL_EMIF_SDRAM_CONFIG_EXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17	NARROW_ONLY	R/W	0h	Should be 1 if EMIF is operated in Narrow mode (16 bit only).
16	EN_ECC	R	0h	ECC not supported - Reserved
15-14	PHY_NUM_OF_SAMPLES	R/W	0h	Controls the number of samples used during read data eye training Should be 0x0 (4 samples) for incremental leveling and 0x3 (128 samples) for full leveling
13	PHY_SEL_LOGIC	R/W	0h	Selects data read eye training algorithm Value 0(algorithm #1) is recommended
12	PHY_ALL_DQ_MPR_RD_RESP	R/W	0h	Controls the number of DQ pins used during read data eye training Value 1 (all DQs used) is recommended if the memory provides a leveling response on all DQs value 0 (only one DQ) recommended if memory only provides a single DQ
11-9	PHY_OUTPUT_STS_SELECT	R/W	0h	Use to select the status to be observed on the spare_out pins through EMIF_SDRAM_STATUS_EXT register 0: phy_reg_rdlvl_start_ratio(7 to 0) 1: phy_reg_rdlvl_start_ratio(15 to 8) 2: phy_reg_rdlvl_end_ratio(7 to 0) 3: phy_reg_rdlvl_end_ratio(15 to 8)
8	DYNAMIC_PWRDN_EN	R/W	0h	Enables dynamic PWRDN control in the IOs to reduce power consumption
7	RESERVED	R/W	0h	

Table 7-288. CTRL_EMIF_SDRAM_CONFIG_EXT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6-5	PHY_RD_LOCAL_ODT	R/W	0h	Value to drive on the 2-bit local_odt PHY outputs when output enable is not asserted and a read is in progress (where in progress is defined as after a read command is issued and until all read data has been returned all the way to the controller.) Typically this is set to the value required to enable termination at the desired strength for read usage. 00 = ODT off. 01 = ODT off. 10 = Full thevenin load. This is equal to the same drive strength defined by the ddr_data0_ioctrl, ddr_data1_ioctrl, ddr_data2_ioctrl registers. Refer to Table 7-9 "DDR Impedance Control Settings" for a description of these settings. 11 = Half thevenin load. This is equal to half the drive strength defined by the ddr_data0_ioctrl, ddr_data1_ioctrl, ddr_data2_ioctrl registers. Refer to Table 7-9 "DDR Impedance Control Settings" for a description of these settings.
4	RESERVED	R/W	0h	
3	DFI_CLOCK_PHASE_CTRL	R/W	0h	DFI clock division phase control in EMIF4D5SS Recommended 0
2	RESERVED	R/W	0h	
1	EN_SLICE_1	R/W	1h	Enable CMD PHY1
0	EN_SLICE_0	R/W	1h	Enable CMD PHY0

7.3.1.278 CTRL_EMIF_SDRAM_STS_EXT Register (Offset = 1464h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_EMIF_SDRAM_STS_EXT is shown in Figure 7-282 and described in Table 7-289.

Return to Summary Table.

Figure 7-282. CTRL_EMIF_SDRAM_STS_EXT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG																															
R-0h																															

Table 7-289. CTRL_EMIF_SDRAM_STS_EXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REG	R	0h	The bit mapping for this register, which connects to spareout of memss, is listed below: Bits [31 to 24] - Slice3 Bits [23 to 16] - Slice2 Bits [15 to 8] - Slice1 Bits [7 to 0] - Slice0 When {EMIF_SDRAM_CFG_EXT[11:9]=0x0}, this status register reflects phy_reg_rdlvl_start_ratio[7:0] for the corresponding 8bit slice. When {EMIF_SDRAM_CFG_EXT[11:9]= 0x1}, this status register reflects phy_reg_rdlvl_start_ratio[15:8] for the corresponding 8bit slice. When {EMIF_SDRAM_CFG_EXT[11:9]= 0x2}, this status register reflects phy_reg_rdlvl_end_ratio[7:0] for the corresponding 8bit slice. When {EMIF_SDRAM_CFG_EXT[11:9]= 0x3}, this status register reflects phy_reg_rdlvl_end_ratio[15:8] for the corresponding 8bit slice.

7.3.1.279 CTRL_DISPPLL_CLKCTRL Register (Offset = 3000h) [reset = 00910001h]

Register mask: FFFFFFFFh

CTRL_DISPPLL_CLKCTRL is shown in Figure 7-283 and described in Table 7-290.

Return to Summary Table.

Figure 7-283. CTRL_DISPPLL_CLKCTRL Register

31	30	29	28	27	26	25	24
CYCLESLIPEN	ENSSC	RESERVED	NWELLTRIM				
R/W-0h	R/W-0h	R/W-0h	R/W-0h				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
IDLE	BYPASSACKZ	STBYRET	CLKOUTEN	RESERVED	ULOWCLKEN	CLKDCOLDOPWDNZ	M2PWDNZ
R/W-1h	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h	R/W-1h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
M3PWDNZ	STOPMODE	LOWCURRSTDBY	LPMODE	DRIFTGUARDEN	REGM4XEN	RESERVED	RELAXED_LOCK
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							TINITZ
R/W-0h							R/W-1h

Table 7-290. CTRL_DISPPLL_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CYCLESLIPEN	R/W	0h	FailSafe enable to trigger re-calibration in case CycleSlip occurs between REFCLK & FBCLK.
30	ENSSC	R/W	0h	Controls Clock Spreading 1: Enables clock spreading 0: Disables clock spreading
29	RESERVED	R/W	0h	
28-24	NWELLTRIM	R/W	0h	Trim values for the PLL
23	IDLE	R/W	1h	Sets PLL to Idle mode. When SYSRESET = 0 and TINITZ = 1 0: PLL will go to Active and Locked 1: PLL will go to Idle Bypass low power
22	BYPASSACKZ	R/W	0h	Bypass status acknowledge signal.
21	STBYRET	R/W	0h	Standby retention control 1: Prepares ADPLLLJ for retention by gating all the internal clocks 0: Prepares ADPLLLJ for relock when out of retention by removing the gating on all internal clocks
20	CLKOUTEN	R/W	1h	CLKOUT enable or disable 1: Synchronously enables CLKOUT 0: Synchronously disables CLKOUT
19	RESERVED	R/W	0h	
18	ULOWCLKEN	R/W	0h	Select CLKOUT source in bypass 0: When ADPLLLJ in bypass mode CLKOUT = CLKINP/(N2+1) 1: When ADPLLLJ in bypass mode CLKOUT = CLKINPULOW
17	CLKDCOLDOPWDNZ	R/W	0h	0: Asynchronous power down for CLKDCOLDO o/p.
16	M2PWDNZ	R/W	1h	0: Asynchronous power down for M2 divider 1: M2 divider is functional
15	M3PWDNZ	R/W	0h	0: Asynchronous power down for M3 divider 1: M3 divider is functional
14	STOPMODE	R/W	0h	When in Lossclk/Stbyret 0: Limp mode 1: Stopmode

Table 7-290. CTRL_DISPPLL_CLKCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	LOWCURRSTDBY	R/W	0h	When in Lossclk/Stbyret/Idle 0: Fast relock 1: Slow relock
12	LPMODE	R/W	0h	Can be set to '1' in cases with CLKINP/(N+1) = 1MHz and CLKINP*M/(N+1) = 100MHz
11	DRIFTGUARDEN	R/W	0h	When RECAL status flag is asserted '1' enables recalibration
10	REGM4XEN	R/W	0h	Enable REGM*4 (Active High). Can be set to '1' only for CLKOUT*M2 > 150MHz when LPMODE = '0' and CLKOUT*M2 > 60MHz when LPMODE = '1'. This bit should not be changed on the fly in locked condition. INITIALIZATION should follow change of this bit.
9	RESERVED	R/W	0h	
8	RELAXED_LOCK	R/W	0h	0: FREQLOCK asserted when DC frequency error 1% 1: FREQLOCK asserted when DC frequency error 2%
7-1	RESERVED	R/W	0h	
0	TINITZ	R/W	1h	PLL core soft reset

7.3.1.280 CTRL_DISPPLL_TEN Register (Offset = 3004h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_DISPPLL_TEN is shown in Figure 7-284 and described in Table 7-291.

Return to Summary Table.

Figure 7-284. CTRL_DISPPLL_TEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															TEN
R/W-0h															R/W-0h

Table 7-291. CTRL_DISPPLL_TEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	TEN	R/W	0h	M N. SD and SELFREQDCO latch (active rise edge)

7.3.1.281 CTRL_DISPPLL_TENIV Register (Offset = 3008h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_DISPPLL_TENIV is shown in Figure 7-285 and described in Table 7-292.

Return to Summary Table.

Figure 7-285. CTRL_DISPPLL_TENIV Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED							TENIV
R/W-0h							R/W-0h

Table 7-292. CTRL_DISPPLL_TENIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	TENIV	R/W	0h	M2 and N2 latch (active rise edge)

7.3.1.282 CTRL_DISPPLL_M2NDIV Register (Offset = 300Ch) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_DISPPLL_M2NDIV is shown in Figure 7-286 and described in Table 7-293.

Return to Summary Table.

Figure 7-286. CTRL_DISPPLL_M2NDIV Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="9">RESERVED</td><td colspan="7">M2</td><td colspan="8">RESERVED</td><td colspan="8">N</td></tr>
<tr><td colspan="9">R/W-0h</td><td colspan="7">R/W-0h</td><td colspan="8">R/W-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 7-293. CTRL_DISPPLL_M2NDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	0h	
22-16	M2	R/W	0h	Post-divider is REGM2
15-8	RESERVED	R/W	0h	
7-0	N	R/W	0h	Pre-divider is REGN+1

7.3.1.283 CTRL_DISPPLL_MN2DIV Register (Offset = 3010h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_DISPPLL_MN2DIV is shown in Figure 7-287 and described in Table 7-294.

Return to Summary Table.

Figure 7-287. CTRL_DISPPLL_MN2DIV Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="12">RESERVED</td><td colspan="4">N2</td><td colspan="4">RESERVED</td><td colspan="12">M</td></tr>
<tr><td colspan="12">R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="12">R/W-0h</td></tr>
</table>

Table 7-294. CTRL_DISPPLL_MN2DIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19-16	N2	R/W	0h	Bypass divider is REGN2+1
15-12	RESERVED	R/W	0h	
11-0	M	R/W	0h	Feedback multiplier is REGM

7.3.1.284 CTRL_DISPPLL_FRACDIV Register (Offset = 3014h) [reset = X]

CTRL_DISPPLL_FRACDIV is shown in Figure 7-288 and described in Table 7-295.

Return to Summary Table.

Figure 7-288. CTRL_DISPPLL_FRACDIV Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">REGSD</td><td colspan="6">RESERVED</td><td colspan="18">FRACTIONALM</td></tr>
<tr><td colspan="8">R-X</td><td colspan="6">R/W-0h</td><td colspan="18">R/W-0h</td></tr>
</table>

Table 7-295. CTRL_DISPPLL_FRACDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	REGSD	R	X	Sigma-Delta Divider
23-18	RESERVED	R/W	0h	
17-0	FRACTIONALM	R/W	0h	Fractional part of the M divider.

7.3.1.285 CTRL_DISPPLL_BWCTRL Register (Offset = 3018h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_DISPPLL_BWCTRL is shown in Figure 7-289 and described in Table 7-296.

Return to Summary Table.

Figure 7-289. CTRL_DISPPLL_BWCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED													BWCTRL		RESERVED
R/W-0h													R/W-0h		R/W-0h

Table 7-296. CTRL_DISPPLL_BWCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-1	BWCTRL	R/W	0h	Change Loop Bandwidth
0	RESERVED	R/W	0h	

7.3.1.286 CTRL_DISPPLL_FRACCTRL Register (Offset = 301Ch) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_DISPPLL_FRACCTRL is shown in Figure 7-290 and described in Table 7-297.

Return to Summary Table.

Figure 7-290. CTRL_DISPPLL_FRACCTRL Register

31	30	29	28	27	26	25	24
DOWNSPREAD	MODFREQDIVIDEREXPONENT			MODFREQDIVIDERMANTISSA			
R/W-0h	R/W-0h			R/W-0h			
23	22	21	20	19	18	17	16
MODFREQDIVIDERMANTISSA			DELTAMSTEPINTEGER			DELTAMSTEPFRACTION	
R/W-0h			R/W-0h			R/W-0h	
15	14	13	12	11	10	9	8
DELTAMSTEPFRACTION							
R/W-0h							
7	6	5	4	3	2	1	0
DELTAMSTEPFRACTION							
R/W-0h							

Table 7-297. CTRL_DISPPLL_FRACCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DOWNSPREAD	R/W	0h	1: Enables low frequency spread only 0: Enables both side frequency spread about the programmed frequency
30-28	MODFREQDIVIDEREXPONENT	R/W	0h	Exponent of the REFCLK divider to define the modulation frequency.
27-21	MODFREQDIVIDERMANTISSA	R/W	0h	Mantissa of the REFCLK divider to define the modulation frequency
20-18	DELTAMSTEPINTEGER	R/W	0h	Integer part of Frequency Spread control.
17-0	DELTAMSTEPFRACTION	R/W	0h	The fraction part of Frequency Spread control

7.3.1.287 CTRL_DISPPLL_STS Register (Offset = 3020h) [reset = X]

CTRL_DISPPLL_STS is shown in Figure 7-291 and described in Table 7-298.

Return to Summary Table.

Figure 7-291. CTRL_DISPPLL_STS Register

31	30	29	28	27	26	25	24
RESERVED	SSACK	LDOPWDN	RECAL_BSTS3	RECAL_PIN	RESERVED		
R-0h	R-X	R-X	R-X	R-X	R-0h		

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED					PHASELOCK	FREQLOCK	BYPASSACK
R-0h					R-X	R-X	R-X

7	6	5	4	3	2	1	0
STBYRETACK	LOSSREF	CLKOUTACK	LOCK2	M2CHANGEACK	LIMP	HIGHJITTER	BYPASS
R-X	R-X	R-X	R-X	R-X	R-X	R-X	R-X

Table 7-298. CTRL_DISPPLL_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30	SSACK	R	X	1: Spread-spectrum Clocking is enabled on output clocks (default) 0: Spread-spectrum Clocking is disabled on output clocks
29	LDOPWDN	R	X	1: Indicates ADPLLLJ internal LDO is power down. VDDLDOOUT will be un-defined in this condition. (default)
28	RECAL_BSTS3	R	X	Recalibration status flag. 1: ADPLLLJ requires recalibration)
27	RECAL_PIN	R	X	Recalibration status flag. 1: ADPLLLJ requires recalibration)
26-11	RESERVED	R	0h	
10	PHASELOCK	R	X	Status on PHASELOCK output pin
9	FREQLOCK	R	X	Status on FREQLOCK output pin
8	BYPASSACK	R	X	Status of BYPASSACK output pin (default:1)
7	STBYRETACK	R	X	1: Indicates to SOC that all internal clocks in ADPLLLJ are gated and it is ready for retention. 0: Indicates to SOC that all internal clocks in ADPLLLJ are active and it is starting the relock process.
6	LOSSREF	R	X	Reference input loss
5	CLKOUTACK	R	X	1/ 0: Indicates enable/disable condition of CLKOUT (default:1)
4	LOCK2	R	X	ADPLL internal loop lock status
3	M2CHANGEACK	R	X	acknowledge for M2 change
2	LIMP	R	X	1: In LIMP mode 0: In Stop Mode
1	HIGHJITTER	R	X	1: Indicates jitter. After PHASELOCK is asserted high the HIGHJITTER flag is asserted high if phase error between REFCLK and FBCLK 24%.
0	BYPASS	R	X	Bypass status signal. 1: CLKOUT in bypass

7.3.1.288 CTRL_DISPPLL_M3DIV Register (Offset = 3024h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_DISPPLL_M3DIV is shown in Figure 7-292 and described in Table 7-299.

Return to Summary Table.

Figure 7-292. CTRL_DISPPLL_M3DIV Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="27">RESERVED</td><td colspan="5">M3</td></tr>
<tr><td colspan="27">R/W-0h</td><td colspan="5">R/W-0h</td></tr>
</table>

Table 7-299. CTRL_DISPPLL_M3DIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4-0	M3	R/W	0h	Post Divider Reg M3

7.3.1.289 CTRL_DISPPLL_RAMPCTRL Register (Offset = 3028h) [reset = 0h]

Register mask: FFFFFFFFh

CTRL_DISPPLL_RAMPCTRL is shown in Figure 7-293 and described in Table 7-300.

Return to Summary Table.

Figure 7-293. CTRL_DISPPLL_RAMPCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED			CLKRAMPLEVEL		CLKRAMPRATE		
R/W-0h			R/W-0h		R/W-0h		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							RELOCK_RAMP_EN
R/W-0h							R/W-0h

Table 7-300. CTRL_DISPPLL_RAMPCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R/W	0h	
20-19	CLKRAMPLEVEL	R/W	0h	Controls the ramp sequence. 00: No ramping 01: Bypass clk /Fout/8 / Fout/4 / Fout/2 / Fout 10: Bypass clk / Fout/4 / Fout/2 / Fout/1.5 /Fout 11: Reserved
18-16	CLKRAMPRATE	R/W	0h	Controls the time spent on each ramp step. 000: 2 REFCLKs 001: 4 REFCLKs 010: 8 REFCLKs 011: 16 REFCLKs 100: 32 REFCLKs 101: 64 REFCLKs 110: 128 REFCLKs 111: 512 REFCLKs
15-1	RESERVED	R/W	0h	
0	RELOCK_RAMP_EN	R/W	0h	

Interrupts

This section describes the interrupts for the device.

Topic **Page**

8.1 ARM Cortex-A9 Interrupts

Table 8-1. ARM Cortex-A9 Interrupts

Int Number	Acronym/name	Source	Signal Name
32	PL310_IRQ	Level-2 Cache Controller interrupt	PL310_IRQ
33	CTIIRQ0	TRIGOUT[6] of Cross Trigger Interface 0 (CTI0)	CTIIRQ[0]
34	Reserved		
35	Reserved		
36	ELM_IRQ	ELM	Sinterrupt (Error location process completion)
37	Reserved		
38	Reserved		
39	NMI	External Pin (Primary Input)	nmi_int
40	Reserved		
41	L3DEBUG	L3	l3_FlagMux_top_FlagOut1
42	L3APPINT	L3	l3_FlagMux_top_FlagOut0
43	PRCMINT	PRCM	irq_mpu
44	EDMACOMPINT	EDMA3CC (TPCC)	tpcc_int_pend_po0
45	EDMAMPERR	EDMA3CC (TPCC)	tpcc_mpint_pend_po
46	EDMAERRINT	EDMA3CC (TPCC)	tpcc_errint_pend_po
47	Reserved		
48	ADC0_GENINT	ADC0	gen_intr_pend
49	Reserved		
50	Reserved		
51	Reserved		
52	PRU_ICSS1_EVTOUT0	pr1_host[0] output/events exported from PRU_ICSS1 [1]	pr1_host_intr0_intr_pend
53	PRU_ICSS1_EVTOUT1	pr1_host[1] output/events exported from PRU_ICSS1 [1]	pr1_host_intr1_intr_pend
54	PRU_ICSS1_EVTOUT2	pr1_host[2] output/events exported from PRU_ICSS1 [1]	pr1_host_intr2_intr_pend
55	PRU_ICSS1_EVTOUT3	pr1_host[3] output/events exported from PRU_ICSS1 [1]	pr1_host_intr3_intr_pend
56	PRU_ICSS1_EVTOUT4	pr1_host[4] output/events exported from PRU_ICSS1 [1]	pr1_host_intr4_intr_pend
57	Reserved		
58	PRU_ICSS1_EVTOUT6	pr1_host[6] output/events exported from PRU_ICSS1 [1]	pr1_host_intr6_intr_pend
59	PRU_ICSS1_EVTOUT7	pr1_host[7] output/events exported from PRU_ICSS1 [1]	pr1_host_intr7_intr_pend
60	MMCSD1INT	MMCSD1	SINTERRUPTN
61	MMCSD2INT	MMCSD2	SINTERRUPTN
62	I2C2INT	I2C2	POINTRPEND
63	eCAP0INT	PWMSS0_ECAP	ecap_intr_intr_pend
64	GPIOINT2A	GPIO2	POINTRPEND1
65	GPIOINT2B	GPIO2	POINTRPEND2
66	Reserved		
67	Reserved		
68	Reserved		
69	GFXINT	GFX (SGX530)	THALIAIRQ
70	Reserved		

Table 8-1. ARM Cortex-A9 Interrupts (continued)

Int Number	Acronym/name	Source	Signal Name
71	ePWM2INT	PWMSS2_EPWM (event/interrupt)	epwm_intr_intr_pend
72	3PGSWRXTHR0	CPSW (Ethernet)	c0_rx_thresh_pend
73	3PGSWRXINT0	CPSW (Ethernet)	c0_rx_pend
74	3PGSWTXINT0	CPSW (Ethernet)	c0_tx_pend
75	3PGSWMISC0	CPSW (Ethernet)	c0_misc_pend
76	UART3INT	UART3	niq
77	UART4INT	UART4	niq
78	UART5INT	UART5	niq
79	eCAP1INT	PWMSS1_ECAP	ecap_intr_intr_pend
80	CCDC0_INT	VPFE0 (Camera0)	ccdc_intr_pend
81	DCAN1_INT0	DCAN1	dcan_intr0_intr_pend
82	CCDC1_INT	VPFE1 (Camera0)	ccdc_intr_pend
83	DCAN0_PARITY	DCAN0	dcan_uerr_intr_pend
84	DCAN0_INT0	DCAN0	dcan_intr0_intr_pend
85	DCAN0_INT1	DCAN0	dcan_intr1_intr_pend
86	Reserved		
87	Reserved		
88	DCAN1_INT1	DCAN1	dcan_intr1_intr_pend
89	DCAN1_PARITY	DCAN1	dcan_uerr_intr_pend
90	ePWM0_TZINT	PWMSS0_EPWM (tz interrupt)	epwm_tz_intr_pend
91	ePWM1_TZINT	PWMSS1_EPWM (tz interrupt)	epwm_tz_intr_pend
92	ePWM2_TZINT	PWMSS2_EPWM (tz interrupt)	epwm_tz_intr_pend
93	eCAP2INT	PWMSS2_ECAP	ecap_intr_intr_pend
94	GPIOINT3A	GPIO3	POINTRPEND1
95	GPIOINT3B	GPIO3	POINTRPEND2
96	MMCSD0INT	MMCSD0	SINTERRUPTN
97	SPI0INT	McSPI0	SINTERRUPTN
98	TINT0	DMTimer0	POINTR_PEND
99	TINT1_1MS	DMTimer1_1MS	POINTR_PEND
100	TINT2	DMTimer2	POINTR_PEND
101	TINT3	DMTimer3	POINTR_PEND
102	I2C0INT	I2C0	POINTRPEND
103	I2C1INT	I2C1	POINTRPEND
104	UAR0TINT	UART0	niq
105	UART1INT	UART1	niq
106	UART2INT	UART2	niq
107	RTCINT	RTC	timer_intr_pend
108	RTCALARMINT	RTC	alarm_intr_pend
109	MBINT0	Mailbox0 (mail_u0_irq)	initiator_sinterrupt_q_n0
110	Reserved		
111	eQEP0INT	PWMSS0_EQEP	eqep_intr_intr_pend
112	MCATXINT0	McASP0	mcasp_x_intr_pend
113	MCARXINT0	McASP0	mcasp_r_intr_pend
114	MCATXINT1	McASP1	mcasp_x_intr_pend
115	MCARXINT1	McASP1	mcasp_r_intr_pend
116	Reserved		
117	Reserved		

Table 8-1. ARM Cortex-A9 Interrupts (continued)

Int Number	Acronym/name	Source	Signal Name
118	ePWM0INT	PWMSS0_EPWM (event/interrupt)	epwm_intr_intr_pend
119	ePWM1INT	PWMSS1_EPWM (event/interrupt)	epwm_intr_intr_pend
120	eQEP1INT	PWMSS1_EQEP	eqep_intr_intr_pend
121	eQEP2INT	PWMSS2_EQEP	eqep_intr_intr_pend
122	DMA_INTR_PIN2	External DMA/Interrupt Pin2	pi_x_dma_event_intr2
123	WDT1INT	WDT1 (WDTimer1)	PO_INT_PEND
124	TINT4	DMTimer4	POINTR_PEND
125	TINT5	DMTimer5	POINTR_PEND
126	TINT6	DMTimer6	POINTR_PEND
127	TINT7	DMTimer7	POINTR_PEND
128	GPIOINT0A	GPIO0	POINTRPEND1
129	GPIOINT0B	GPIO0	POINTRPEND2
130	GPIOINT1A	GPIO1	POINTRPEND1
131	GPIOINT1B	GPIO1	POINTRPEND2
132	GPMCINT	GPMC	gpmc_sinterrupt
133	DDRERR0	DDR EMIF0	sys_err_intr_pend
134	Reserved		
135	Reserved		
136	Reserved		
137	Reserved		
138	GPIOINT4A	GPIO4	POINTRPEND1
139	GPIOINT4B	GPIO4	POINTRPEND2
140	Reserved		
141	Reserved		
142	Reserved		
143	Reserved		
144	TCERRINT0	EDMA3TC0	tptc_erint_pend_po
145	TCERRINT1	EDMA3TC1	tptc_erint_pend_po
146	TCERRINT2	EDMA3TC2	tptc_erint_pend_po
147	ADC1_GENINT	ADC1	gen_intr_pend
148	Reserved		
149	Reserved		
150	Reserved		
151	Reserved		
152	Reserved		
153	Reserved		
154	Reserved		
155	DMA_INTR_PIN0	External DMA/Interrupt Pin0 (xdma_event_intr0)	pi_x_dma_event_intr0
156	DMA_INTR_PIN1	External DMA/Interrupt Pin1 (xdma_event_intr1)	pi_x_dma_event_intr1
157	SPI1INT	McSPI1	SINTERRUPTN
158	SPI2INT	McSPI2	SINTERRUPTN
159	DSSINT	DSS (Display SS)	dss_irq
160	Reserved		
161	Reserved		
162	Reserved		

Table 8-1. ARM Cortex-A9 Interrupts (continued)

Int Number	Acronym/name	Source	Signal Name
163	TINT8	DMTimer8	POINTR_PEND
164	TINT9	DMTimer9	POINTR_PEND
165	TINT10	DMTimer10	POINTR_PEND
166	TINT11	DMTimer11	POINTR_PEND
167	Reserved		
168	SPI3INT	McSPI3	SINTERRUPTN
169	SPI4INT	McSPI4	SINTERRUPTN
170	QSPIINT	QSPI	intr_pend
171	HDQINT	HDQ1W	hdq_irq
172	Reserved		
173	ePWM3INT	PWMSS3_EPWM (event/interrupt)	epwm_intr_intr_pend
174	ePWM3_TZINT	PWMSS3_EPWM (tz event)	epwm_tz_intr_pend
175	ePWM4INT	PWMSS4_EPWM (event/interrupt)	epwm_intr_intr_pend
176	ePWM4_TZINT	PWMSS4_EPWM (tz event)	epwm_tz_intr_pend
177	ePWM5INT	PWMSS5_EPWM (event/interrupt)	epwm_intr_intr_pend
178	ePWM5_TZINT	PWMSS5_EPWM (tz event)	epwm_tz_intr_pend
179	Reserved		
180	GPIOINT5A	GPIO5	POINTRPEND1
181	GPIOINT5B	GPIO5	POINTRPEND2
182	DMA_INTR_PIN3	External DMA/Interrupt Pin3 (xdma_event_intr3)	pi_x_dma_event_intr3
183	DMA_INTR_PIN4	External DMA/Interrupt Pin4 (xdma_event_intr4)	pi_x_dma_event_intr4
184	DMA_INTR_PIN5	External DMA/Interrupt Pin5 (xdma_event_intr5)	pi_x_dma_event_intr5
185	DMA_INTR_PIN6	External DMA/Interrupt Pin6 (xdma_event_intr6)	pi_x_dma_event_intr6
186	DMA_INTR_PIN7	External DMA/Interrupt Pin7 (xdma_event_intr7)	pi_x_dma_event_intr7
187	DMA_INTR_PIN8	External DMA/Interrupt Pin8 (xdma_event_intr8)	pi_x_dma_event_intr8
188	Reserved		
189	Reserved		
190	Reserved		
191	PRU_ICSS0_EVTOUT0	pr0_host[0] output/events exported from PRU_ICSS0 [2]	pr0_host_intr0_intr_pend
192	PRU_ICSS0_EVTOUT1	pr0_host[1] output/events exported from PRU_ICSS0 [2]	pr0_host_intr1_intr_pend
193	PRU_ICSS0_EVTOUT2	pr0_host[2] output/events exported from PRU_ICSS0 [2]	pr0_host_intr2_intr_pend
194	PRU_ICSS0_EVTOUT3	pr0_host[3] output/events exported from PRU_ICSS0 [2]	pr0_host_intr3_intr_pend
195	PRU_ICSS0_EVTOUT4	pr0_host[4] output/events exported from PRU_ICSS0 [2]	pr0_host_intr4_intr_pend
196	PRU_ICSS0_EVTOUT6	pr0_host[6] output/events exported from PRU_ICSS0 [2]	pr0_host_intr6_intr_pend
197	PRU_ICSS0_EVTOUT7	pr0_host[7] output/events exported from PRU_ICSS0 [2]	pr0_host_intr7_intr_pend
198	Reserved		

Table 8-1. ARM Cortex-A9 Interrupts (continued)

Int Number	Acronym/name	Source	Signal Name
199	Reserved		
200	USB0_MAIN0_INT	USB0	main0_intr_pend
201	USB0_MAIN1_INT	USB0	main1_intr_pend
202	USB0_MAIN2_INT	USB0	main2_intr_pend
203	USB0_MAIN3_INT	USB0	main3_intr_pend
204	USB0_MISC_INT	USB0	misc_intr_pend
205	Reserved		
206	USB1_MAIN0_INT	USB1	main0_intr_pend
207	USB1_MAIN1_INT	USB1	main1_intr_pend
208	USB1_MAIN2_INT	USB1	main2_intr_pend
209	USB1_MAIN3_INT	USB1	main3_intr_pend
210	USB1_MISC_INT	USB1	misc_intr_pend
211	Reserved		

(1) pr1_host_intr[0:7] corresponds to Host-2 to Host-9 of the PRU-ICSS interrupt controller.

(2) pr0_host_intr[0:7] corresponds to Host-2 to Host-9 of the PRU-ICSS interrupt controller.

8.2 PWM Events

Table 8-2. Timer and eCAP Event Capture

Event #	IP	Interrupt Name/Pin
0	For Timer 5 MUX input from IO signal TIMER5	TIMER5 IO pin
	For Timer 6 MUX input from IO signal TIMER6	TIMER6 IO pin
	For Timer 7 MUX input from IO signal TIMER7	TIMER7 IO pin
	For eCAP 0 MUX input from IO signal eCAP0	eCAP0 IO pin
	For eCAP 1 MUX input from IO signal eCAP1	eCAP1 IO pin
	For eCAP 2 MUX input from IO signal eCAP2	eCAP2 IO pin
1	UART0	UART0INT
2	UART1	UART1INT
3	UART2	UART2INT
4	UART3	UART3INT
5	UART4	UART4INT
6	UART5	UART5INT
7	3PGSW	3PGSWRXTHR0
8	3PGSW	3PGSWRXINT0
9	3PGSW	3PGSWTXINT0
10	3PGSW	3PGSWMISC0
11	McASP0	MCATXINT0
12	McASP0	MCARXINT0
13	McASP1	MCATXINT1
14	McASP1	MCARXINT1
15	GPIO4	GPIOINT4A
16	GPIO4	GPIOINT4B

Table 8-2. Timer and eCAP Event Capture (continued)

Event #	IP	Interrupt Name/Pin
17	GPIO0	GPIOINT0A
18	GPIO0	GPIOINT0B
19	GPIO1	GPIOINT1A
20	GPIO1	GPIOINT1B
21	GPIO2	GPIOINT2A
22	GPIO2	GPIOINT2B
23	GPIO3	GPIOINT3A
24	GPIO3	GPIOINT3B
25	DCAN0	DCAN0_INT0
26	DCAN0	DCAN0_INT1
27	DCAN0	DCAN0_PARITY
28	DCAN1	DCAN1_INT0
29	DCAN1	DCAN1_INT1
30	DCAN1	DCAN1_PARITY

Memory Subsystem

This chapter describes the memory subsystem of the device.

Topic **Page**

9.1 GPMC

9.1.1 Introduction

The general-purpose memory controller (GPMC) is an unified memory controller dedicated to interfacing external memory devices:

- Asynchronous SRAM-like memories and application-specific integrated circuit (ASIC) devices
- Asynchronous, synchronous, and page mode (only available in non-multiplexed mode) burst NOR flash devices
- NAND Flash
- Pseudo-SRAM devices

9.1.1.1 GPMC Features

The general features of the GPMC module include:

- Data path to external memory device can be 16- or 8-bit wide
- 32-bit OCPIP 2.0 compliant core, single slave interface. Support non-wrapping and wrapping burst up to 16x32bits.
- Up to 100 MHz external memory clock performance (single device)
- Support for the following memory types:
  - External asynchronous or synchronous 8-bit width memory or device (non burst device)
  - External asynchronous or synchronous 16-bit width memory or device
  - External 16-bit non-multiplexed NOR Flash device
  - External 16-bit address and data multiplexed NOR Flash device
  - External 8-bit and 16-bit NAND flash device
  - External 16-bit pSRAM device
- Up to 16-bit ECC support for NAND flash using BCH code (t=4, 8 or 16) or Hamming code for 8-bit or 16-bit NAND-flash, organized with page size of 512 bytes, 1K bytes, or more.
- Support 512M Bytes maximum addressing capability which can be divided into seven independent chip-select with programmable bank size and base address on 16M Bytes, 32M Bytes, 64M Bytes, or 128M Bytes boundary
- Fully pipelined operation for optimal memory bandwidth usage
- Support external device clock frequency of 1, 2, 3 and 4 divider from L3 clock.
- Support programmable auto-clock gating when there is no access.
- Support Midlereq/SidleAck protocol
- Support the following interface protocols when communicating with external memory or external devices.
  - Asynchronous read/write access
  - Asynchronous read page access (4-8-16 Word16)
  - Synchronous read/write access
  - Synchronous read burst access without wrap capability (4-8-16 Word16)
  - Synchronous read burst access with wrap capability (4-8-16 Word16)
- Address and Data multiplexed access
- Each chip-select as independent and programmable control signal timing parameters for Setup and Hold time. Parameters are set according to the memory device timing parameters, with one L3 clock cycle timing granularity.
- Flexible internal access time control (wait state) and flexible handshake mode using external WAIT pins monitoring (up to two WAIT pins)
- Support bus keeping
- Support bus turn around

- Pre-fetch and write posting engine associated with system DMA to get full performance from NAND device with minimum impact on NOR/SRAM concurrent access.
- On the fly ECC Hamming Code calculation to improve NAND usage reliability with minimum impact on SW

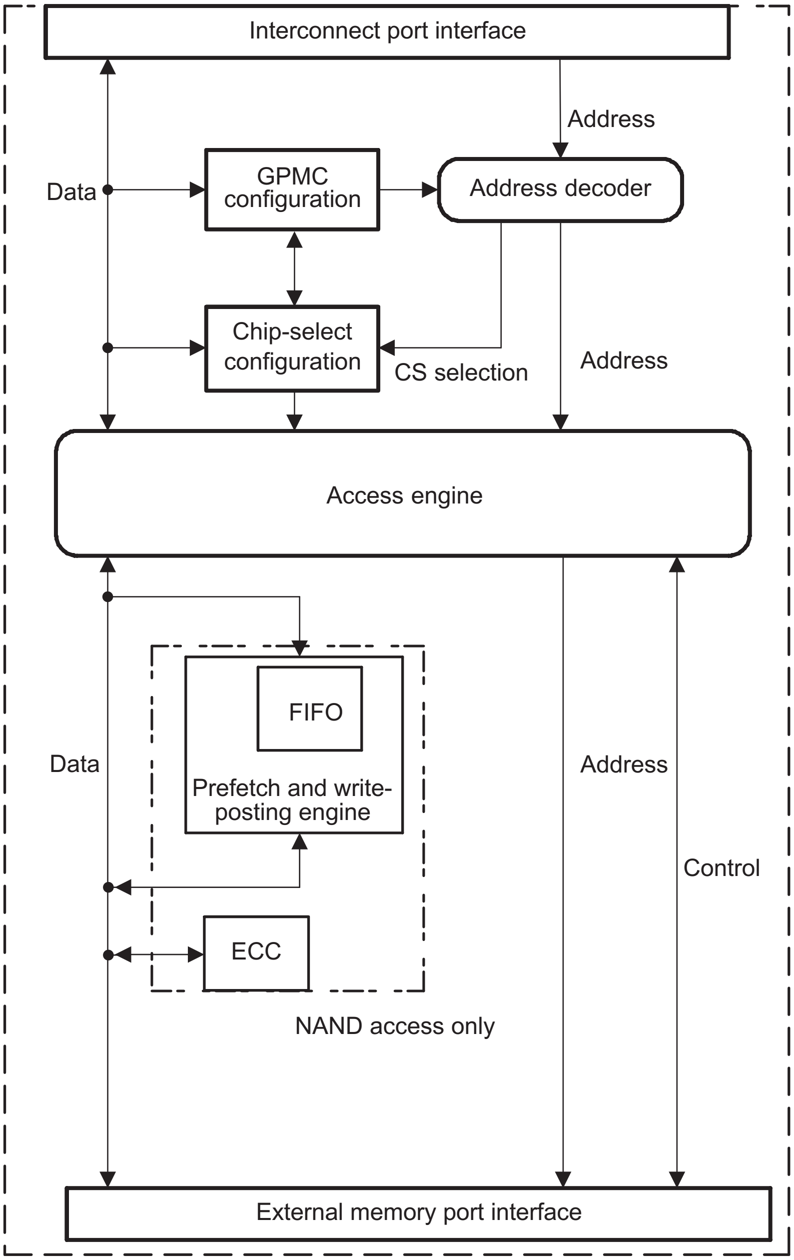
9.1.1.2 Block Diagram

The GPMC can access various external devices through the L3 Slow Interconnect. The flexible programming model allows a wide range of attached device types and access schemes. Based on the programmed configuration bit fields stored in the GPMC registers, the GPMC is able to generate all control signals timing depending on the attached device and access type. Given the chip-select decoding and its associated configuration registers, the GPMC selects the appropriate device type control signals timing.

Figure 9-1 shows the GPMC functional block diagram. The GPMC consists of six blocks:

- Interconnect port interface
- Address decoder, GPMC configuration, and chip-select configuration register file
- Access engine
- Prefetch and write-posting engine
- Error correction code engine (ECC)
- External device/memory port interface

Figure 9-1. GPMC Block Diagram



9.1.1.3 Unsupported GPMC Features

The following module features are not supported in this device.

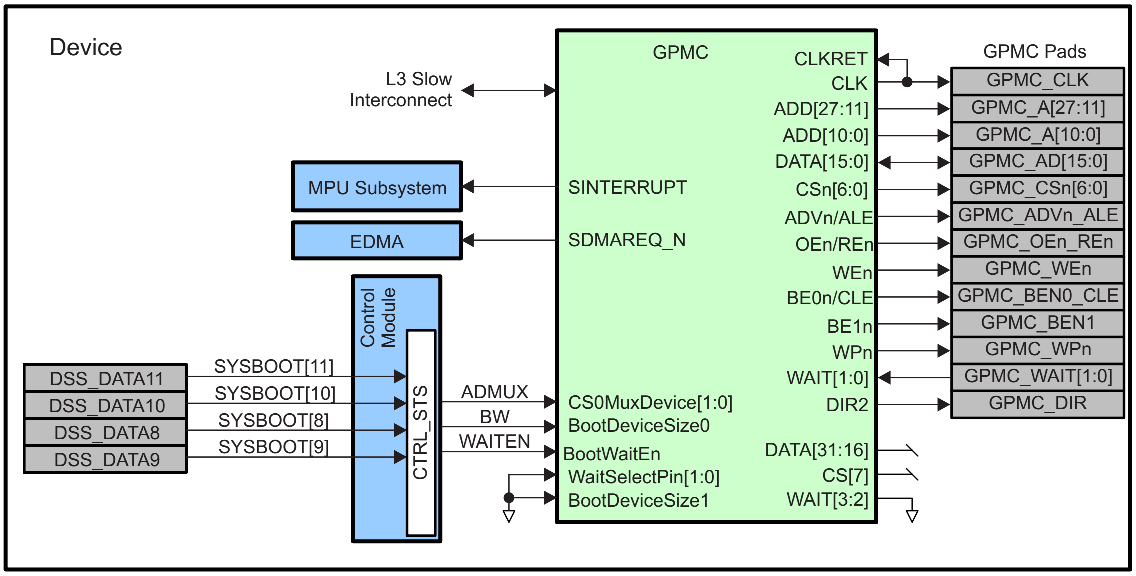
Table 9-1. Unsupported GPMC Features

Feature	Reason
Chip Select 7	Not pinned out
32-bit devices	Only 16 data lines pinned out
WAIT[3:2]	Not pinned out. All CS regions must use WAIT0 or WAIT1

9.1.2 Integration

An instantiation of GPMC provides this device with access to NAND Flash, NOR Flash, and other asynchronous and synchronous interface peripherals. Figure 9-2 shows the integration of the GPMC module in this device.

Figure 9-2. GPMC Integration



9.1.2.1 GPMC Connectivity Attributes

The general connectivity attributes for the GPMC module are shown in Table 9-2.

Table 9-2. GPMC Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L3S_GCLK
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	1 interrupt to MPU Subsystem (GPMCINT)
DMA Requests	1 DMA request to EDMA (GPMCEVT)
Physical Address	L3 Slow Slave Port Memory and control register regions qualified with MAddressSpace bit

9.1.2.2 GPMC Clock and Reset Management

The GPMC is a synchronous design and operates from the same clock as the Slow L3. All timings use this clock as a reference.

Table 9-3. GPMC Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
prcm_gpmc_clk Interface / Functional clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l3s_gclk From PRCM

9.1.2.3 GPMC Signal List

The GPMC external interface signals are shown in Table 9-4.

Table 9-4. GPMC Signal List

Signal	Type	Description
GPMC_A[27:0]	O	Address outputs
GPMC_AD[15:0]	I/O	Data[15:0] in non-muxed mode. A[16:1], D[15:0] in AD-muxed mode. A[27:17], A[16:1], D[15:0] in AAD-muxed mode.
GPMC_CSn[6:0]	O	Chip selects (active low)
GPMC_CLK	O(1)	Synchronous mode clock
GPMC_ADVn_ALE	O	Address Valid or Address Latch Enable depending if NOR or NAND protocol memories are selected.
GPMC_OEn_REn	O	Output Enable (active low). Also used as Read Enable (active low) for NAND protocol memories
GPMC_WEn	O	Write Enable (active low)
GPMC_BE0n_CLE	O	Lower Byte Enable (active low). Also used as Command Latch Enable for NAND protocol memories
GPMC_BE1n	O	Upper Byte Enable (active low)
GPMC_WPn	O	Write Protect (active low)
GPMC_WAIT[1:0]	I	External wait signal for NOR and NAND protocol memories.
GPMC_DIR	O	GPMC.D[15:0] signal direction control Low during transmit (for write access: data OUT from GPMC to memory) High during receive (for read access: data IN from memory to GPMC)

(1) These signals are also used as inputs to re-time or sync data. The associated CONF_<module>_pin_RXACTIVE bit for these signals must be set to 1 to enable the inputs back to the module. It is also recommended to place a 33-ohm resistor in series (close to the processor) on each of these signals to avoid signal reflections.

9.1.3 Functional Description

9.1.3.1 GPMC Signals

Table 9-5 shows the use of address and data GPMC controller pins based on the type of external device.

Table 9-5. GPMC Pin Multiplexing Options

GPMC Signal	Non Multiplexed Address Data 16-Bit Device(1)	Non Multiplexed Address Data 8-Bit Device	Multiplexed Address Data 16-Bit Device(1)	16-Bit NAND Device	8-Bit NAND Device
GPMC_A[27]	A26	A27	A26	Not Used	Not Used
GPMC_A[26]	A25	A26	Not Used	Not Used	Not Used
GPMC_A[25]	A24	A25	Not Used	Not Used	Not Used
GPMC_A[24]	A23	A24	Not Used	Not Used	Not Used
GPMC_A[23]	A22	A23	Not Used	Not Used	Not Used
GPMC_A[22]	A21	A22	Not Used	Not Used	Not Used
GPMC_A[21]	A20	A21	Not Used	Not Used	Not Used
GPMC_A[20]	A19	A20	Not Used	Not Used	Not Used
GPMC_A[19]	A18	A19	Not Used	Not Used	Not Used
GPMC_A[18]	A17	A18	Not Used	Not Used	Not Used
GPMC_A[17]	A16	A17	Not Used	Not Used	Not Used
GPMC_A[16]	A15	A16	Not Used	Not Used	Not Used
GPMC_A[15]	A14	A15	Not Used	Not Used	Not Used
GPMC_A[14]	A13	A14	Not Used	Not Used	Not Used
GPMC_A[13]	A12	A13	Not Used	Not Used	Not Used
GPMC_A[12]	A11	A12	Not Used	Not Used	Not Used
GPMC_A[11]	A10	A11	Not Used	Not Used	Not Used
GPMC_A[10]	A9	A10	A25	Not Used	Not Used
GPMC_A[9]	A8	A9	A24	Not Used	Not Used
GPMC_A[8]	A7	A8	A23	Not Used	Not Used
GPMC_A[7]	A6	A7	A22	Not Used	Not Used
GPMC_A[6]	A5	A6	A21	Not Used	Not Used
GPMC_A[5]	A4	A5	A20	Not Used	Not Used
GPMC_A[4]	A3	A4	A19	Not Used	Not Used
GPMC_A[3]	A2	A3	A18	Not Used	Not Used
GPMC_A[2]	A1	A2	A17	Not Used	Not Used
GPMC_A[1]	A0	A1	A16	Not Used	Not Used
GPMC_A[0]	Not Used	A0	Not Used	Not Used	Not Used
GPMC_AD[15]	D15	Not Used	A/D[15]	D15	Not Used
GPMC_AD[14]	D14	Not Used	A/D[14]	D14	Not Used
GPMC_AD[13]	D13	Not Used	A/D[13]	D13	Not Used
GPMC_AD[12]	D12	Not Used	A/D[12]	D12	Not Used
GPMC_AD[11]	D11	Not Used	A/D[11]	D11	Not Used
GPMC_AD[10]	D10	Not Used	A/D[10]	D10	Not Used
GPMC_AD[9]	D9	Not Used	A/D[9]	D9	Not Used
GPMC_AD[8]	D8	Not Used	A/D[8]	D8	Not Used
GPMC_AD[7]	D7	D7	A/D[7]	D7	D7
GPMC_AD[6]	D6	D6	A/D[6]	D6	D6

---

(1) The values in this column represent the signals on the memory. Be aware that some 16-bit memories may label the address lines differently. Some label the LSB as A0, while others use A1 for the LSB. These columns assume the LSB is A0.

Table 9-5. GPMC Pin Multiplexing Options (continued)

GPMC Signal	Non Multiplexed Address Data 16-Bit Device[1]	Non Multiplexed Address Data 8-Bit Device	Multiplexed Address Data 16-Bit Device[1]	16-Bit NAND Device	8-Bit NAND Device
GPMC_AD[5]	D5	D5	A/D[5]	D5	D5
GPMC_AD[4]	D4	D4	A/D[4]	D4	D4
GPMC_AD[3]	D3	D3	A/D[3]	D3	D3
GPMC_AD[2]	D2	D2	A/D[2]	D2	D2
GPMC_AD[1]	D1	D1	A/D[1]	D1	D1
GPMC_AD[0]	D0	D0	A/D[0]	D0	D0
GPMC_CS[0]n	CS0n (Chip Select)	CS0n (Chip Select)	CS0n (Chip Select)	CE0n (Chip Enable)	CE0n (Chip Enable)
GPMC_CS[1]n	CS1n	CS1n	CS1n	CE1n	CE1n
GPMC_CS[2]n	CS2n	CS2n	CS2n	CE2n	CE2n
GPMC_CS[3]n	CS3n	CS3n	CS3n	CE3n	CE3n
GPMC_CS[4]n	CS4n	CS4n	CS4n	CE4n	CE4n
GPMC_CS[5]n	CS5n	CS5n	CS5n	CE5n	CE5n
GPMC_CS[6]n	CS6n	CS6n	CS6n	CE6n	CE6n
GPMC_ADVn_ALE	ADVn (Address Value)	ADVn (Address Value)	ADVn (Address Value)	ALE (address latch enable)	ALE (address latch enable)
GPMC_BE0n_CLE	BE0n (Byte Enable)	BE0n (Byte Enable)	BE0n (Byte Enable)	CLE (command latch enable)	CLE (command latch enable)
GPMC_BE1n	BE1n	BE1n	BE1n		
GPMC_CLK	CLK	CLK	CLK		
GPMC_OE_REn	OEn (Output Enable)	OEn (Output Enable)	OEn (Output Enable)	REn (read enable)	REn (read enable)
GPMC_WAIT0	WAIT0	WAIT0	WAIT0	R/B0n (ready/busy)	R/B0n (ready/busy)
GPMC_WAIT1	WAIT1	WAIT1	WAIT1	R/B1n (ready/busy)	R/B1n (ready/busy)
GPMC_WEn	WEn (Write Enable)	WEn (Write Enable)	WEn (Write Enable)	WEn (write enable)	WEn (write enable)
GPMC_WPn	WPn (Write Protect)	WPn (Write Protect)	WPn (Write Protect)	WPn (write protect)	WPn (write protect)

With all device types, the GPMC does not drive unnecessary address lines. They stay at their reset value of 00.

Address mapping supports address/data-multiplexed 16-bit wide devices:

- The NOR flash memory controller still supports non-multiplexed address and data memory devices.
- Multiplexing mode can be selected through the GPMC_CONFIG1_i[9-8] MUXADDDATA bit field.
- Asynchronous page mode is not supported for multiplexed address and data devices.

9.1.3.2 GPMC Modes

This section shows three GPMC external connections options:

- Figure 9-3 shows a connection between the GPMC and a 16-bit synchronous address/data-multiplexed (or AAD-multiplexed, but this protocol use less address pins) external memory device.
- Figure 9-4 shows a connection between the GPMC and a 16-bit synchronous nonmultiplexed external memory device .
- Figure 9-5 shows a connection between the GPMC and a 8-bit NAND device

Figure 9-3. GPMC to 16-Bit Address/Data-Multiplexed Memory

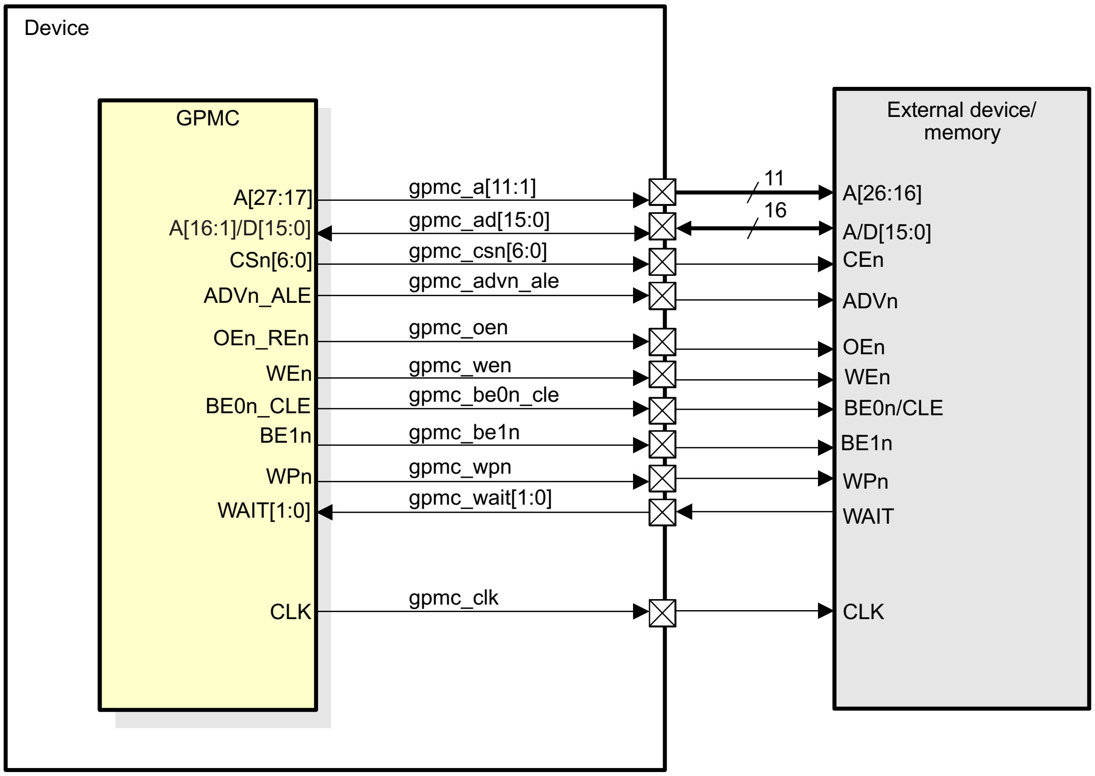


Figure 9-4. GPMC to 16-Bit Non-multiplexed Memory

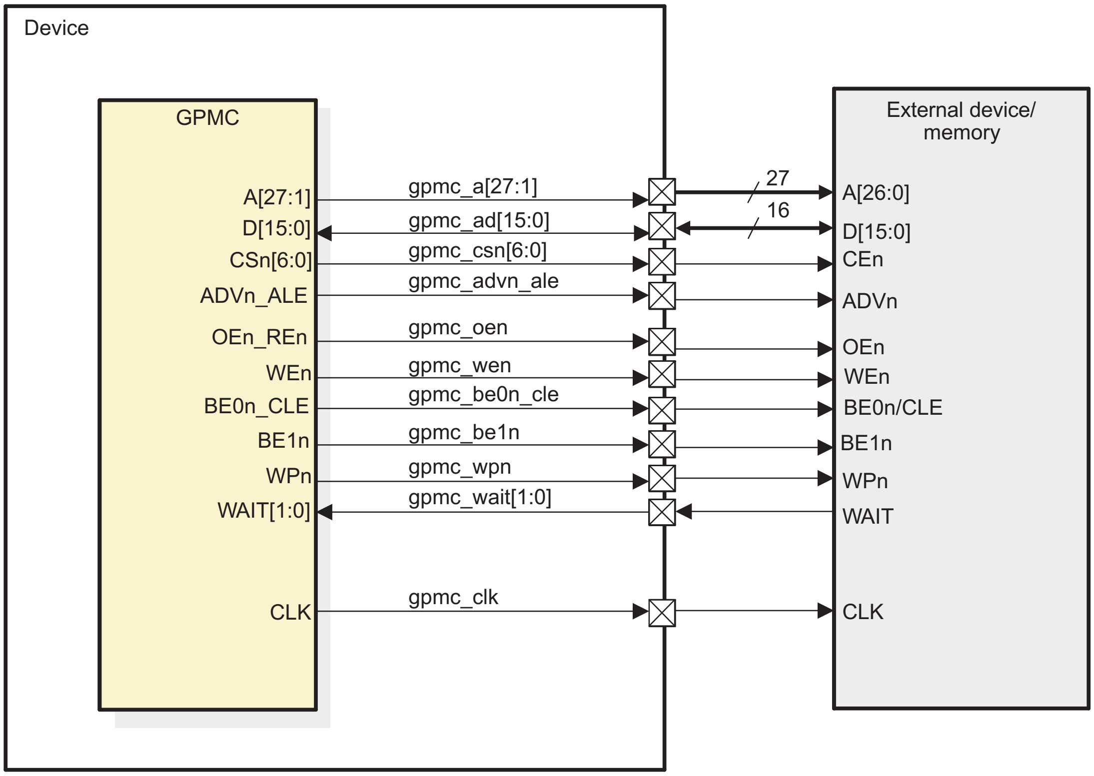
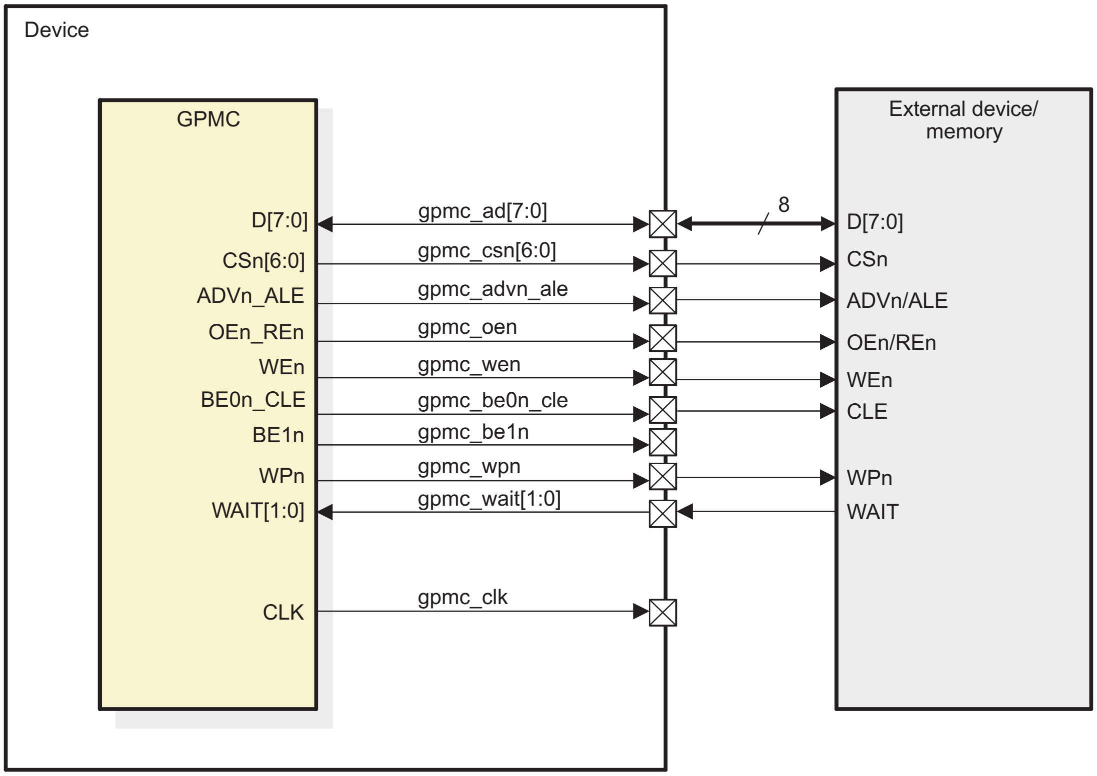


Figure 9-5. GPMC to 8-Bit NAND Device



9.1.3.3 GPMC Functional Description

The GPMC basic programming model offers maximum flexibility to support various access protocols for each of the configurable chip-selects. Use optimal chip-select settings, based on the characteristics of the external device:

- Different protocols can be selected to support generic asynchronous or synchronous random-access devices (NOR flash, SRAM) or to support specific NAND devices.
- The address and the data bus can be multiplexed on the same external bus.
- Read and write access can be independently defined as asynchronous or synchronous.
- System requests (byte, 16-bit word, burst) are performed through single or multiple accesses. External access profiles (single, multiple with optimized burst length, native- or emulated-wrap) are based on external device characteristics (supported protocol, bus width, data buffer size, native-wrap support).
- System burst read or write requests are synchronous-burst (multiple-read or multiple-write). When neither burst nor page mode is supported by external memory or ASIC devices, system burst read or write requests are translated to successive single synchronous or asynchronous accesses (single reads or single writes). 8-bit wide devices are supported only in single synchronous or single asynchronous read or write mode.
- To simulate a programmable internal-wait state, an external wait pin can be monitored to dynamically control external access at the beginning (initial access time) of and during a burst access.

Each control signal is controlled independently for each chip-select. The internal functional clock of the GPMC (GPMC_FCLK) is used as a time reference to specify the following:

- Read- and write-access duration
- Most GPMC external interface control-signal assertion and deassertion times
- Data-capture time during read access
- External wait-pin monitoring time
- Duration of idle time between accesses, when required

9.1.3.3.1 GPMC Clock Configuration

Table 9-6 describes the GPMC clocks.

Table 9-6. GPMC Clocks

Signal	I/O	Description
GPMC_FCLK	I	Functional and interface clock
GPMC_CLK	O	External clock provided to synchronous external memory devices.

The GPMC_CLK is generated by the GPMC from the internal GPMC_FCLK clock. The source of the GPMC_FCLK is described in . The GPMC_CLK is configured via the GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER field (for i = 0 to 3) as shown in Table 9-7.

Table 9-7. GPMC_CONFIG1_i Configuration

Source Clock	GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER	GPMC_CLK Generated Clock Provided to External Memory Device
GPMC_FCLK	00	GPMC_FCLK
	01	GPMC_FCLK/2
	10	GPMC_FCLK/3
	11	GPMC_FCLK/4

9.1.3.3.2 GPMC Software Reset

The GPMC can be reset by software through the GPMC_SYSCONFIG[1] SOFTRESET bit. Setting the bit to 1 enables an active software reset that is functionally equivalent to a hardware reset. Hardware and software resets initialize all GPMC registers and the finite state-machine (FSM) immediately and unconditionally. The GPMC_SYSSTS[0] RESETDONE bit indicates that the software reset is complete when its value is 1. The software must ensure that the software reset completes before doing GPMC operations.

9.1.3.3.3 GPMC Power Management

GPMC power is supplied by the CORE power domain, and GPMC power management complies with system power-management guidelines. Table 9-8 describes power-management features available for the GPMC module.

Table 9-8. GPMC Local Power Management Features

Feature	Registers	Description
Clock Auto Gating	GPMC_SYSCONFIG[0] AUTOIDLE] bit	This bit allows a local power optimization inside the module, by gating the GPMC_FCLK clock upon the internal activity.
Slave Idle Modes	GPMC_SYSCONFIG[4-3] SIDLEMODE bit field	Force-idle, No-idle and Smart-idle wakeup modes are available
Clock Activity	N/A	Feature not available
Master Standby Modes	N/A	Feature not available
Global Wake-up Enable	N/A	Feature not available
Wake-up Sources Enable	N/A	Feature not available

9.1.3.3.4 GPMC Interrupt Requests

The GPMC generates one interrupt event as shown in Figure 9-2.

- The interrupt request goes from GPMC (GPMC_IRQ) to the MPU subsystem: A_IRQ_132

Table 9-9 lists the event flags, and their mask, that can cause module interrupts.

Table 9-9. GPMC Interrupt Events

Event Flag	Event Mask	Sensitivity	Map to	Description
GPMC_IRQSTS[9] WAIT1EDGEDETECTIONSTS	GPMC_IRQEN[9] WAIT1EDGEDETECTIONEN	Edge	A_IRQ_132	Wait1 edge detection interrupt: Triggered if a rising or falling edge is detected on the GPMC_WAIT1 signal. The rising or falling edge detection of Wait1 is selected through GPMC_CONFIG[9] WAIT1PINPOLARITY bit.
GPMC_IRQSTS[8] WAIT0EDGEDETECTIONSTS	GPMC_IRQEN[8] WAIT0EDGEDETECTIONEN	Edge	A_IRQ_132	Wait0 edge detection interrupt: Triggered if a rising or falling edge is detected on the GPMC_WAIT0 signal. The rising or falling edge detection of Wait0 is selected through GPMC_CONFIG[8] WAIT0PINPOLARITY bit.
GPMC_IRQSTS[1] TERMINALCOUNTSTS	GPMC_IRQEN[1] TERMINALCOUNTENABLE	Level	A_IRQ_132	Terminal count event: Triggered on prefetch process completion, that is when the number of currently remaining data to be requested reaches 0.
GPMC_IRQSTS[0] FIFOEVTSTS	GPMC_IRQEN[0] FIFOEVTEN	Level	A_IRQ_132	FIFO event interrupt: Indicates FIFO levels availability for in Write-Posting mode and prefetch mode. GPMC_PREFETCH_CONFIG[2] DMAMODE bit shall be cleared to 0.

9.1.3.3.5 GPMC DMA Requests

The GPMC generates one DMA event, from GPMC (GPMC_DMA_REQ) to the eDMA: e_DMA_52.

9.1.3.3.6 L3 Slow Interconnect Interface

The GPMC L3 Slow interconnect interface is a pipelined interface including an 16 × 32-bit word write buffer. Any system host can issue external access requests through the GPMC. The device system can issue the following requests through this interface:

- One 8-bit / 16-bit / 32-bit interconnect access (read/write)
- Two incrementing 32-bit interconnect accesses (read/write)
- Two wrapped 32-bit interconnect accesses (read/write)
- Four incrementing 32-bit interconnect accesses (read/write)
- Four wrapped 32-bit interconnect accesses (read/write)
- Eight incrementing 32-bit interconnect accesses (read/write)
- Eight wrapped 32-bit interconnect accesses (read/write)

Only linear burst transactions are supported; interleaved burst transactions are not supported. Only power-of-two-length precise bursts 2 × 32, 4 × 32, 8 × 32 or 16 × 32 with the burst base address aligned on the total burst size are supported (this limitation applies to incrementing bursts only).

This interface also provides one interrupt and one DMA request line, for specific event control.

It is recommended to program the GPMC_CONFIG1_i ATTACHEDDEVICEPAGELENGTH field ([24-23]) according to the effective attached device page length and to enable the GPMC_CONFIG1_i WRAPBURST bit ([31]) if the attached device supports wrapping burst. However, it is possible to emulate wrapping burst on a non-wrapping memory by providing relevant addresses within the page or splitting transactions. Bursts larger than the memory page length are chopped into multiple bursts transactions. Due to the alignment requirements, a page boundary is never crossed.

9.1.3.3.7 GPMC Address and Data Bus

The current application supports GPMC connection to NAND devices and to address/data-multiplexed memories or devices. Connection to address/data-nonmultiplexed memories Depending on the GPMC configuration of each chip-select, address and data bus lines that are not required for a particular access protocol are not updated (changed from current value) and are not sampled when input (input data bus).

- For address/data-multiplexed and AAD-multiplexed NOR devices, the address is multiplexed on the data bus.
- 8-bit wide NOR devices do not use GPMC I/O: GPMC_AD[15-8] for data (they are used for address if needed).
- 16-bit wide NAND devices do not use GPMC I/O: GPMC_A[27-0].
- 8-bit wide NAND devices do not use GPMC I/O: GPMC_A[27-0] and GPMC I/O: GPMC_AD[15-8].

9.1.3.3.7.1 GPMC I/O Configuration Setting

NOTE: In this section and next sections, the i in GPMC_CONFIGx_i stands for the GPMC chip-select i where i = 0 to 6.

To select a NAND device, program the following register fields:

- GPMC_CONFIG1_i[11-10] DEVICETYPE field = 10b
- GPMC_CONFIG1_i[9-8] MUXADDDATA bit = 00

To select an address/data-multiplexed device, program the following register fields:

- GPMC_CONFIG1_i[11-10] DEVICETYPE field = 00
- GPMC_CONFIG1_i[9-8] MUXADDDATA bit = 10b

To select an address/address/data-multiplexed device, program the following register fields:

- GPMC_CONFIG1_i[11-10] DEVICETYPE field = 00
- GPMC_CONFIG1_i[9-8] MUXADDDATA bit = 01b

To select an address/data-nonmultiplexed device , program the following register fields:

- GPMC_CONFIG1_i[11-10] DEVICETYPE field = 00
- GPMC_CONFIG1_i[9-8] MUXADDDATA bit = 00

9.1.3.3.8 Address Decoder and Chip-Select Configuration

Addresses are decoded accordingly with the address request of the chip-select and the content of the chip-select base address register file, which includes a set of global GPMC configuration registers and eight sets of chip-select configuration registers.

The GPMC configuration register file is memory-mapped and can be read or written with byte, 16-bit word, or 32-bit word accesses. The register file should be configured as a noncacheable, nonbufferable region to prevent any desynchronization between host execution (write request) and the completion of register configuration (write completed with register updated). Section 9.1.6 provides the GPMC register locations. For the map of GPMC memory locations, see .

After the chip-select is configured, the access engine accesses the external device, drives the external interface control signals, and applies the interface protocol based on user-defined timing parameters and settings.

9.1.3.3.8.1 Chip-Select Base Address and Region Size

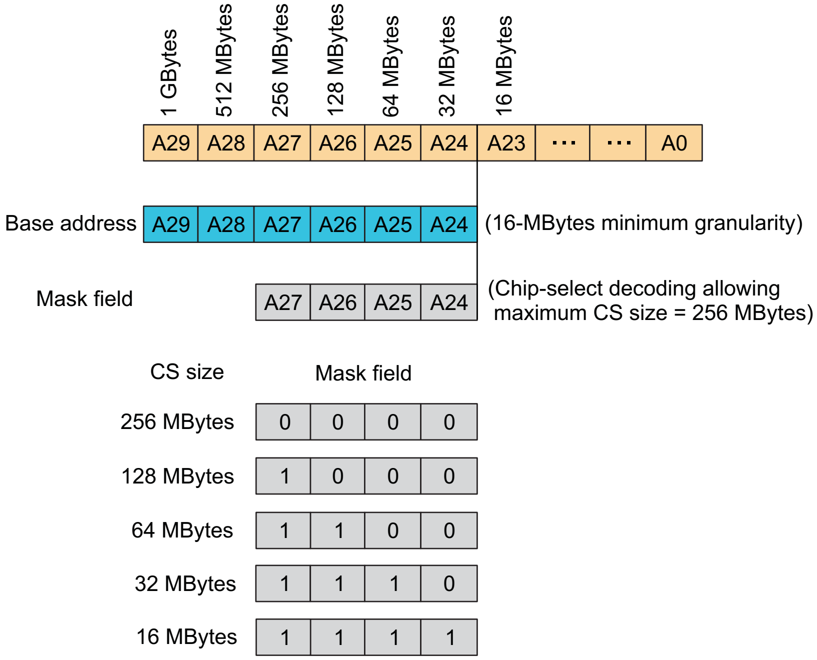
Any external memory or ASIC device attached to the GPMC external interface can be accessed by any device system host within the GPMC 512-Mbyte contiguous address space. For details, see .

The GPMC 512 Mbyte address space can be divided into a maximum of seven chip-select regions with programmable base address and programmable CS size. The CS size is programmable from 16 to 256MB (must be a power-of-2) and is defined by the mask field. Attached memory smaller than the programmed CS region size is accessed through the entire CS region (aliasing).

Each chip-select has a 6-bit base address encoding and a 4-bit decoding mask, which must be programmed according to the following rules:

- The programmed chip-select region base address must be aligned on the chip-select region size address boundary and is limited to a power-of-2 address value. During access decoding, the register base address value is used for address comparison with the address-bit line mapping as described in Figure 9-6 (with A0 as the device system byte-address line). Base address is programmed through the GPMC_CONFIG7_i[5-0] BASEADDRESS bit field.
- The register mask is used to exclude some address lines from the decoding. A register mask bit field cleared to 0 suppresses the associated address line from the address comparison (incoming address bit line is don't care). The register mask value must be limited to the subsequent value, based on the desired chip-select region size. Any other value has an undefined result. When multiple chip-select regions with overlapping addresses are enabled concurrently, access to these chip-select regions is cancelled and a GPMC access error is posted. The mask field is programmed through the GPMC_CONFIG7_i[11-8] MASKADDRESS bit field.

Figure 9-6. Chip-Select Address Mapping and Decoding Mask



A mask value of 0010 or 1001 must be avoided because it will create holes in the chip-select address space.

Chip-select configuration (base and mask address or any protocol and timing settings) must be performed while the associated chip-select is disabled through the GPMC_CONFIG7_i[6] CSVALID bit. In addition, a chip-select configuration can only be disabled if there is no ongoing access to that chip-select. This requires activity monitoring of the prefetch or write-posting engine if the engine is active on the chip-select. Also, the write buffer state must be monitored to wait for any posted write completion to the chip-select.

Any access attempted to a nonvalid GPMC address region (CSVALID disabled or address decoding outside a valid chip-select region) is not propagated to the external interface and a GPMC access error is posted. In case of chip-selects overlapping, an error is generated and no access will occur on either chip-select. Chip-select 0 is the only chip-select region enabled after either a power-up or a GPMC reset.

Although the GPMC interface can drive up to seven chip-selects, the frequency specified for this interface is for a specific load. If this load is exceeded, the maximum frequency cannot be reached. One solution is to implement a board with buffers, to allow the slowest device to maintain the total load on the lines.

9.1.3.3.8.2 Access Protocol

9.1.3.3.8.2.1 Supported Devices

The access protocol of each chip-select can be independently specified through the GPMC_CONFIG1_i[11-10] DEVICETYPE parameter for:

- Random-access synchronous or asynchronous memory like NOR flash, SRAM
- NAND flash asynchronous devices

For more information about the NAND flash GPMC basic programming model and NAND support, see Section 9.1.3.3.12 and Section 9.1.3.3.12.1.

9.1.3.3.8.2.2 Access Size Adaptation and Device Width

Each chip-select can be independently configured through the GPMC_CONFIG1_i[13-12] DEVICESIZE field to interface with a 16-bit wide device or an 8-bit wide device. System requests with data width greater than the external device data bus width are split into successive accesses according to both the external device data-bus width and little-endian data organization.

An 8-bit wide device must be interfaced to the D0-D7 external interface bus lane. GPMC data accesses only use this bus lane when the associated chip-select is attached to an 8-bit wide device.

The 8-bit wide device can be interfaced in asynchronous or synchronous mode in single data phase (no 8-bit wide device burst mode). If the 8-bit wide device is set in the chip-select configuration register, ReadMultiple and WriteMultiple bit fields are considered “don’t care” and only single accesses are performed.

A 16-bit wide device can be interfaced in asynchronous or synchronous mode, with single or multiple data phases for an access, and with native or emulated wrap mode support.

9.1.3.3.8.2.3 Address/Data-Multiplexing Interface

For random synchronous or asynchronous memory interfacing (DEVICETYPE = 0b00), an address- and data-multiplexing protocol can be selected through the GPMC_CONFIG1_i[[9-8] MUXADDDATA bit field. The ADVn signal must be used as the external device address latch control signal. For the associated chip-select configuration, ADVn assertion and deassertion time and OEn assertion time must be set to the appropriate value to meet the address latch setup/hold time requirements of the external device (see Section 9.1.2).

This address/data-multiplexing interface is not applicable to NAND device interfacing. NAND devices require a specific address, command, and data multiplexing protocol (see Section 9.1.3.3.12).

9.1.3.3.8.3 External Signals

9.1.3.3.8.3.1 WAIT Pin Monitoring Control

GPMC access time can be dynamically controlled using an external gpmc_wait pin when the external device access time is not deterministic and cannot be defined and controlled only using the GPMC internal RDACCESSTIME, WRACCESSTIME and PAGEBURSTACCESSTIME wait state generator.

The GPMC features two input wait pin:gpmc_wait1, and gpmc_wait0. This pin allow control of external devices with different wait-pin polarity. They also allow the overlap of wait-pin assertion from different devices without affecting access to devices for which the wait pin is not asserted.

- The GPMC_CONFIG1_i[17-16] WAITPINSELECT bit field (where i = 0 to 6) selects which input gpmc_wait pin is used for the device attached to the corresponding chip-select.
- The polarity of the wait pin is defined through the WAITxPINPOLARITY bit of the GPMC_CONFIG register. A wait pin configured to be active low means that low level on the WAIT signal indicates that the data is not ready and that the data bus is invalid. When WAIT is inactive, data is valid.

The GPMC access engine can be configured per CS to monitor the wait pin of the external memory device or not, based on the access type: read or write.

- The GPMC_CONFIG1_i[22] WAITREADMONITORING bit defines whether the wait pin should be monitored during read accesses or not.
- The GPMC_CONFIG1_i[21] WAITWRITEMONITORING bit defines whether the wait pin should be monitored during write accesses or not.

The GPMC access engine can be configured to monitor the wait pin of the external memory device asynchronously or synchronously with the GPMC_CLK clock, depending on the access type: synchronous or asynchronous (the GPMC_CONFIG1_i[29] READTYPE and GPMC_CONFIG1_i[27] WRITETYPE bits).

9.1.3.3.8.3.2 Wait Monitoring During an Asynchronous Read Access

When wait-pin monitoring is enabled for read accesses (WAITREADMONITORING), the effective access time is a logical AND combination of the RDACCESSTIME timing completion and the wait-deasserted state.

During asynchronous read accesses with wait-pin monitoring enabled, the wait pin must be at a valid level (asserted or deasserted) for at least two GPMC clock cycles before RDACCESSTIME completes, to ensure correct dynamic access-time control through wait-pin monitoring. The advance pipelining of the two GPMC clock cycles is the result of the internal synchronization requirements for the WAIT signal.

In this context, RDACCESSTIME is used as a WAIT invalid timing window and is set to such a value that the wait pin is at a valid state two GPMC clock cycles before RDACCESSTIME completes.

Similarly, during a multiple-access cycle (for example, asynchronous read page mode), the effective access time is a logical AND combination of PAGEBURSTACCESSTIME timing completion and the wait-deasserted state. Wait-monitoring pipelining is also applicable to multiple accesses (access within a page).

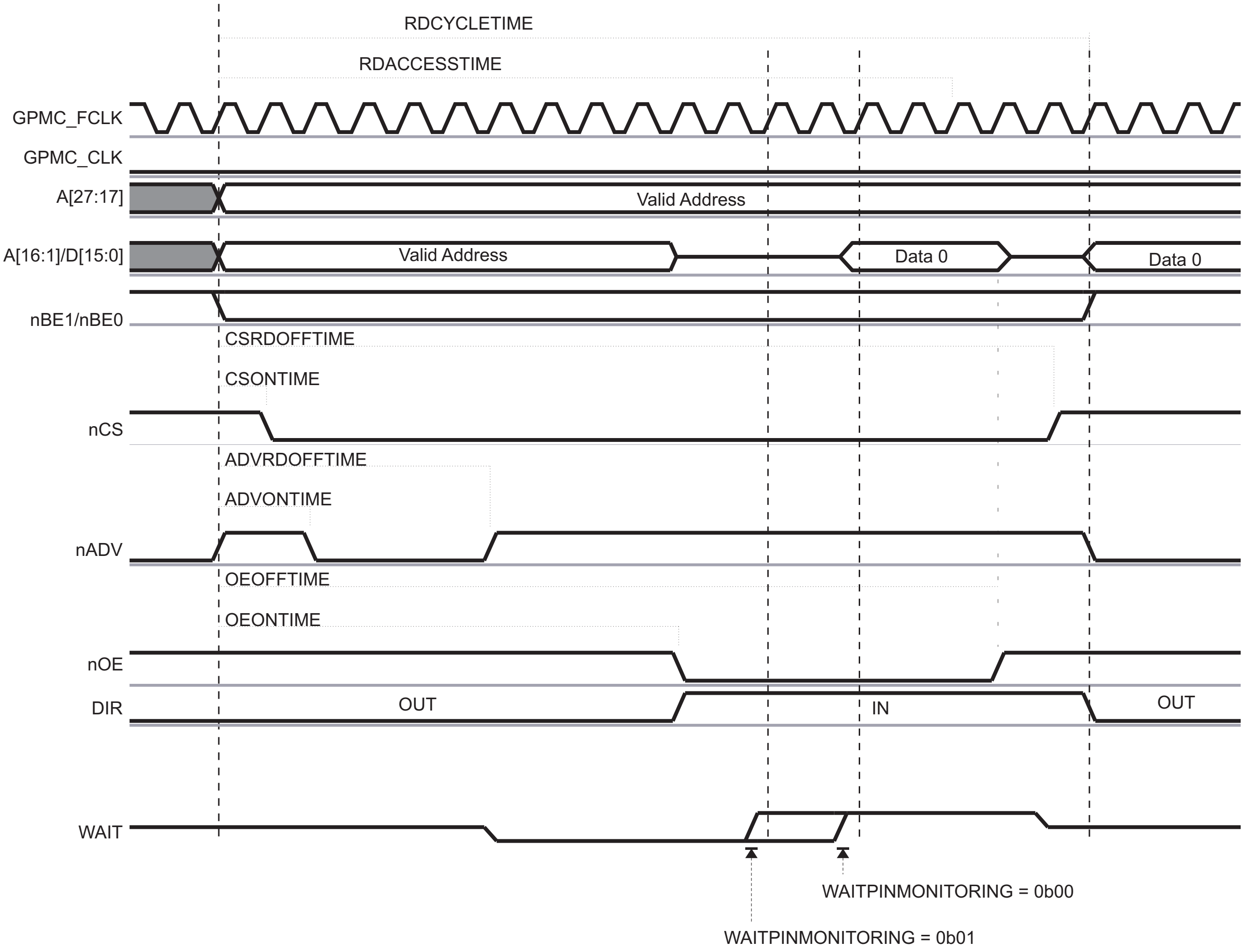
- WAIT monitored as active freezes the CYCLETIME counter. For an access within a page, when the CYCLETIME counter is by definition in a lock state, WAIT monitored as asserted extends the current access time in the page. Control signals are kept in their current state. The data bus is considered invalid, and no data are captured during this clock cycle.
- WAIT monitored as inactive unfreezes the CYCLETIME counter. For an access within a page, when the CYCLETIME counter is by definition in a lock state, WAIT monitored as inactive completes the current access time and starts the next access phase in the page. The data bus is considered valid, and data are captured during this clock cycle. In case of a single access or if this was the last access in a multiple-access cycle, all signals are controlled according to their related control timing value and according to the CYCLETIME counter status.

When a delay larger than two GPMC clocks must be observed between wait-pin deactivation time and data valid time (including the required GPMC and the device data setup time), an extra delay can be added between wait-pin deassertion time detection and effective data-capture time and the effective unlock of the CYCLETIME counter. This extra delay can be programmed in the GPMC_CONFIG1_i[19-18] WAITMONITORINGTIME field.

- The WAITMONITORINGTIME parameter does not delay the wait-pin active or inactive detection, nor does it modify the two GPMC clocks pipelined detection delay.
- This extra delay is expressed as a number of GPMC_CLK clock cycles, even though the access is defined as asynchronous, and no GPMC_CLK clock is provided to the external device. Still, GPMCFCLKDIVIDER is used as a divider for the GPMC clock, so it must be programmed to define the correct WAITMONITORINGTIME delay.

Figure 9-7 shows wait behavior during an asynchronous single read access.

Figure 9-7. Wait Behavior During an Asynchronous Single Read Access (GPMCFCLKDivider = 1)



The WAIT signal is active low. GPMC_CONFIG1_i[19-18] WAITMONITORINGTIME = 00b or 01b.

9.1.3.3.8.3.3 Wait Monitoring During an Asynchronous Write Access

When wait-pin monitoring is enabled for write accesses (GPMC_CONFIG1_i[21] WAITWRITEMONITORING bit = 1), the WAIT-invalid timing window is defined by the WRACCESSTIME field. WRACCESSTIME must be set so that the wait pin is at a valid state two GPMC clock cycles before WRACCESSTIME completes. The advance pipelining of the two GPMC clock cycles is the result of the internal synchronization requirements for the WAIT signal.

- WAIT monitored as active freezes the CYCLETIME counter. This informs the GPMC that the data bus is not captured by the external device. The control signals are kept in their current state. The data bus still drives the data.
- WAIT monitored as inactive unfreezes the CYCLETIME counter. This informs that the data bus is correctly captured by the external device. All signals, including the data bus, are controlled according to their related control timing value and to the CYCLETIME counter status.

When a delay larger than two GPMC clock cycles must be observed between wait-pin deassertion time and the effective data write into the external device (including the required GPMC data setup time and the device data setup time), an extra delay can be added between wait-pin deassertion time detection and effective data write time into the external device and the effective unfreezing of the CYCLETIME counter. This extra delay can be programmed in the GPMC_CONFIG1_i[19-18] WAITMONITORINGTIME fields.

- The WAITMONITORINGTIME parameter does not delay the wait-pin assertion or deassertion detection, nor does it modify the two GPMC clock cycles pipelined detection delay.
- This extra delay is expressed as a number of GPMC_CLK clock cycles, even though the access is defined as asynchronous, and even though no clock is provided to the external device. Still, GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER is used as a divider for the GPMC clock and so it must be programmed to define the correct WAITMONITORINGTIME delay.

9.1.3.3.8.3.4 Wait Monitoring During a Synchronous Read Access

During synchronous accesses with wait-pin monitoring enabled, the wait pin is captured synchronously with GPMC_CLK, using the rising edge of this clock.

The WAIT signal can be programmed to apply to the same clock cycle it is captured in. Alternatively, it can be sampled one or two GPMC_CLK cycles ahead of the clock cycle it applies to. This pipelining is applicable to the entire burst access, and to all data phase in the burst access. This WAIT pipelining depth is programmed in the GPMC_CONFIG1_i[19-18] WAITMONITORINGTIME field, and is expressed as a number of GPMC_CLK clock cycles.

In synchronous mode, when wait-pin monitoring is enabled (GPMC_CONFIG1_i[22] WAITREADMONITORING bit), the effective access time is a logical AND combination of the RDACCESSTIME timing completion and the WAIT deasserted-state detection.

Depending on the programmed WAITMONITORINGTIME value, the wait pin should be at a valid level, either asserted or deasserted:

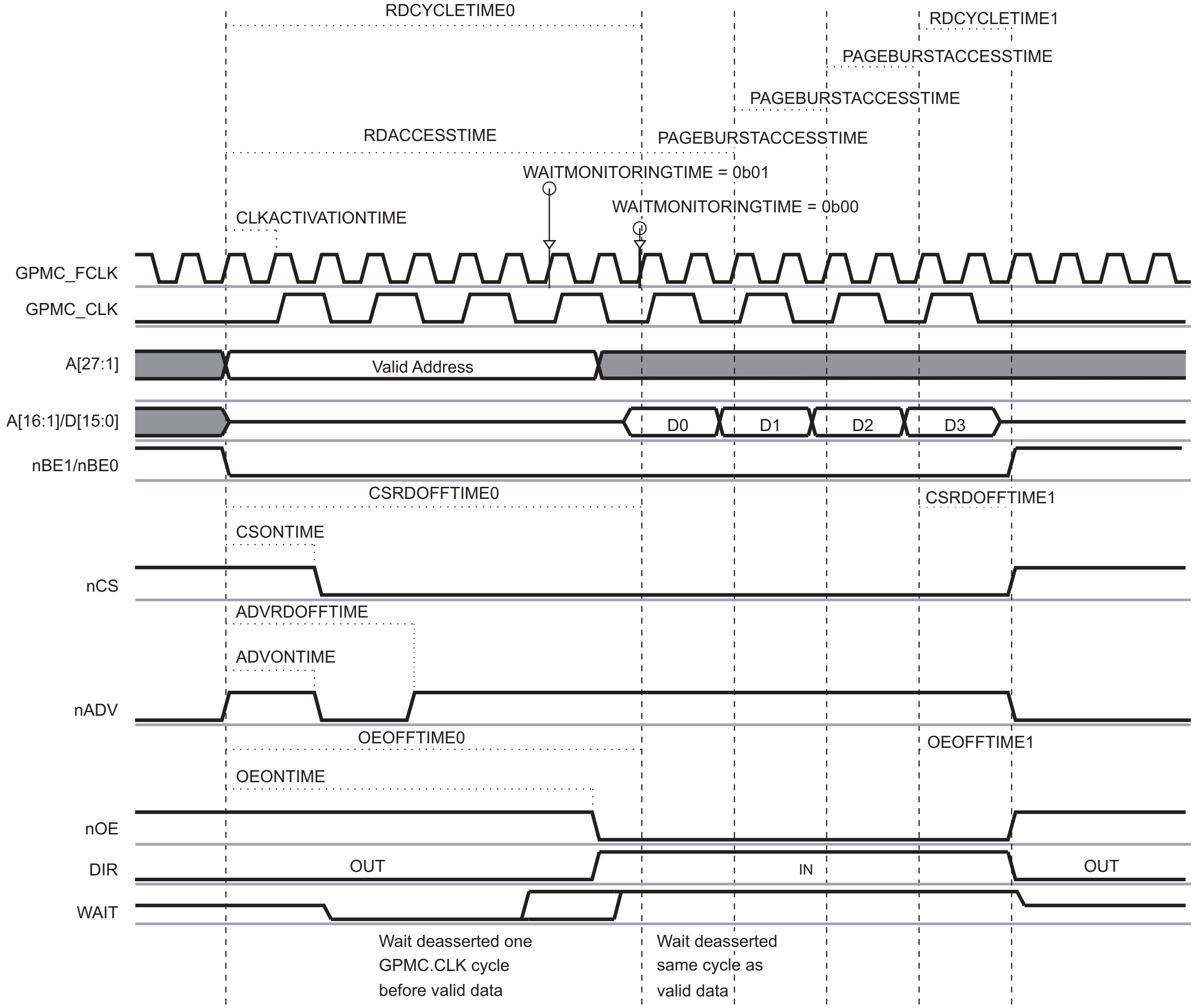
- In the same clock cycle the data is valid if WAITMONITORINGTIME = 0 (at RDACCESSTIME completion)
- In the WAITMONITORINGTIME x (GPMCFCLKDIVIDER + 1) GPMC_FCLK clock cycles before RDACCESSTIME completion if WAITMONITORINGTIME not equal to 0

Similarly, during a multiple-access cycle (burst mode), the effective access time is a logical AND combination of PAGEBURSTACCESSTIME timing completion and the wait-inactive state. The Wait pipelining depth programming applies to the whole burst access.

- WAIT monitored as active freezes the CYCLETIME counter. For an access within a burst (when the CYCLETIME counter is by definition in a lock state), WAIT monitored as active extends the current access time in the burst. Control signals are kept in their current state. The data bus is considered invalid, and no data are captured during this clock cycle.
- WAIT monitored as inactive unfreezes the CYCLETIME counter. For an access within a burst (when the CYCLETIME counter is by definition in lock state), WAIT monitored as inactive completes the current access time and starts the next access phase in the burst. The data bus is considered valid, and data are captured during this clock cycle. In a single access or if this was the last access in a multiple-access cycle, all signals are controlled according to their relative control timing value and the CYCLETIME counter status.

Figure 9-8 shows wait behavior during a synchronous read burst access.

Figure 9-8. Wait Behavior During a Synchronous Read Burst Access



The WAIT signal is active low. WAITMONITORINGTIME = 00b or 01b.

9.1.3.3.8.3.5 Wait Monitoring During a Synchronous Write Access

During synchronous accesses with wait-pin monitoring enabled (the WAITWRITEMONITORING bit), the wait pin is captured synchronously with GPMC_CLK, using the rising edge of this clock.

If enabled, external wait-pin monitoring can be used in combination with WRACCESSTIME to delay the effective memory device GPMC_CLK capture edge.

Wait-monitoring pipelining depth is similar to synchronous read access:

- At WRACCESSTIME completion if WAITMONITORINGTIME = 0
- In the WAITMONITORINGTIME x (GPMCFCLKDIVIDER + 1) GPMC_FCLK cycles before WRACCESSTIME completion if WAITMONITORINGTIME not equal to 0.

Wait-monitoring pipelining definition applies to whole burst accesses:

- WAIT monitored as active freezes the CYCLETIME counter. For accesses within a burst, when the CYCLETIME counter is by definition in a lock state, WAIT monitored as active indicates that the data bus is not being captured by the external device. Control signals are kept in their current state. The data bus is kept in its current state.
- WAIT monitored as inactive unfreezes the CYCLETIME counter. For accesses within a burst, when the CYCLETIME counter is by definition in a lock state, WAIT monitored as inactive indicates the effective data capture of the bus by the external device and starts the next access of the burst. In case of a single access or if this was the last access in a multiple access cycle, all signals, including the data bus, are controlled according to their related control timing value and the CYCLETIME counter status.

Wait monitoring is supported for all configurations except for GPMC_CONFIG1_i[19-18] WAITMONITORINGTIME = 0 for write bursts with a clock divider of 1 or 2 (GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER field equal to 0 or 1, respectively).

9.1.3.3.8.3.6 WAIT With NAND Device

For details about the use of the wait pin for communication with a NAND flash external device, see Section 9.1.3.3.12.2.

9.1.3.3.8.3.7 Idle Cycle Control Between Successive Accesses

9.1.3.3.8.3.7.1 Bus Turnaround (BUSTURNAROUND)

To prevent data-bus contention, an access that follows a read access to a slow memory/device must be delayed (in other words, control the CSn/OEn de-assertion to data bus in high-impedance delay).

The bus turnaround is a time-out counter starting after CSn or OEn de-assertion time, whichever occurs first, and delays the next access start-cycle time. The counter is programmed through the GPMC_CONFIG6_i[3-0] BUSTURNAROUND bit field.

After a read access to a chip-select with a non zero BUSTURNAROUND, the next access is delayed until the BUSTURNAROUND delay completes, if the next access is one of the following:

- A write access to any chip-select (same or different from the chip-select data was read from)
- A read access to a different chip-select from the chip-select data was read access from
- A read or write access to a chip-select associated with an address/data-multiplexed device

Bus keeping starts after bus turnaround completion so that DIR changes from IN to OUT after bus turnaround. The bus will not have enough time to go into high-impedance even though it could be driven with the same value before bus turnaround timing.

BUSTURNAROUND delay runs in parallel with GPMC_CONFIG6_i[3-0] CYCLE2CYCLEDELAY delays. It should be noted that BUSTURNAROUND is a timing parameter for the ending chip-select access while CYCLE2CYCLEDELAY is a timing parameter for the following chip-select access. The effective minimum delay between successive accesses is driven by these delay timing parameters and by the access type of the following access. See Figure 9-9 to Figure 9-11.

Another way to prevent bus contention is to define an earlier CSn or OEn deassertion time for slow devices or to extend the value of RDCYCLETIME. Doing this prevents bus contention, but affects all accesses of this specific chip-select.

Figure 9-9. Read to Read for an Address-Data Multiplexed Device, On Different CS, Without Bus Turnaround (CS0n Attached to Fast Device)

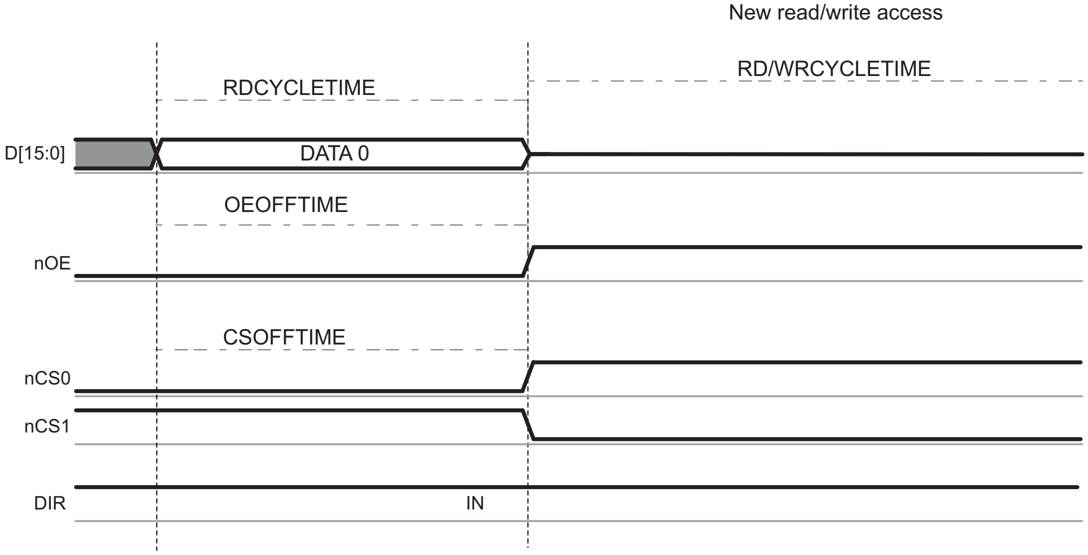


Figure 9-10. Read to Read / Write for an Address-Data Multiplexed Device, On Different CS, With Bus Turnaround

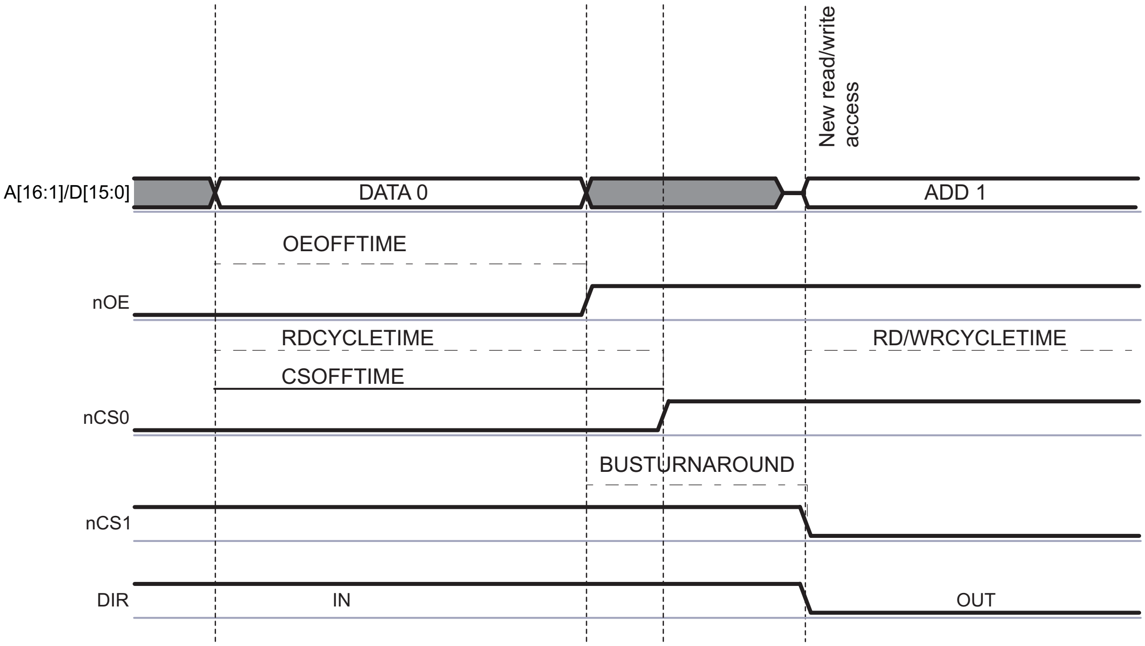
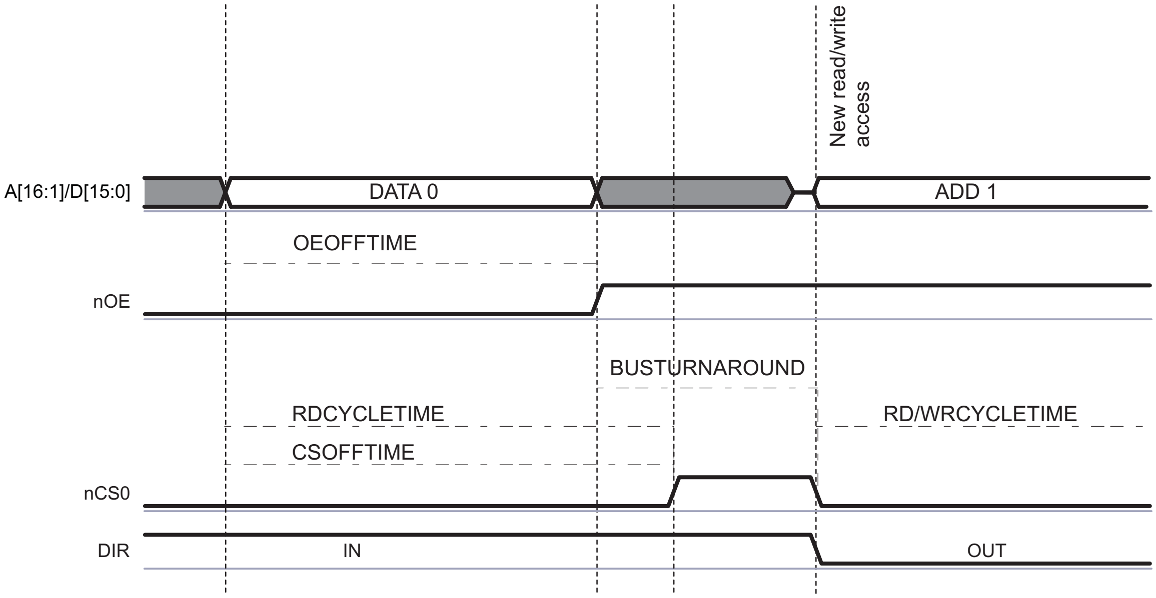


Figure 9-11. Read to Read / Write for a Address-Data or AAD-Multiplexed Device, On Same CS, With Bus Turnaround



9.1.3.3.8.3.7.2 Idle Cycles Between Accesses to Same Chip-Select (CYCLE2CYCLESAMECSEN, CYCLE2CYCLEDELAY)

Some devices require a minimum chip-select signal inactive time between accesses. The GPMC_CONFIG6_i[7] CYCLE2CYCLESAMECSEN bit enables insertion of a minimum number of GPMC_FCLK cycles, defined by the GPMC_CONFIG6_i[11-8] CYCLE2CYCLEDELAY field, between successive accesses of any type (read or write) to the same chip-select.

If CYCLE2CYCLESAMECSEN is enabled, any subsequent access to the same chip-select is delayed until its CYCLE2CYCLEDELAY completes. The CYCLE2CYCLEDELAY counter starts when CSRDOFFTIME/CSWROFFTIME completes.

The same applies to successive accesses occurring during 32-bit word or burst accesses split into successive single accesses when the single-access mode is used (GPMC_CONFIG1_i[30] READMULTIPLE = 0 or GPMC_CONFIG1_i[28] WRITEMULTIPLE = 0).

All control signals are kept in their default states during these idle GPMC_FCLK cycles. This prevents back-to-back accesses to the same chip-select without idle cycles between accesses.

9.1.3.3.8.3.7.3 Idle Cycles Between Accesses to Different Chip-Select (CYCLE2CYCLEDIFFCSEN, CYCLE2CYCLEDELAY)

Because of the pipelined behavior of the system, successive accesses to different chip-selects can occur back-to-back with no idle cycles between accesses. Depending on the control signals (CSn, ADV_ALEn, BE0_CLEn, OE_REn, WEn) assertion and de-assertion timing parameters and on the IC timing parameters, some control signals assertion times may overlap between the successive accesses to different CS. Similarly, some control signals (WEn, OE_REn) may not respect required transition times.

To work around the overlapping and to observe the required control-signal transitions, a minimum of CYCLE2CYCLEDELAY inactive cycles is inserted between the access being initiated to this chip-select and the previous access ending for a different chip-select. This applies to any type of access (read or write).

If GPMC_CONFIG6_i[6] CYCLE2CYCLEDIFFCSEN is enabled, the chip-select access is delayed until CYCLE2CYCLEDELAY cycles have expired since the end of a previous access to a different chip-select. CYCLE2CYCLEDELAY count starts at CSRDOFFTIME/CSWROFFTIME completion. All control signals are kept inactive during the idle GPMC_FCLK cycles.

CYCLE2CYCLESAMECSEN and CYCLE2CYCLEDIFFCSEN should be set in registers to respectively get idle cycles inserted between accesses on this chip-select and after accesses to a different chip-select.

The CYCLE2CYCLEDELAY delay runs in parallel with the BUSTURNAROUND delay. It should be noted that BUSTURNAROUND is a timing parameter defined for the ending chip-select access, whereas CYCLE2CYCLEDELAY is a timing parameter defined for the starting chip-select access. The effective minimum delay between successive accesses is based on the larger delay timing parameter and on access type combination, since bus turnaround does not apply to all access types. See Section 9.1.3.3.8.3.7.1 for more details on bus turnaround.

Table 9-10 describes the configuration required for idle cycle insertion.

Table 9-10. Idle Cycle Insertion Configuration

First Access Type	BUSTURNAROUND Timing Parameter	Second Access Type	Chip-Select	Addr/Data Multiplexed	CYCLE2CYCLE SAMECSEN Parameter	CYCLE2CYCLE DIFFCSEN Parameter	Idle Cycle Insertion Between the Two Accesses
R/W	0	R/W	Any	Any	0	x	No idle cycles are inserted if the two accesses are well pipelined.
R	>0	R	Same	Nonmuxed	x	0	No idle cycles are inserted if the two accesses are well pipelined.
R	>0	R	Different	Nonmuxed	0	0	BUSTURNAROUND cycles are inserted.
R	>0	R/W	Any	Muxed	0	0	BUSTURNAROUND cycles are inserted.
R	>0	W	Any	Any	0	0	BUSTURNAROUND cycles are inserted.
W	>0	R/W	Any	Any	0	0	No idle cycles are inserted if the two accesses are well pipelined.
R/W	0	R/W	Same	Any	1	x	CYCLE2CYCLEDELAY cycles are inserted.
R/W	0	R/W	Different	Any	x	1	CYCLE2CYCLEDELAY cycles are inserted.
R/W	>0	R/W	Same	Any	1	x	CYCLE2CYCLEDELAY cycles are inserted. If BTA idle cycles already apply on these two back-to-back accesses, the effective delay is max (BUSTURNAROUND, CYCLE2CYCLEDELAY).
R/W	>0	R/W	Different	Any	x	1	CYCLE2CYCLEDELAY cycles are inserted. If BTA idle cycles already apply on these two back-to-back accesses, the effective delay is maximum (BUSTURNAROUND, CYCLE2CYCLEDELAY).

9.1.3.3.8.3.8 Slow Device Support (TIMEPARAGRANULARITY Parameter)

All access-timing parameters can be multiplied by 2 by setting the GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY bit. Increasing all access timing parameters allows support of slow devices.

9.1.3.3.8.3.9 GPMC_DIR Pin

The GPMC_DIR pin is used to control I/O direction on the GPMC data bus GPMC_D[15-0]. Depending on top-level pad multiplexing, this signal can be output and used externally to the device, if required. The GPMC_DIR pin is low during transmit (OUT) and high during receive (IN).

For write accesses, the GPMC_DIR pin stays OUT from start-cycle time to end-cycle time.

For read accesses, the GPMC_DIR pin goes from OUT to IN at OEn assertion time and stays IN until:

- BUSTURNAROUND is enabled
  - The GPMC_DIR pin goes from IN to OUT at end-cycle time plus programmable bus turnaround time.
- BUSTURNAROUND is disabled
  - After an asynchronous read access, the GPMC_DIR pin goes from IN to OUT at RDACCESSTIME + 1 GPMC_FCLK cycle or when RDCYCLETIME completes, whichever occurs last.
  - After a synchronous read access, the GPMC_DIR pin goes from IN to OUT at RDACCESSTIME + 2 GPMC_FCLK cycles or when RDCYCLETIME completes, whichever occurs last.

Because of the bus-keeping feature of the GPMC, after a read or write access and with no other accesses pending, the default value of the GPMC_DIR pin is OUT (see Section 9.1.3.3.9.10). In nonmultiplexed devices, the GPMC_DIR pin stays IN between two successive read accesses to prevent unnecessary toggling.

9.1.3.3.8.3.10 Reset

No reset signal is sent to the external memory device by the GPMC. For more information about external-device reset, see Chapter 6, *Power, Reset, and Clock Management (PRCM)*.

The PRCM module provides an input pin, global_rst_n, to the GPMC:

- The global_rst_n pin is activated during device warm reset and cold reset.
- The global_rst_n pin initializes the internal state-machine and the internal configuration registers.

9.1.3.3.8.3.11 Write Protect Signal (WPn)

When connected to the attached memory device, the write protect signal can enable or disable the lockdown function of the attached memory. The GPMC_WPn output pin value is controlled through the GPMC_CONFIG[4] WRITEPROTECT bit, which is common to all CS.

9.1.3.3.8.3.12 Byte Enable (BE1n/BE0n)

Byte enable signals (BE1n/BE0n) are:

- Valid (asserted or nonasserted according to the incoming system request) from access start to access completion for asynchronous and synchronous single accesses
- Asserted low from access start to access completion for asynchronous and synchronous multiple read accesses
- Valid (asserted or nonasserted, according to the incoming system request) synchronously to each written data for synchronous multiple write accesses

9.1.3.3.8.4 Error Handling

When an error occurs in the GPMC, the error information is stored in the GPMC_ERR_TYPE register and the address of the illegal access is stored in the GPMC_ERR_ADDR register. The GPMC keeps only the first error abort information until the GPMC_ERR_TYPE register is reset. Subsequent accesses that cause errors are not logged until the error is cleared by hardware with the GPMC_ERR_TYPE[0]ERRORVALID bit.

- ERRORNOTSUPPADD occurs when an incoming system request address decoding does not match any valid chip-select region, or if two chip-select regions are defined as overlapped, or if a register file access is tried outside the valid address range of 1KB.
- ERRORNOTSUPPMCMD occurs when an unsupported command request is decoded at the L3 Slow interconnect interface
- ERRORTIMEOUT: A time-out mechanism prevents the system from hanging. The start value of the

9-bit time-out counter is defined in the GPMC_TIMEOUT_CTRL register and enabled with the GPMC_TIMEOUT_CTRL[0] TIMEOUTEN bit. When enabled, the counter starts at start-cycle time until it reaches 0 and data is not responded to from memory, and then a time-out error occurs. When data are sent from memory, this counter is reset to its start value. With multiple accesses (asynchronous page mode or synchronous burst mode), the counter is reset to its start value for each data access within the burst.

The GPMC does not generate interrupts on these errors. True abort to the MPU or interrupt generation is handled at the interconnect level.

9.1.3.3.9 Timing Setting

The GPMC offers the maximum flexibility to support various access protocols. Most of the timing parameters of the protocol access used by the GPMC to communicate with attached memories or devices are programmable on a chip-select basis. Assertion and deassertion times of control signals are defined to match the attached memory or device timing specifications and to get maximum performance during accesses. For more information on GPMC_CLK and GPMC_FCLK see Section 9.1.3.3.9.6.

In the following sections, the start access time refer to the time at which the access begins.

9.1.3.3.9.1 Read Cycle Time and Write Cycle Time (RDCYCLETIME / WRCYCLETIME)

The GPMC_CONFIG5_i[4-0] RDCYCLETIME and GPMC_CONFIG5_i[12-8] WRCYCLETIME fields define the address bus and byte enables valid times for read and write accesses. To ensure a correct duty cycle of GPMC_CLK between accesses, RDCYCLETIME and WRCYCLETIME are expressed in GPMC_FCLK cycles and must be multiples of the GPMC_CLK cycle. RDCYCLETIME and WRCYCLETIME bit fields can be set with a granularity of 1 or 2 throught GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY.

When either RDCYCLETIME or WRCYCLETIME completes, if they are not already deasserted, all control signals (CSn, ADV_ALEn, OE_REn, WEn, and BE0_CLEn) are deasserted to their reset values, regardless of their deassertion time parameters.

An exception to this forced deassertion occurs when a pipelined request to the same chip-select or to a different chip-select is pending. In such a case, it is not necessary to deassert a control signal with deassertion time parameters equal to the cycle-time parameter. This exception to forced deassertion prevents any unnecessary glitches. This requirement also applies to BE signals, thus avoiding an unnecessary BE glitch transition when pipelining requests.

If no inactive cycles are required between successive accesses to the same or to a different chip-select (GPMC_CONFIG6_i[7] CYCLE2CYCLESAMECSEN = 0 or GPMC_CONFIG6_i[6] CYCLE2CYCLEDIFFCSEN = 0, where i = 0 to 3), and if assertion-time parameters associated with the pipelined access are equal to 0, asserted control signals (CSn, ADV_ALEn, BE0_CLEn, WEn, and OE_REn) are kept asserted. This applies to any read/write to read/write access combination.

If inactive cycles are inserted between successive accesses, that is, CYCLE2CYCLESAMECSEN = 1 or CYCLE2CYCLEDIFFCSEN = 1, the control signals are forced to their respective default reset values for the number of GPMC_FCLK cycles defined in CYCLE2CYCLEDELAY.

9.1.3.3.9.2 CSn: Chip-Select Signal Control Assertion/Deassertion Time (CSONTIME / CSRDOFFTIME / CSWROFFTIME / CSEXTRADELAY)

The GPMC_CONFIG2_i[3-0] CSONTIME field defines the CSn signal-assertion time relative to the start access time. It is common for read and write accesses.

The GPMC_CONFIG2_i[12-8] CSRDOFFTIME (read access) and GPMC_CONFIG2_i[20-16] CSWROFFTIME (write access) bit fields define the CSn signal deassertion time relative to start access time.

CSONTIME, CSRDOFFTIME and CSWROFFTIME parameters are applicable to synchronous and asynchronous modes. CSONTIME can be used to control an address and byte enable setup time before chip-select assertion. CSRDOFFTIME and CSWROFFTIME can be used to control an address and byte enable hold time after chip-select deassertion.

CSn signal transitions as controlled through CSONTIME, CSRDOFFTIME, and CSWROFFTIME can be delayed by half a GPMC_FCLK period by enabling the GPMC_CONFIG2_i[7] CSEXTRADELAY bit. This half of a GPMC_FCLK period provides more granularity on the CSn assertion and deassertion time to guarantee proper setup and hold time relative to GPMC_CLK. CSEXTRADELAY is especially useful in configurations where GPMC_CLK and GPMC_FCLK have the same frequency, but can be used for all GPMC configurations. If enabled, CSEXTRADELAY applies to all parameters controlling CSn transitions.

The CSEXTRADELAY bit must be used carefully to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program the RDCYCLETIME and WRCYCLETIME bit fields to be greater than the CSn signal-deassertion time, including the extra half-GPMC_FCLK-period delay.

9.1.3.3.9.3 ADVn/ALE: Address Valid/Address Latch Enable Signal Control Assertion/Deassertion Time (ADVONTIME / ADVRDOFFTIME / ADVWROFFTIME / ADVEXTRADELAY/ADVAADMUXONTIME/ADVAADMUXRDOFFTIME/ADVAADMUXWROFFTIME)

The GPMC_CONFIG3_i[3-0] ADVONTIME field defines the ADVn_ALE signal-assertion time relative to start access time. It is common to read and write accesses.

The GPMC_CONFIG3_i[12-8] ADVRDOFFTIME (read access) and GPMC_CONFIG3_i[20-16] ADVWROFFTIME (write access) bit fields define the ADVn_ALE signal-deassertion time relative to start access time.

ADVONTIME can be used to control an address and byte enable valid setup time control before ADVn_ALE assertion. ADVRDOFFTIME and ADVWROFFTIME can be used to control an address and byte enable valid hold time control after ADVn_ALE de-assertion. ADVRDOFFTIME and ADVWROFFTIME are applicable to both synchronous and asynchronous modes.

ADVn_ALE signal transitions as controlled through ADVONTIME, ADVRDOFFTIME, and ADVWROFFTIME can be delayed by half a GPMC_FCLK period by enabling the GPMC_CONFIG3_i[7] ADVEXTRADELAY bit. This half of a GPMC_FCLK period provides more granularity on ADVn_ALE assertion and deassertion time to assure proper setup and hold time relative to GPMC_CLK. The ADVEXTRADELAY configuration parameter is especially useful in configurations where GPMC_CLK and GPMC_FCLK have the same frequency, but can be used for all GPMC configurations. If enabled, ADVEXTRADELAY applies to all parameters controlling ADVn_ALE transitions.

ADVEXTRADELAY must be used carefully to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program the RDCYCLETIME and WRCYCLETIME bit fields to be greater than ADVn_ALE signal-deassertion time, including the extra half-GPMC_FCLK-period delay.

The GPMC_CONFIG3_i[6-4] ADVAADMUXONTIME, GPMC_CONFIG3_i[26-24] ADVAADMUXRDOFFTIME, and GPMC_CONFIG3_i[30-28] ADVAADMUXWROFFTIME parameters have the same functions as ADVONTIME, ADVRDOFFTIME, and ADVWROFFTIME, but apply to the first address phase in the AAD-multiplexed protocol. It is the user responsibility to make sure ADVAADMUXxxOFFTIME is programmed to a value lower than or equal to ADVxxOFFTIME. Functionality in AAD-mux mode is undefined if the settings do not comply with this requirement. ADVAADMUXxxOFFTIME can be programmed to the same value as ADVONTIME if no high ADVn pulse is needed between the two AAD-mux address phases, which is the typical case in synchronous mode. In this configuration, ADVn is kept low until it reaches the correct ADVxxOFFTIME.

See Section 9.1.3.3.12 for more details on ADVONTIME, ADVRDOFFTIME, ADVWROFFTIME, and ADVAADMUXRDOFFTIME, ADVAADMUXWROFFTIME usage for CLE and ALE (Command / Address Latch Enable) usage for a NAND Flash interface.

9.1.3.3.9.4 OEn/REn: Output Enable / Read Enable Signal Control Assertion / Deassertion Time (OEONTIME / OEOFFTIME / OEEXTRADELAY / OEAADMUXONTIME / OEAADMUXOFFTIME)

The GPMC_CONFIG4_i[3-0] OEONTIME field defines the OEn_REn signal assertion time relative to start access time. It is applicable only to read accesses.

The GPMC_CONFIG4_i[12-8] OEOFFTIME field defines the OEn_REn signal deassertion time relative to start access time. It is applicable only to read accesses. OEn_REn is not asserted during a write cycle.

OEONTIME, OEOFFTIME, OEAADMUXONTIME and OEAADMUXOFFTIME parameters are applicable to synchronous and asynchronous modes. OEONTIME can be used to control an address and byte enable valid setup time control before OEn_REn assertion. OEOFFTIME can be used to control an address and byte enable valid hold time control after OEn_REn assertion.

OEAADMUXONTIME and OEAADMUXOFFTIME parameters have the same functions as OEONTIME and OEOFFTIME, but apply to the first OE assertion in the AAD-multiplexed protocol for a read phase, or to the only OE assertion for a write phase. It is the user responsibility to make sure OEAADMUXOFFTIME is programmed to a value lower than OEONTIME. Functionality in AAD-mux mode is undefined if the settings do not comply with this requirement. OEAADMUXOFFTIME shall never be equal to OEONTIME because the AAD-mux protocol requires a second address phase with the OEn signal de-asserted before OEn can be asserted again to define a read command.

The OEn_REn signal transitions as controlled through OEONTIME, OEOFFTIME, OEAADMUXONTIME and OEAADMUXOFFTIME can be delayed by half a GPMC_FCLK period by enabling the GPMC_CONFIG4_i[7] OEEXTRADELAY bit. This half of a GPMC_FCLK period provides more granularity on OEn_REn assertion and deassertion time to assure proper setup and hold time relative to GPMC_CLK. If enabled, OEEXTRADELAY applies to all parameters controlling OEn_REn transitions.

OEEXTRADELAY must be used carefully, to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program RDCYCLETIME and WRCYCLETIME to be greater than OEn_REn signal-deassertion time, including the extra half-GPMC_FCLK-period delay.

When the GPMC generates a read access to an address-/data-multiplexed device, it drives the address bus until OEn assertion time.

9.1.3.3.9.5 WEn: Write Enable Signal Control Assertion / Deassertion Time (WEONTIME / WEOFFTIME / WEEXTRADELAY)

The GPMC_CONFIG4_i[19-16] WEONTIME field (where i = 0 to 3) defines the WEn signal-assertion time relative to start access time. The GPMC_CONFIG4_i[28-24] WEOFFTIME field defines the WEn signal-deassertion time relative to start access time. These bit fields only apply to write accesses. WEn is not asserted during a read cycle.

WEONTIME can be used to control an address and byte enable valid setup time control before WEn assertion. WEOFFTIME can be used to control an address and byte enable valid hold time control after WEn assertion.

WEn signal transitions as controlled through WEONTIME, and WEOFFTIME can be delayed by half a GPMC_FCLK period by enabling the GPMC_CONFIG4_i[23] WEEXTRADELAY bit. This half of a GPMC_FCLK period provides more granularity on WEn assertion and deassertion time to guaranty proper setup and hold time relative to GPMC_CLK. If enabled, WEEXTRADELAY applies to all parameters controlling WEn transitions.

The WEEXTRADELAY bit must be used carefully to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program the WRCYCLETIME bit field to be greater than the WEn signal-deassertion time, including the extra half-GPMC_FCLK-period delay.

9.1.3.3.9.6 GPMC_CLK

GPMC_CLK is the external clock provided to the attached synchronous memory or device.

- The GPMC_CLK clock frequency is the GPMC_FCLK functional clock frequency divided by 1, 2, 3, or 4, depending on the GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER bit field, with a guaranteed 50-percent duty cycle.
- The GPMC_CLK clock is only activated when the access in progress is defined as synchronous (read or write access).
- The GPMC_CONFIG1_i[26-25] CLKACTIVATIONTIME field defines the number of GPMC_FCLK cycles from start access time to GPMC_CLK activation.
- The GPMC_CLK clock is stopped when cycle time completes and is asserted low between accesses.
- The GPMC_CLK clock is kept low when access is defined as asynchronous.
- When the GPMC is configured for synchronous mode, the GPMC_CLK signal (which is an output) must also be set as an input in the Pin Mux configuration for the pin. GPMC_CLK is looped back through the output and input buffers of the corresponding GPMC_CLK pad at the device boundary. The looped-back clock is used to synchronize the sampling of the memory signals.

When cycle time completes, the GPMC_CLK may be high because of the GPMCFCLKDIVIDER bit field. To ensure correct stoppage of the GPMC_CLK clock within the 50-percent required duty cycle, it is the user's responsibility to extend the RDCYCLETIME or WRCYCLETIME value.

To ensure a correct external clock cycle, the following rules must be applied:

- (RDCYCLETIME - CLKACTIVATIONTIME) must be a multiple of (GPMCFCLKDIVIDER + 1).
- The PAGEBURSTACCESSTIME value must be a multiple of (GPMCFCLKDIVIDER + 1).

9.1.3.3.9.7 GPMC_CLK and Control Signals Setup and Hold

Control-signal transition (assertion and deassertion) setup and hold values with respect to the GPMC_CLK edge can be controlled in the following ways:

- For the GPMC_CLK signal, the GPMC_CONFIG1_i[26-25] CLKACTIVATIONTIME field allows setup and hold control of control-signal assertion time.
- The use of a divided GPMC_CLK allows setup and hold control of control-signal assertion and deassertion times.
- When GPMC_CLK runs at the GPMC_FCLK frequency so that GPMC_CLK edge and control-signal transitions refer to the same GPMC_FCLK edge, the control-signal transitions can be delayed by half of a GPMC_FCLK period to provide minimum setup and hold times. This half-GPMC_FCLK delay is enabled with the CSEXTRADELAY, ADVEXTRADELAY, OEEXTRADELAY, or WEEXTRADELAY parameter. This delay must be used carefully to prevent control-signal overlap between successive accesses to different chip-selects. This implies that the RDCYCLETIME and WRCYCLETIME are greater than the last control-signal deassertion time, including the extra half-GPMC_FCLK cycle.

9.1.3.3.9.8 Access Time (RDACCESSTIME / WRACCESSTIME)

The read access time and write access time durations can be programmed independently through GPMC_CONFIG5_i[20-16] RDACCESSTIME and GPMC_CONFIG6_i[28-24] WRACCESSTIME. This allows OEn and GPMC data capture timing parameters to be independent of WEn and memory device data capture timing parameters. RDACCESSTIME and WRACCESSTIME bit fields can be set with a granularity of 1 or 2 throught GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY.

9.1.3.3.9.8.1 Access Time on Read Access

In asynchronous read mode, for single and paged accesses, GPMC_CONFIG5_i[[20-16] RDACCESSTIME field defines the number of GPMC_FCLK cycles from start access time to the GPMC_FCLK rising edge used for the first data capture. RDACCESSTIME must be programmed to the rounded greater value (in GPMC_FCLK cycles) of the read access time of the attached memory device.

In synchronous read mode, for single or burst accesses, RDACCESSTIME defines the number of GPMC_FCLK cycles from start access time to the GPMC_FCLK rising edge corresponding to the GPMC_CLK rising edge used for the first data capture.

GPMC_CLK which is sent to the memory device for synchronization with the GPMC controller, is internally retimed to correctly latch the returned data. GPMC_CONFIG5_i[4-0] RDCYCLETIME must be greater than RDACCESSTIME in order to let the GPMC latch the last return data using the internally retimed GPMC_CLK.

The external WAIT signal can be used in conjunction with RDACCESSTIME to control the effective GPMC data-capture GPMC_FCLK edge on read access in both asynchronous mode and synchronous mode. For details about wait monitoring, see Section 9.1.3.3.8.1.

9.1.3.3.9.8.2 Access Time on Write Access

In asynchronous write mode, the GPMC_CONFIG6_i[[28-24] WRACCESSTIME timing parameter is not used to define the effective write access time. Instead, it is used as a WAIT invalid timing window, and must be set to a correct value so that the gpmc_wait pin is at a valid state two GPMC_CLK cycles before WRACCESSTIME completes. For details about wait monitoring, see Section 9.1.3.3.8.1.

In synchronous write mode , for single or burst accesses, WRACCESSTIME defines the number of GPMC_FCLK cycles from start access time to the GPMC_CLK rising edge used by the memory device for the first data capture.

The external WAIT signal can be used in conjunction with WRACCESSTIME to control the effective memory device data capture GPMC_CLK edge for a synchronous write access. For details about wait monitoring, see Section 9.1.3.3.8.1.

9.1.3.3.9.9 Page Burst Access Time (PAGEBURSTACCESSTIME)

GPMC_CONFIG5_i[27-24] PAGEBURSTACCESSTIME bit field can be set with a granularity of 1 or 2 throught the GPMC_CONFIG1_i[[4] TIMEPARAGRANULARITY.

9.1.3.3.9.9.1 Page Burst Access Time on Read Access

In asynchronous page read mode, the delay between successive word captures in a page is controlled through the PAGEBURSTACCESSTIME bit field. The PAGEBURSTACCESSTIME parameter must be programmed to the rounded greater value (in GPMC_FCLK cycles) of the read access time of the attached device.

In synchronous burst read mode, the delay between successive word captures in a burst is controlled through the PAGEBURSTACCESSTIME field.

The external WAIT signal can be used in conjunction with PAGEBURSTACCESSTIME to control the effective GPMC data capture GPMC_FCLK edge on read access. For details about wait monitoring, see Section 9.1.3.3.8.1.

9.1.3.3.9.9.2 Page Burst Access Time on Write Access

Asynchronous page write mode is not supported. PAGEBURSTACCESSTIME is irrelevant in this case.

In synchronous burst write mode, PAGEBURSTACCESSTIME controls the delay between successive memory device word captures in a burst.

The external WAIT signal can be used in conjunction with PAGEBURSTACCESSTIME to control the effective memory-device data capture GPMC_CLK edge in synchronous write mode. For details about wait monitoring, see Section 9.1.3.3.8.1.

9.1.3.3.9.10 Bus Keeping Support

At the end-cycle time of a read access, if no other access is pending, the GPMC drives the bus with the last data read after RDCYCLETIME completion time to prevent bus floating and reduce power consumption.

After a write access, if no other access is pending, the GPMC keeps driving the data bus after WRCYCLETIME completes with the same data to prevent bus floating and power consumption.

9.1.3.3.10 NOR Access Description

For each chip-select configuration, the read access can be specified as either asynchronous or synchronous access through the GPMC_CONFIG1_i[29] READTYPE bit. For each chip-select configuration, the write access can be specified as either synchronous or asynchronous access through the GPMC_CONFIG1_i[27] WRITETYPE bit.

Asynchronous and synchronous read and write access time and related control signals are controlled through timing parameters that refer to GPMC_FCLK. The primary difference of synchronous mode is the availability of a configurable clock interface (GPMC_CLK) to control the external device. Synchronous mode also affects data-capture and wait-pin monitoring schemes in read access.

For details about asynchronous and synchronous access, see the descriptions of GPMC_CLK, RdAccessTime, WrAccessTime, and wait-pin monitoring.

For more information about timing-parameter settings, see the sample timing diagrams in this chapter.

The address bus and BE[1:0]n are fixed for the duration of a synchronous burst read access, but they are updated for each beat of an asynchronous page-read access.

9.1.3.3.10.1 Asynchronous Access Description

This section describes:

- Asynchronous single read operation on an address/data multiplexed device
- Asynchronous single write operation on an address/data-multiplexed device
- Asynchronous single read operation on an AAD-multiplexed device
- Asynchronous single write operation on an AAD-multiplexed device
- Asynchronous multiple (page) read operation on a non-multiplexed device

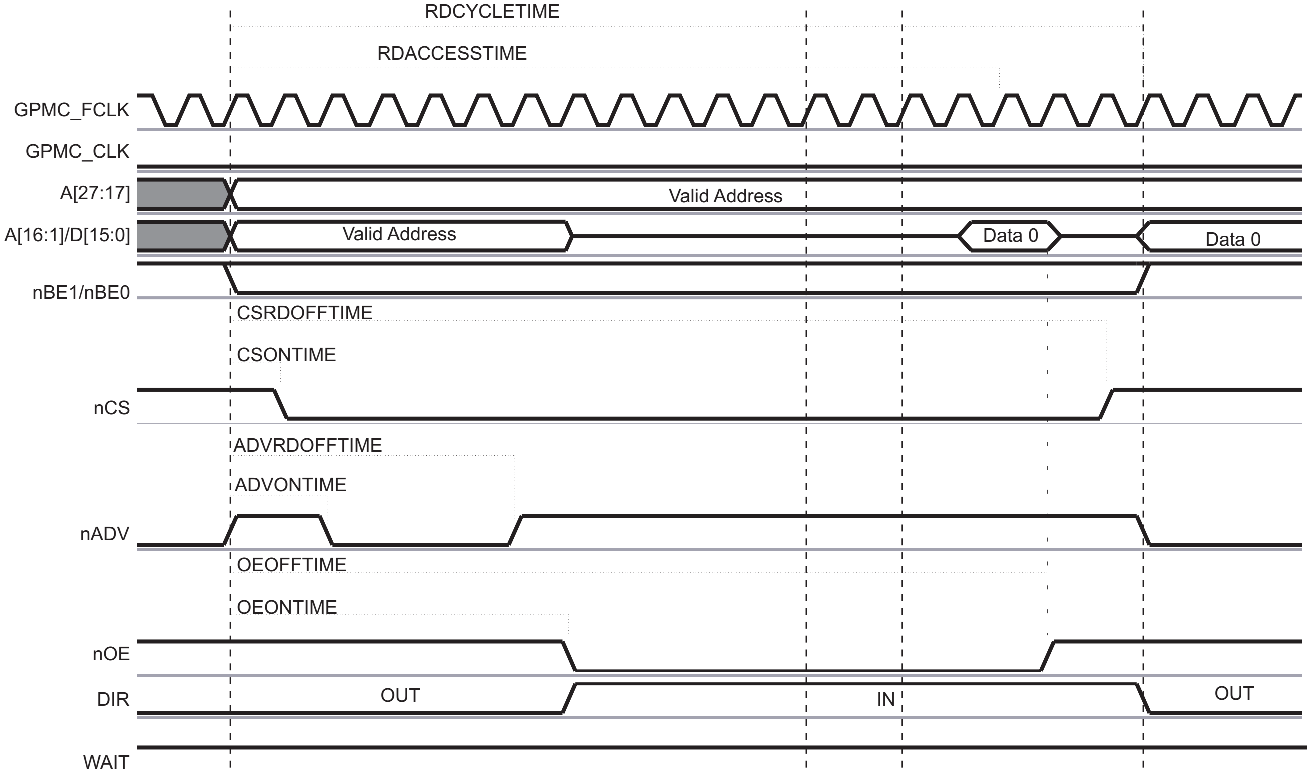
In asynchronous operations GPMC_CLK is not provided outside the GPMC and is kept low.

9.1.3.3.10.1.1 Access on Address/Data Multiplexed Devices

9.1.3.3.10.1.1.1 Asynchronous Single-Read Operation on an Address/Data Multiplexed Device

Figure 9-12 shows an asynchronous single read operation on an address/data-multiplexed device.

Figure 9-12. Asynchronous Single Read Operation on an Address/Data Multiplexed Device



9.1.3.3.10.1.1.2 Asynchronous Single Read on an Address/Data-Multiplexed Device

See the device-specific datasheet for formulas to calculate timing parameters.

Table 9-41 lists the timing bit fields to set up in order to configure the GPMC in asynchronous single read mode.

When the GPMC generates a read access to an address/data-multiplexed device, it drives the address bus until OEn assertion time. For details, see Section 9.1.3.3.8.2.3.

Address bits (A[16:1] from a GPMC perspective, A[15:0] from an external device perspective) are placed on the address/data bus, and the remaining address bits GPMC_A[25:16] are placed on the address bus. The address phase ends at OEn assertion, when the DIR signal goes from OUT to IN.

- Chip-select signal CSn
  - CSn assertion time is controlled by the GPMC_CONFIG2_i[3-0] CSONTIME field. It controls the address setup time to CSn assertion.
  - CSn deassertion time is controlled by the GPMC_CONFIG2_i[12-8] CSRDOFFTIME field. It controls the address hold time from CSn deassertion
- Address valid signal ADVn
  - ADVn assertion time is controlled by the GPMC_CONFIG3_i[3-0] ADVONTIME field.
  - ADVn deassertion time is controlled by the GPMC_CONFIG3_i[[12-8] ADVRDOFFTIME field.

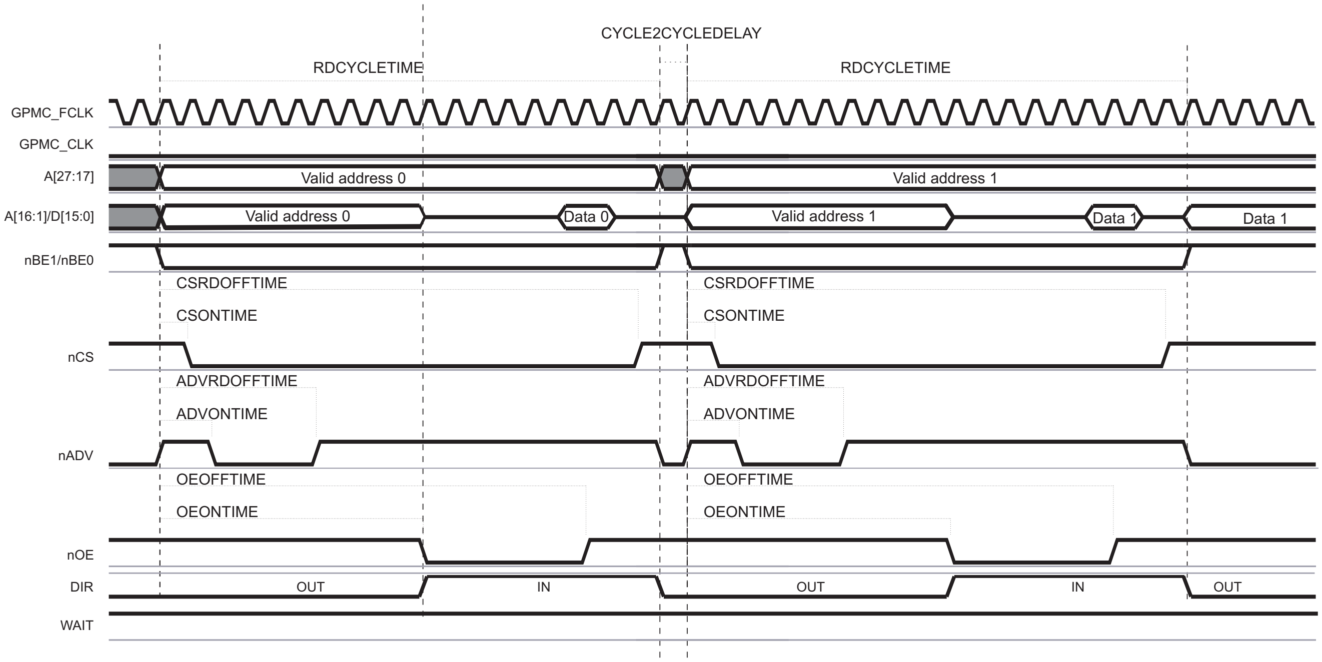
- Output enable signal OEn
  - OEn assertion indicates a read cycle.
  - OEn assertion time is controlled by the GPMC_CONFIG4_i[[3-0] OEONTIME field.
  - OEn deassertion time is controlled by the GPMC_CONFIG4_i[[12-8] OEOFFTIME field.
- Read data is latched when RDACCESSTIME completes. Access time is defined in the GPMC_CONFIG5_i[20-16] RDACCESSTIME field.
- Direction signal DIR: DIR goes from OUT to IN at the same time that OEn is asserted.
- The end of the access is defined by the GPMC_CONFIG5_i[4-0] RDCYCLETIME parameter.

In the GPMC, when a 16-bit wide device is attached to the controller, a 32-bit word write access is split into two 16-bit word write accesses. For more information about GPMC access size and type adaptation, see Section 9.1.3.3.10.5. Between two successive accesses, if a OEn pulse is needed:

- The GPMC_CONFIG6_i[[11-8] CYCLE2CYCLEDELAY field can be programmed with GPMC_CONFIG6_i[[7] CYCLE2CYCLESAMECSEN enabled.
- The CSWROFFTIME and CSONTIME parameters also allow a chip-select pulse, but this affects all other types of access.

Figure 9-13 shows two asynchronous single-read accesses on an address/data-multiplexed devices.

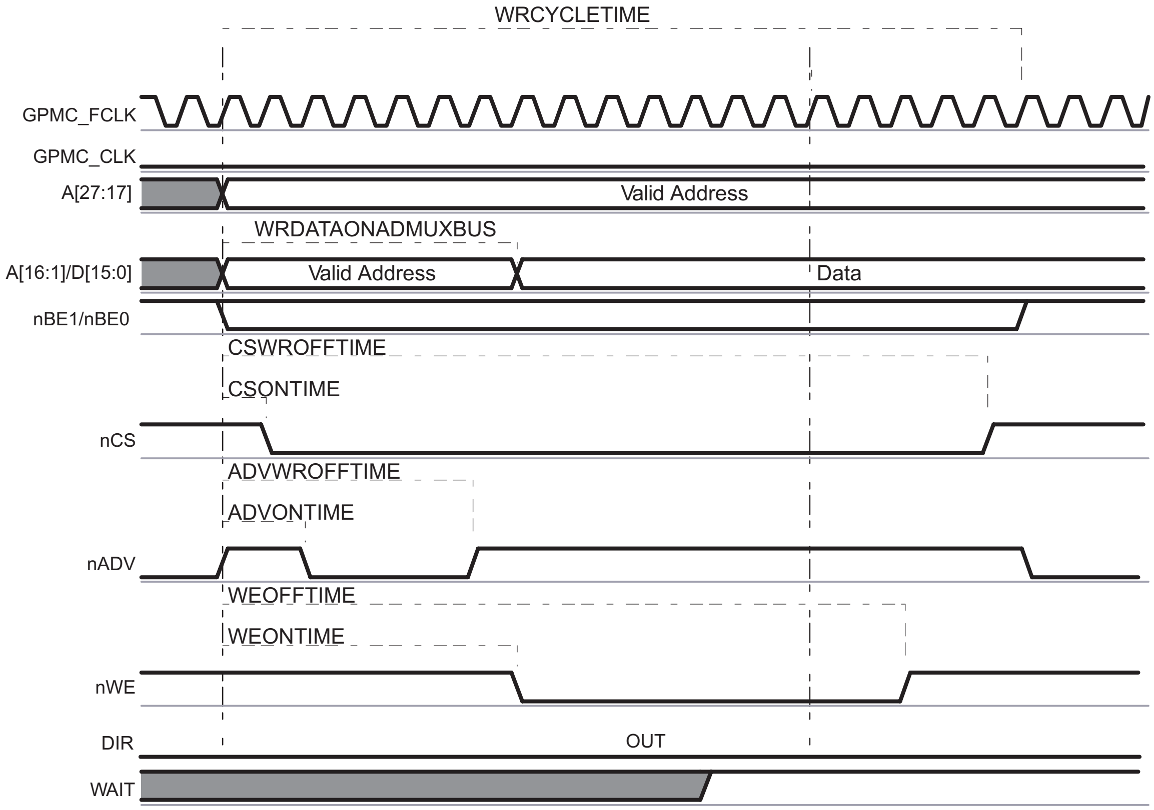
Figure 9-13. Two Asynchronous Single Read Accesses on an Address/Data Multiplexed Device (32-Bit Read Split Into 2 × 16-Bit Read)



9.1.3.3.10.1.1.3 Asynchronous Single Write Operation on an Address/Data-Multiplexed Device

Figure 9-14 shows an asynchronous single write operation on an address/data-multiplexed device.

Figure 9-14. Asynchronous Single Write on an Address/Data-Multiplexed Device



9.1.3.3.10.1.1.4 Asynchronous Single Write on an Address/Data-Multiplexed Device

See the device-specific datasheet for formulas to calculate timing parameters.

Table 9-41 lists the timing bit fields to set up in order to configure the GPMC in asynchronous single write mode. When the GPMC generates a write access to an address/data-multiplexed device, it drives the address bus until WEn assertion time. For more information, see Section 9.1.3.3.8.2.3.

The CSn and ADVn signals are controlled in the same way as for asynchronous single read operation on an address/data-multiplexed device.

- Write enable signal WEn
  - WEn assertion indicates a write cycle.
  - WEn assertion time is controlled by the GPMC_CONFIG4_i[19-16] WEONTIME field.
  - WEn deassertion time is controlled by the GPMC_CONFIG4_i[28-24] WEOFFTIME field.
- Direction signal DIR: DIR signal is OUT during the entire access.
- The end of the access is defined by the GPMC_CONFIG5_i[12-8] WRCYCLETIME parameter.

Address bits A[16:1] (GPMC point of view) are placed on the address/data bus at the start of cycle time, and the remaining address bits A[26:17] are placed on the address bus.

Data is driven on the address/data bus at a GPMC_CONFIG6_i[19-16] WRDATAONADMUXBUS time.

Write multiple access in asynchronous mode is not supported. If WRITEMULTIPLE is enabled with WRITETYPE as asynchronous, the GPMC processes single asynchronous accesses.

After a write operation, if no other access (read or write) is pending, the data bus keeps its previous value. See Section 9.1.3.3.9.10.

9.1.3.3.10.1.1.5 Asynchronous Multiple (Page) Write Operation on an Address/Data-Multiplexed Device

Write multiple (page) access in asynchronous mode is not supported for address/data-multiplexed devices. If GPMC_CONFIG1_i[28] WRITEMULTIPLE is enabled (1) with GPMC_CONFIG1_i[27] WRITETYPE as asynchronous (0), the GPMC processes single asynchronous accesses.

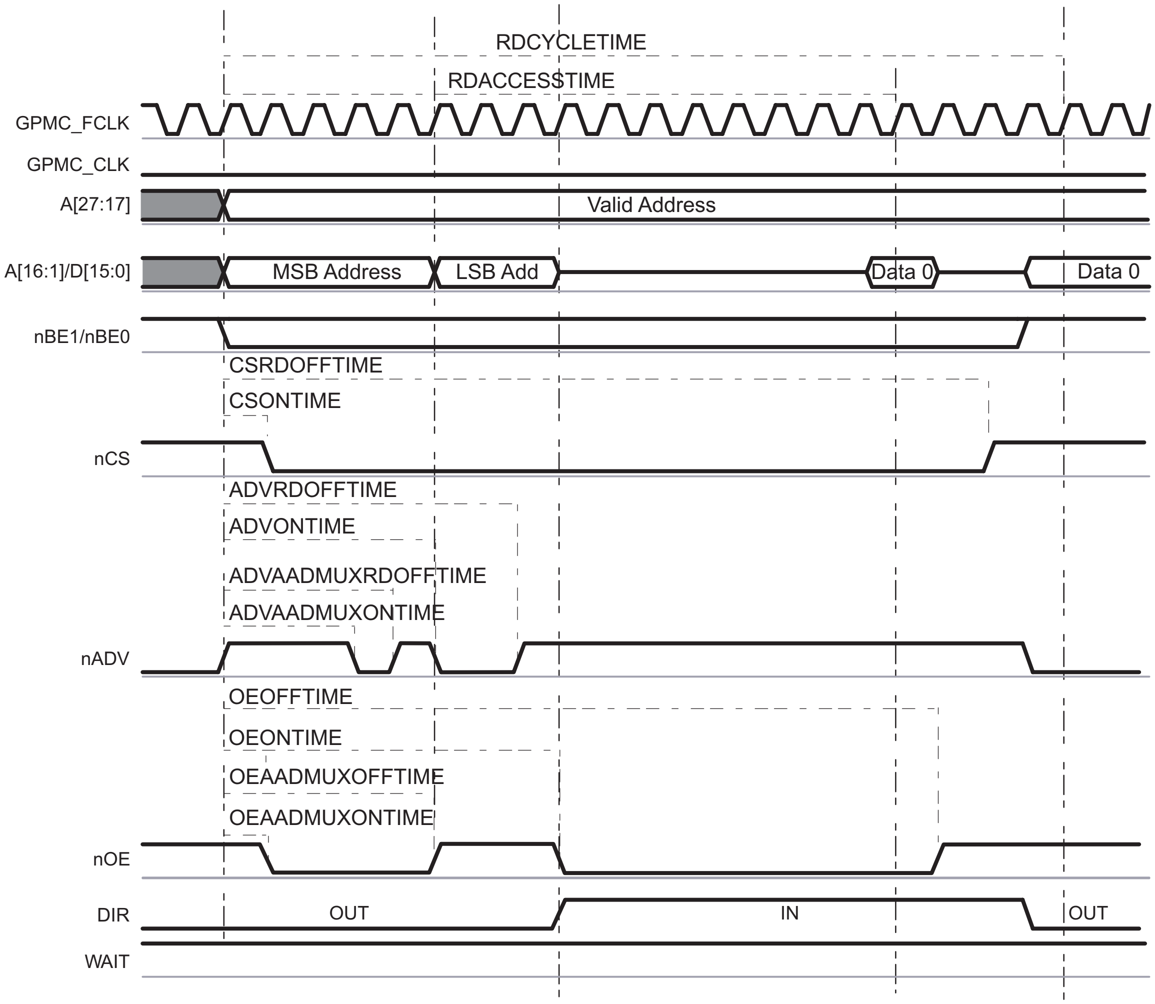
For accesses on non-multiplexed devices, see Section 9.1.3.3.10.3.

9.1.3.3.10.1.2 Access on Address/Address/Data (AAD) Multiplexed Devices

9.1.3.3.10.1.2.1 Asynchronous Single Read Operation on an AAD-Multiplexed Device

Figure 9-15 shows an asynchronous single read operation on an AAD-multiplexed device.

Figure 9-15. Asynchronous Single-Read on an AAD-Multiplexed Device



9.1.3.3.10.1.2.2 Asynchronous Single Read on an AAD-Multiplexed Device

See the device-specific datasheet for formulas to calculate timing parameters.

Table 9-41 lists the timing bit fields to set up in order to configure the GPMC in asynchronous single write mode.

When the GPMC generates a read access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with OEn driven low. The first address phase ends at the first OEn deassertion time. The second phase for LSB address is qualified with OEn driven high. The second address phase ends at the second OEn assertion time, when the DIR signal goes from OUT to IN.

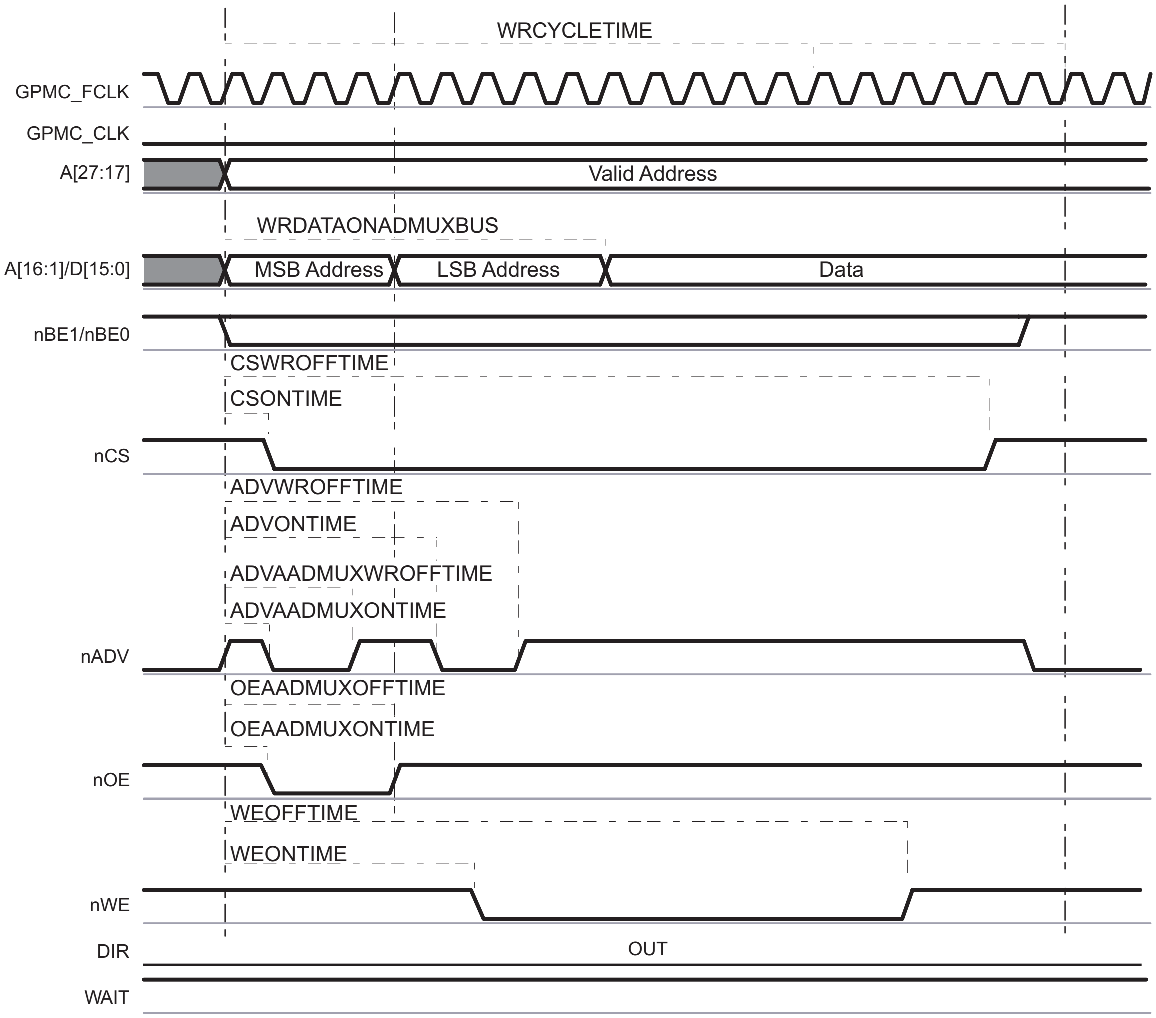
The CSn and DIR signals are controlled in the same way as for asynchronous single read operation on an address/data-multiplexed device.

- Address valid signal ADVn. ADVn is asserted and deasserted twice during a read transaction:
  - ADVn first assertion time is controlled by the GPMC_CONFIG3_i[6-4] ADVAADMUXONTIME field.
  - ADVn first deassertion time is controlled by the GPMC_CONFIG3_i[26-24] ADVAADMUXRDOFFTIME field.
  - ADVn second assertion time is controlled by the GPMC_CONFIG3_i[3-0] ADVONTIME field.
  - ADVn second deassertion time is controlled by the GPMC_CONFIG3_i[12-8] ADVRDOFFTIME field.
- Output Enable signal OEn. OEn is asserted and deasserted twice during a read transaction (OEn second assertion indicates a read cycle):
  - OEn first assertion time is controlled by the GPMC_CONFIG4_i[6-4] OEAADMUXONTIME field.
  - OEn first deassertion time is controlled by the GPMC_CONFIG3_i[15-13] OEAADMUXOFFTIME field.
  - OEn second assertion time is controlled by the GPMC_CONFIG4_i[3-0] OEONTIME field.
  - OEn second deassertion time is controlled by the GPMC_CONFIG4_i[12-8] OEOFFTIME field.

9.1.3.3.10.1.2.3 Asynchronous Single Write Operation on an AAD-Multiplexed Device

Figure 9-16 shows an asynchronous single write operation on an AAD-multiplexed device.

Figure 9-16. Asynchronous Single Write on an AAD-Multiplexed Device



See the device-specific datasheet for formulas to calculate timing parameters.

Table 9-41 lists the timing bit fields to set up to configure the GPMC in asynchronous single write mode.

When the GPMC generates a write access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with OEn driven low. The second phase for LSB address is qualified with OEn driven high. The address phase ends at WEn assertion time.

The CSn, WEn, and DIR signals are controlled in the same way as for asynchronous single write operation on an address/data-multiplexed device.

- Address valid signal ADVn is asserted and deasserted twice during a write transaction
  - ADVn first assertion time is controlled by the GPMC_CONFIG3_i[6-4] ADVAADMUXONTIME field.
  - ADVn first deassertion time is controlled by the GPMC_CONFIG3_i[30-28] ADVAADMUXWROFFTIME field.
  - ADVn second assertion time is controlled by the GPMC_CONFIG3_i[3-0] ADVONTIME field.
  - ADVn second deassertion time is controlled by the GPMC_CONFIG3_i[20-16] ADVWROFFTIME field.
- Output Enable signal OEn is asserted during the address phase of a write transaction
  - OEn assertion time is controlled by the GPMC_CONFIG4_i[6-4] OEAADMUXONTIME field.
  - OEn deassertion time is controlled by the GPMC_CONFIG3_i[15-13] OEAADMUXOFFTIME field.

The address bits for the first address phase are driven onto the data bus until OEn deassertion. Data is driven onto the address/data bus at the clock edge defined by the GPMC_CONFIG6_i[19-16] WRDATAONADMUXBUS parameter.

9.1.3.3.10.1.2.4 Asynchronous Multiple (Page) Read Operation on an AAD-Multiplexed Device

Write multiple (page) access in asynchronous mode is not supported for AAD-multiplexed devices.

If GPMC_CONFIG1_i[28] WRITEMULTIPLE is enabled (1) with GPMC_CONFIG1_i[27] WRITETYPE as asynchronous (0), the GPMC processes single asynchronous accesses.

For accesses on non-multiplexed devices, see Section 9.1.3.3.10.3.

9.1.3.3.10.2 Synchronous Access Description

This section details read and write synchronous accesses on address/data multiplexed. All information in this section can be applied to any type of memory - non-multiplexed, address and data multiplexed or AAD-multiplexed - with a difference limited to the address phase. For accesses on non-multiplexed devices, see Section 9.1.3.3.10.3.

In synchronous operations:

- The GPMC_CLK clock is provided outside the GPMC when accessing the memory device.
- The GPMC_CLK clock is derived from the GPMC_FCLK clock using the GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER field. In the following section, i stands for the chip-select number, i = 0 to 3.
- The GPMC_CONFIG1_i[26-25] CLKACTIVATIONTIME field specifies that the GPMC_CLK is provided outside the GPMC 0, 1, or 2 GPMC_FCLK cycles after start access time until RDCYCLETIME or WRCYCLETIME completion.

9.1.3.3.10.2.1 Synchronous Single Read

Figure 9-17 and Figure 9-18 show a synchronous single-read operation with GPMCFCLKDIVIDER equal to 0 and 1, respectively.

Figure 9-17. Synchronous Single Read (GPMCFCLKDIVIDER = 0)

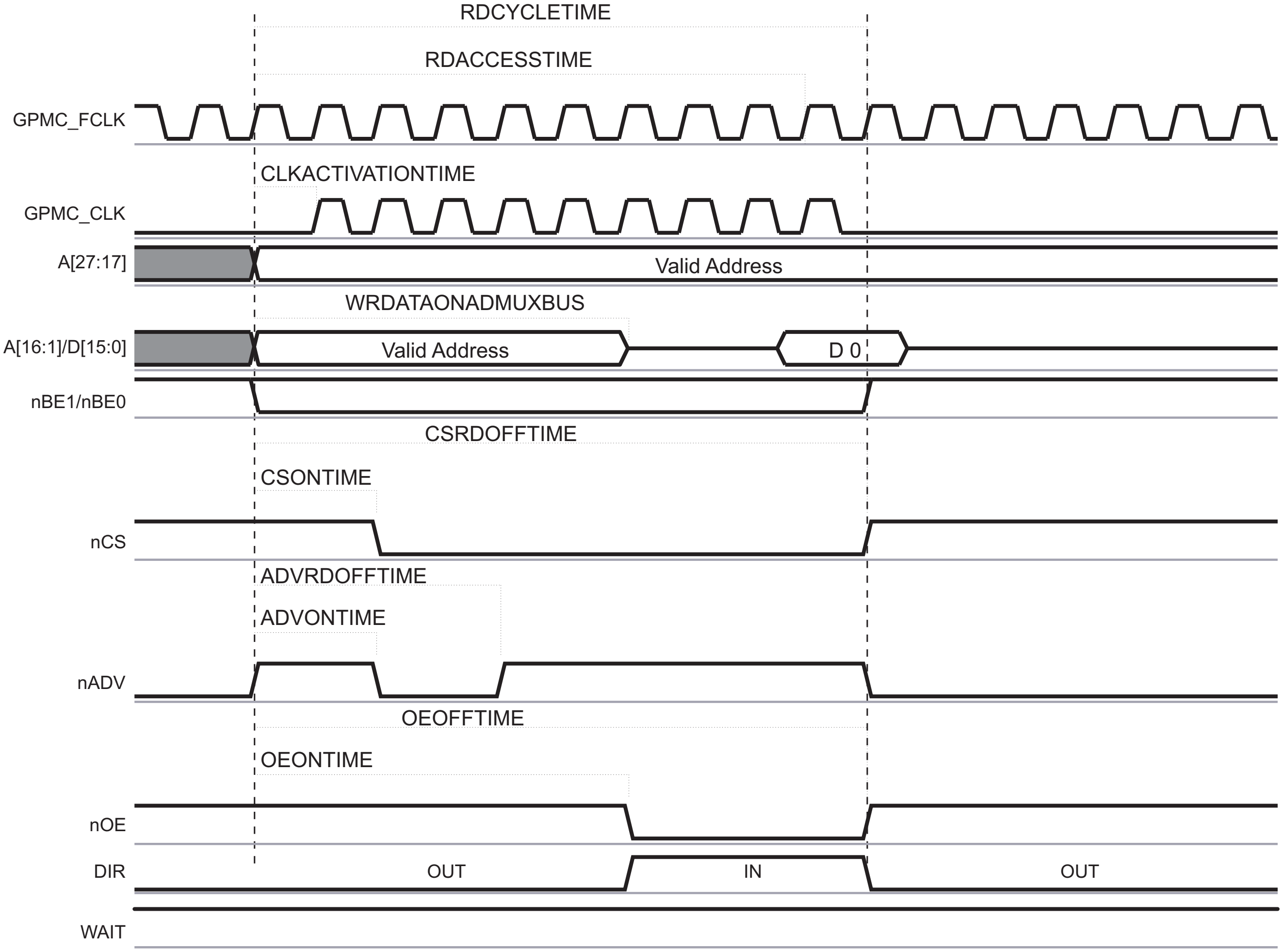
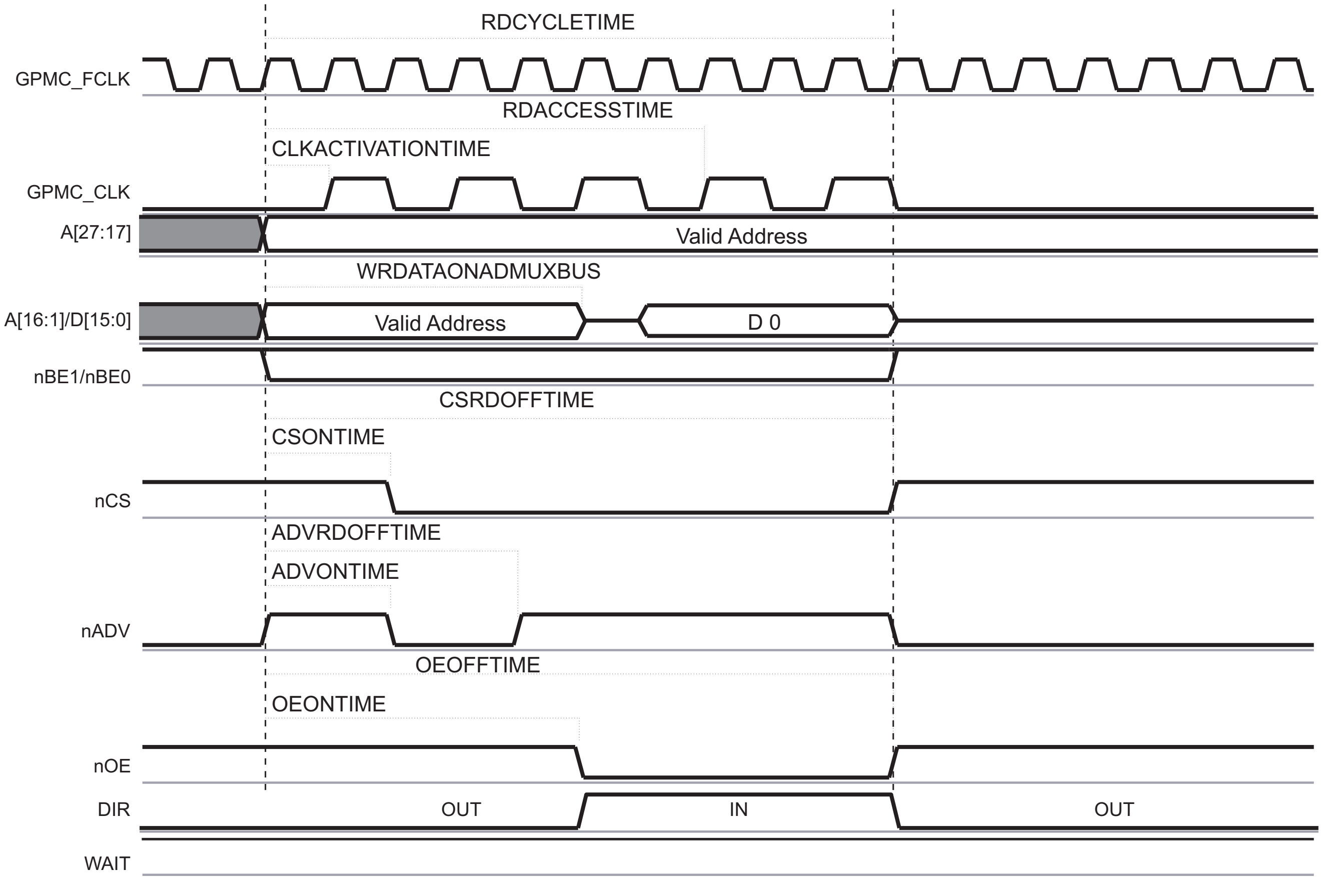


Figure 9-18. Synchronous Single Read (GPMCFCLKDIVIDER = 1)



See the device-specific datasheet for formulas to calculate timing parameters.

Table 9-41 lists the timing bit fields to set up in order to configure the GPMC in asynchronous single read mode.

When the GPMC generates a read access to an address/data-multiplexed device, it drives the address bus until OEn assertion time. For details, see Section 9.1.3.3.8.2.3.

- Chip-select signal CSn
  - CSn assertion time is controlled by the GPMC_CONFIG2_i[3-0] CSONTIME field and ensures address setup time to CSn assertion.
  - CSn deassertion time is controlled by the GPMC_CONFIG2_i[12-8] CSRDOFFTIME field and ensures address hold time to CSn deassertion.
- Address valid signal ADVn
  - ADVn assertion time is controlled by the GPMC_CONFIG3_i[3-0] ADVONTIME field.
  - ADVn deassertion time is controlled by the GPMC_CONFIG3_i[12-8] ADVRDOFFTIME field.
- Output enable signal OEn
  - OEn assertion indicates a read cycle.
  - OEn assertion time is controlled by the GPMC_CONFIG4_i[3-0] OEONTIME field.
  - OEn deassertion time is controlled by the GPMC_CONFIG4_i[12-8] OEOFFTIME field.
- Initial latency for the first read data is controlled by GPMC_CONFIG5_i[20-16] RDACCESSTIME or by monitoring the WAIT signal.
- Total access time (GPMC_CONFIG5_i[4-0] RDCYCLETIME) corresponds to RDACCESSTIME plus the address hold time from CSn deassertion, plus time from RDACCESSTIME to CSRDOFFTIME.

- Direction signal DIR: DIR goes from OUT to IN at the same time as OEn assertion.

When the GPMC generates a write access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with OEn driven low. The second phase for LSB address is qualified with OEn driven high. The address phase ends at WEn assertion time.

The CSn and DIR signals are controlled in the same way as for synchronous single read operation on an address/data-multiplexed device.

- Address valid signal ADVn is asserted and deasserted twice during a read transaction
  - ADVn first assertion time is controlled by the GPMC_CONFIG3_i[6-4] ADVAADMUXONTIME field.
  - ADVn first deassertion time is controlled by the GPMC_CONFIG3_i[26-24] ADVAADMUXRDOFFTIME field.
  - ADVn second assertion time is controlled by the GPMC_CONFIG3_i[3-0] ADVONTIME field.
  - ADVn second deassertion time is controlled by the GPMC_CONFIG3_i[12-8] ADVRDOFFTIME field.
- Output Enable signal OEn is asserted and deasserted twice during a read transaction (OEn second assertion indicates a read cycle)
  - OEn first assertion time is controlled by the GPMC_CONFIG4_i[6-4] OEAADMUXONTIME field.
  - OEn first deassertion time is controlled by the GPMC_CONFIG3_i[15-13] OEAADMUXOFFTIME field.
  - OEn second assertion time is controlled by the GPMC_CONFIG4_i[3-0] OEONTIME field.
  - OEn second deassertion time is controlled by the GPMC_CONFIG4_i[12-8] OEOFFTIME field.

After a read operation, if no other access (read or write) is pending, the data bus is driven with the previous read value. See Section 9.1.3.3.9.10.

9.1.3.3.10.2.2 Synchronous Multiple (Burst) Read (4-, 8-, 16-Word16 Burst With Wraparound Capability)

Figure 9-19 and Figure 9-20 show a synchronous multiple read operation with GPMCFCLKDivider equal to 0 and 1, respectively.

Figure 9-19. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 0)

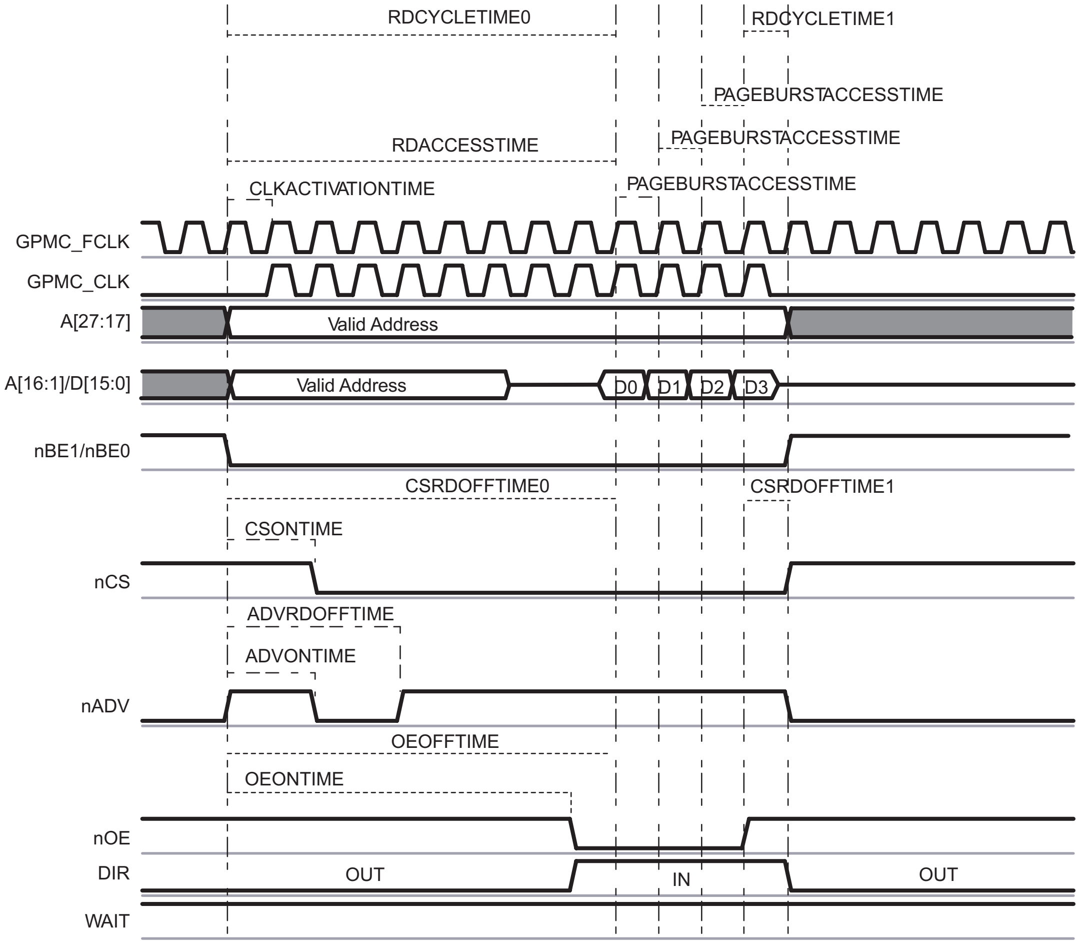
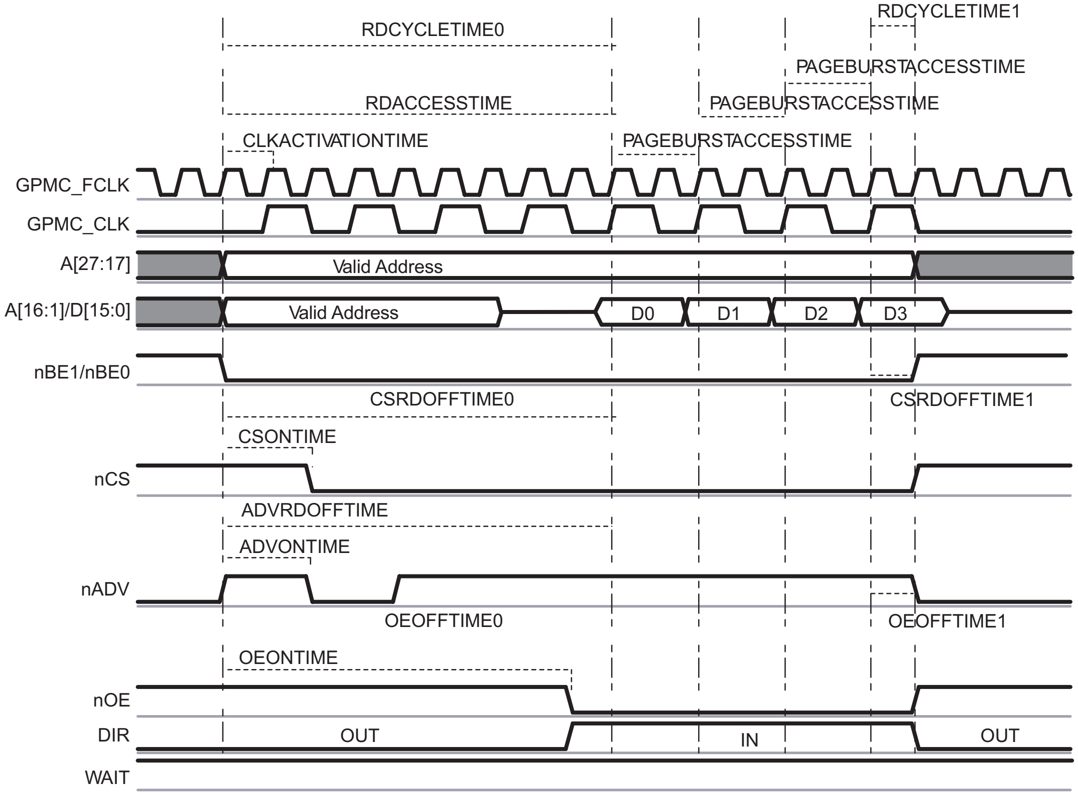


Figure 9-20. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 1)



When GPMC_CONFIG5_i[20-16] RDACCESSTIME completes, control-signal timings are frozen during the multiple data transactions, corresponding to GPMC_CONFIG5_i[27-24] PAGEBURSTACCESSTIME multiplied by the number of remaining data transactions.

The CSn, ADVn, OEn and DIR signals are controlled in the same way as for synchronous single read operation. See Section 9.1.3.3.10.2.1.

Initial latency for the first read data is controlled by RDACCESSTIME or by monitoring the WAIT signal. Successive read data are provided by the memory device each one or two GPMC_CLK cycles. The PAGEBURSTACCESSTIME parameter must be set accordingly with GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER and the memory-device internal configuration. Depending on the device page length, the GPMC checks device page crossing during a new burst request and purposely insert initial latency (of RDACCESSTIME) when required.

Total access time GPMC_CONFIG5_i[4-0] RDCYCLETIME corresponds to RDACCESSTIME plus the address hold time from CSn deassertion. In Figure 9-19, RDCYCLETIME programmed value equals to RDCYCLETIME0 + RDCYCLETIME1.

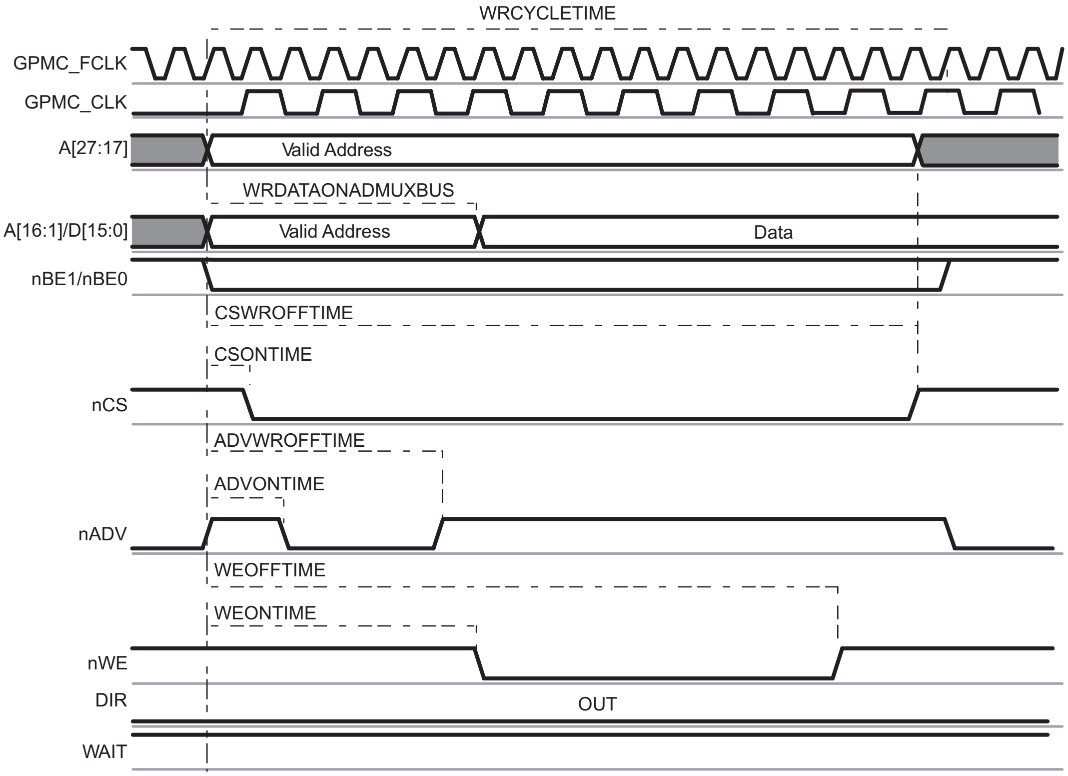
After a read operation, if no other access (read or write) is pending, the data bus is driven with the previous read value. See Section 9.1.3.3.9.10.

Burst wraparound is enabled through the GPMC_CONFIG1_i[31] WRAPBURST bit and allows a 4-, 8-, or 16-Word16 linear burst access to wrap within its burst-length boundary through GPMC_CONFIG1_i[24-23] ATTACHEDDEVICEPAGELENGTH.

9.1.3.3.10.2.3 Synchronous Single Write

Burst write mode is used for synchronous single or burst accesses (see Figure 9-21).

Figure 9-21. Synchronous Single Write on an Address/Data-Multiplexed Device



When the GPMC generates a write access to an address/data-multiplexed device, it drives the data bus (with address bits A[16:1]) until [19:16] WRDATAONADMUXBUS time. First data of the burst is driven on the address/data bus at WRDATAONADMUXBUS time.

9.1.3.3.10.2.4 Synchronous Multiple (Burst) Write

Synchronous burst write mode provides synchronous single or consecutive accesses. Figure 9-22 shows a synchronous burst write access when the chip-select is configured in address/data-multiplexed mode.

Figure 9-22. Synchronous Multiple Write (Burst Write) in Address/Data-Multiplexed Mode

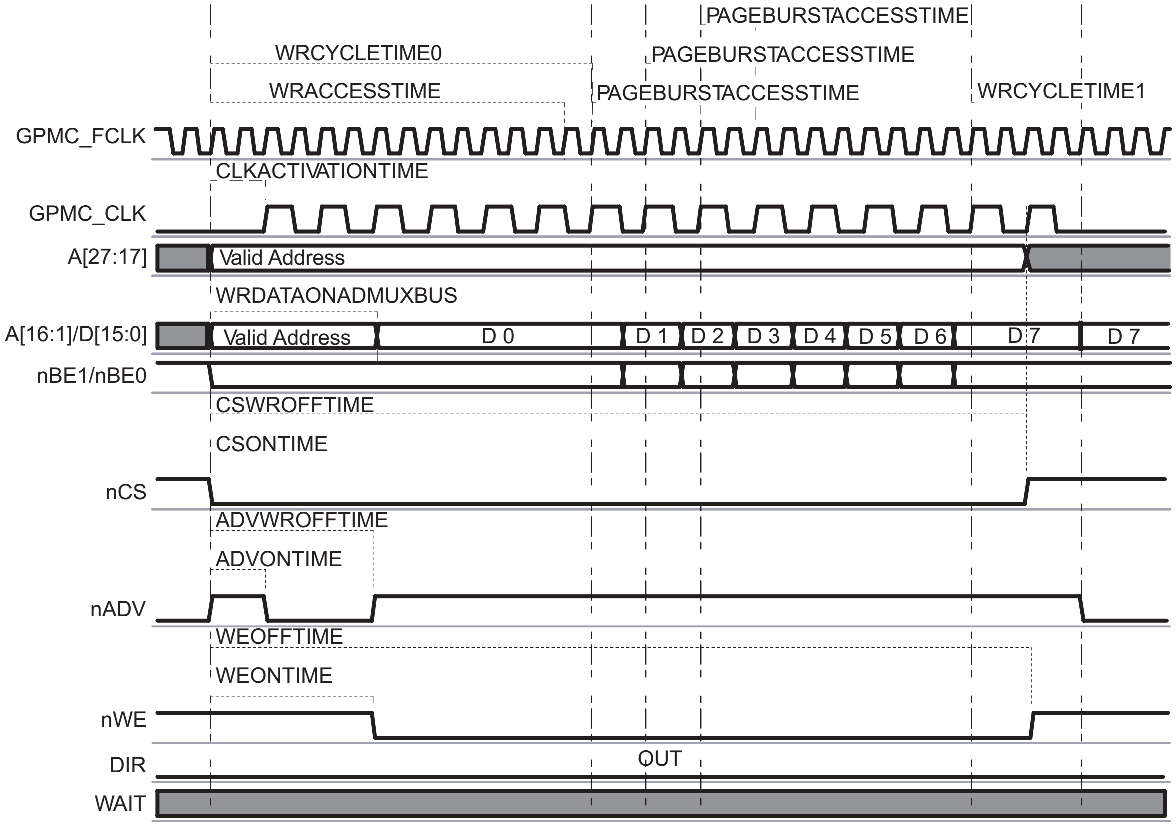
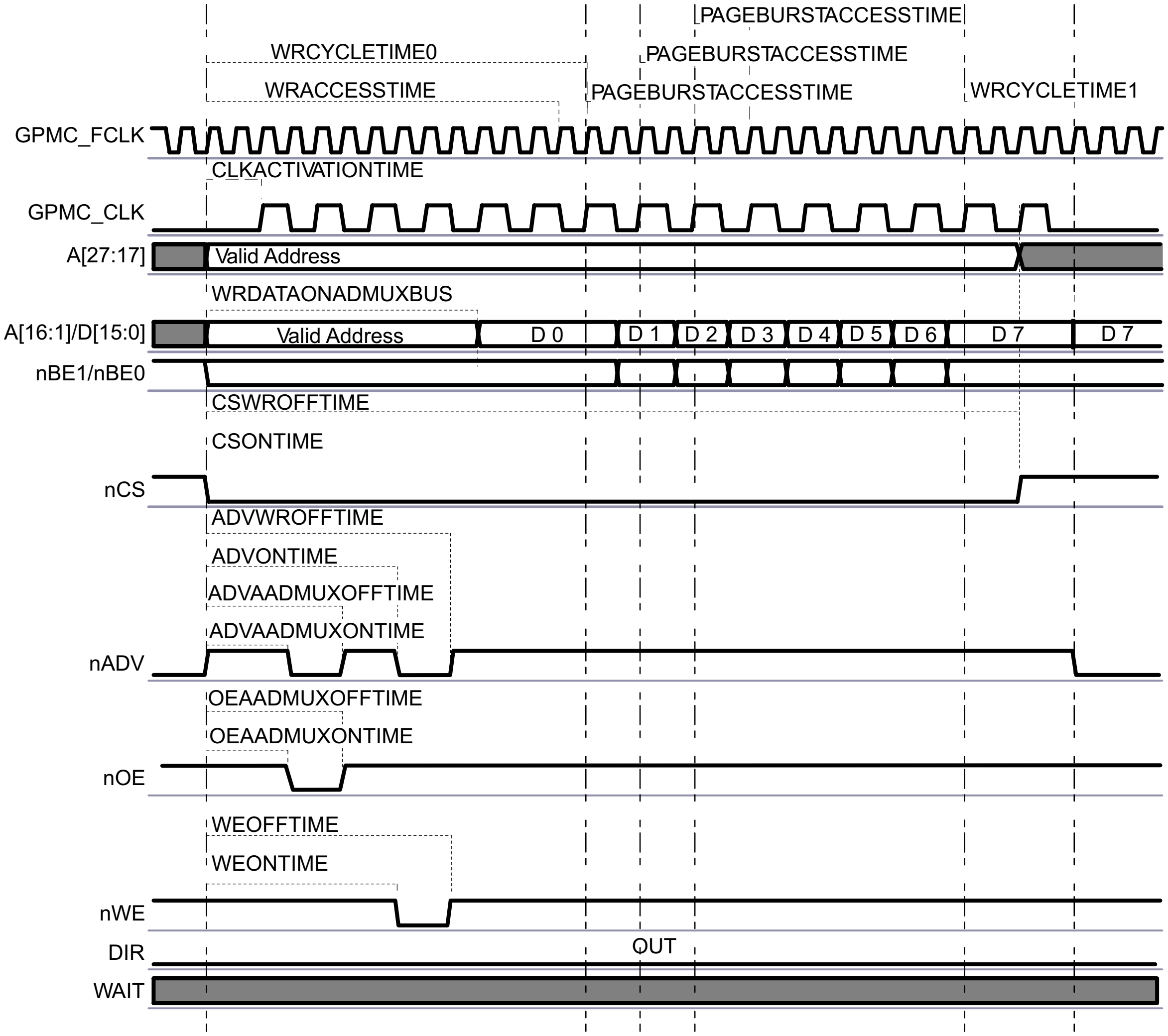


Figure 9-23 shows the same synchronous burst write access when the chip-select is configured in address/address/data-multiplexed (AAD-multiplexed) mode.

Figure 9-23. Synchronous Multiple Write (Burst Write) in Address/Address/Data-Multiplexed Mode



The first data of the burst is driven on the A/D bus at GPMC_CONFIG6_i[19:16] WRDATAONADMUXBUS.

When WRACCESSTIME completes, control-signal timings are frozen during the multiple data transactions, corresponding to the GPMC_CONFIG5_i[27-24] PAGEBURSTACCESSTIME multiplied by the number of remaining data transactions.

When the GPMC generates a read access to an address/data-multiplexed device, it drives the address bus until OEn assertion time. For details, see Section 9.1.3.3.8.2.3.

- Chip-select signal CSn
  - CSn assertion time is controlled by the GPMC_CONFIG2_i[3-0] CSONTIME field and ensures address setup time to CSn assertion.
  - CSn deassertion time controlled by the GPMC_CONFIG2_i[20-16] CSWROFFTIME field and ensures address hold time to CSn deassertion.
- Address valid signal ADVn
  - ADVn assertion time is controlled by the GPMC_CONFIG3_i[3-0] ADVONTIME field.
  - ADVn deassertion time is controlled by the GPMC_CONFIG3_i[20-16] ADVWROFFTIME field.

- Write enable signal WEn
  - WEn assertion indicates a read cycle.
  - WEn assertion time is controlled by the GPMC_CONFIG4_i[19-16] WEONTIME field.
  - WEn deassertion time is controlled by the GPMC_CONFIG4_i[28-24] WEOFFTIME field.

  The WEn falling edge must not be used to control the time when the burst first data is driven in the address/data bus because some new devices require the WEn signal at low during the address phase.
- Direction signal DIR is OUT during the entire access.

When the GPMC generates a write access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with OEn driven low. The second phase for LSB address is qualified with OEn driven high. The address phase ends at WEn assertion time.

The CSn and DIR signals are controlled as detailled above.

- Address valid signal ADVn is asserted and deasserted twice during a read transaction
  - ADVn first assertion time is controlled by the GPMC_CONFIG3_i[[6-4] ADVAADMUXONTIME field.
  - ADVn first deassertion time is controlled by the GPMC_CONFIG3_i[[26-24] ADVAADMUXRDOFFTIME field.
  - ADVn second assertion time is controlled by the GPMC_CONFIG3_i[[3-0] ADVONTIME field.
  - ADVn second deassertion time is controlled by the GPMC_CONFIG3_i[[12-8] ADVRDOFFTIME field.
- Output Enable signal OEn is asserted and deasserted twice during a read transaction (OEn second assertion indicates a read cycle)
  - OEn first assertion time is controlled by the GPMC_CONFIG4_i[[6-4] OEAADMUXONTIME field.
  - OEn first deassertion time is controlled by the GPMC_CONFIG4_i[15-13] OEAADMUXOFFTIME field.
  - OEn second assertion time is controlled by the GPMC_CONFIG4_i[3-0] OEONTIME field.
  - OEn second deassertion time is controlled by the GPMC_CONFIG4_i[12-8] OEOFFTIME field.

First write data is driven by the GPMC at GPMC_CONFIG6_i[19-16] WRDATAONADMUXBUS, when in address/data mux configuration. The next write data of the burst is driven on the bus at WRACCESSTIME + 1 during GPMC_CONFIG5_i[27-24] PAGEBURSTACCESSTIME GPMC_FCLK cycles. The last data of the synchronous burst write is driven until GPMC_CONFIG5_i[12-8] WRCYCLETIME completes.

- WRACCESSTIME is defined in the GPMC_CONFIG6_i[28-24] register.
- The PAGEBURSTACCESSTIME parameter must be set accordingly with GPMCFCLKDIVIDER and the memory-device internal configuration.

Total access time GPMC_CONFIG5_i[12-8] WRCYCLETIME corresponds to WRACCESSTIME plus the address hold time from CSn deassertion. In Figure 9-23 the WRCYCLETIME programmed value equals WRCYCLETIME0 + WRCYCLETIME1. WRCYCLETIME0 and WRCYCLETIME1 delays are not actual parameters and are only a graphical representation of the full WRCYCLETIME value.

After a write operation, if no other access (read or write) is pending, the data bus keeps the previous value. See Section 9.1.3.3.9.10.

9.1.3.3.10.3 Asynchronous and Synchronous Accesses in Nonmultiplexed Mode

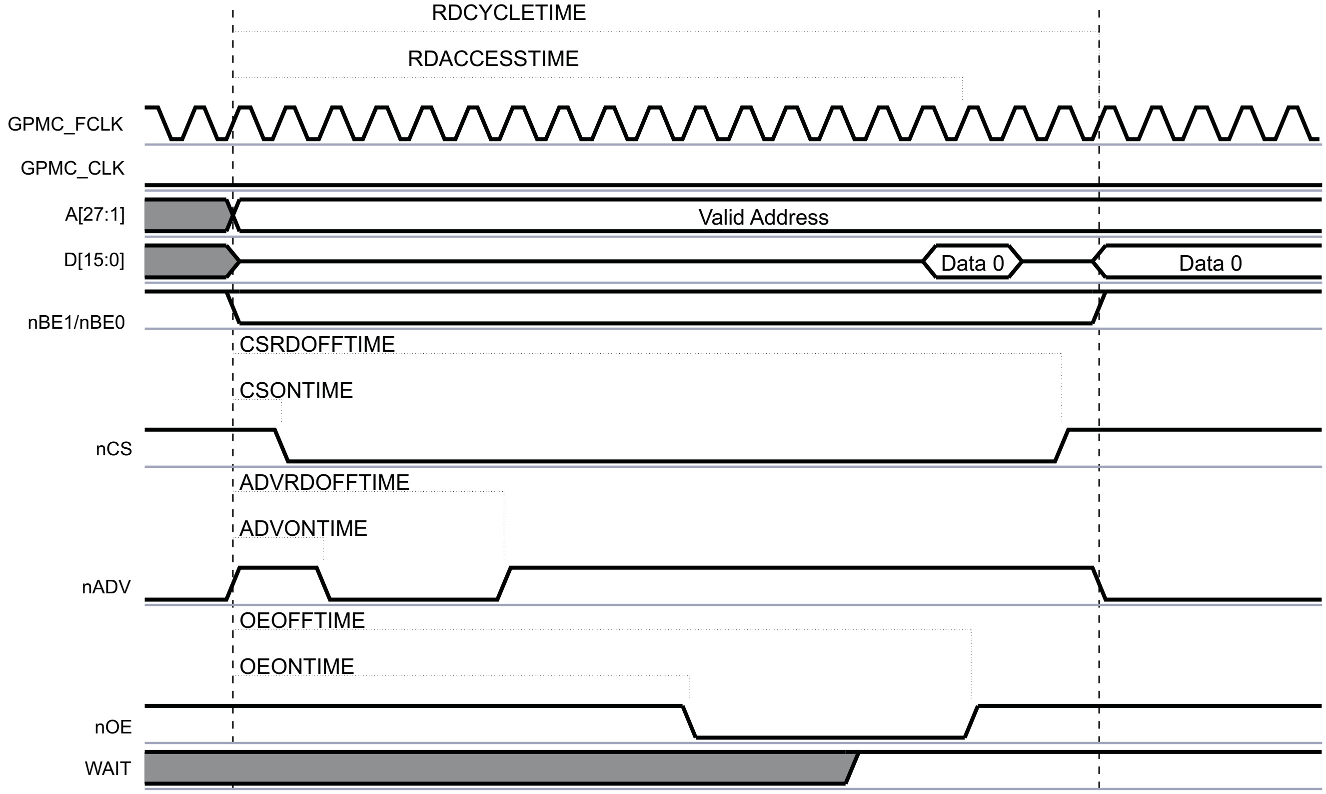
Page mode is only available in non-multiplexed mode.

- Asynchronous single read operation on a nonmultiplexed device
- Asynchronous single write operation on a nonmultiplexed device
- Asynchronous multiple (page mode) read operation on a nonmultiplexed device
- Synchronous operations on a nonmultiplexed device

9.1.3.3.10.3.1 Asynchronous Single Read Operation on a Nonmultiplexed Device

Figure 9-24 shows an asynchronous single read operation on a nonmultiplexed device.

Figure 9-24. Asynchronous Single Read on an Address/Data-Nonmultiplexed Device



The 27-bit address is driven onto the address bus A[27:1] and the 16-bit data is driven onto the data bus D[15:0].

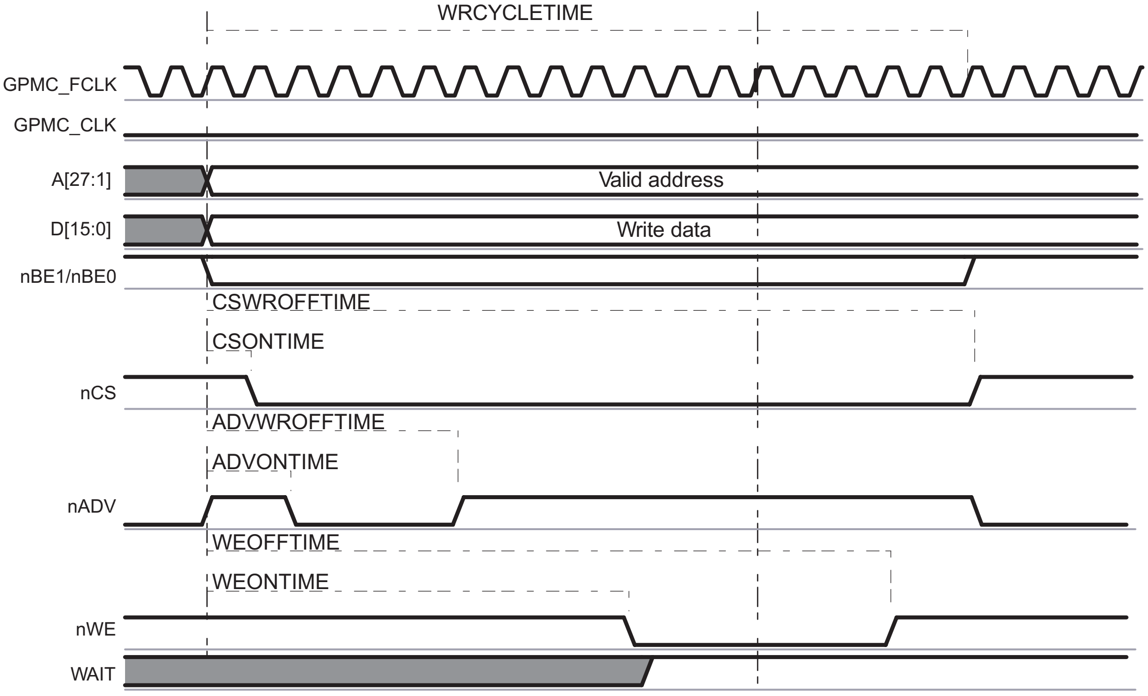
Read data is latched at GPMC_CONFIG1_5[20-16] RDACCESSTIME completion time. The end of the access is defined by the GPMC_CONFIG1_5[4-0] RDCYCLETIME parameter.

CSn, ADVn, OEn and DIR signals are controlled in the same way as address/data multiplexed accesses, see Section 9.1.3.3.10.1.1.2.

9.1.3.3.10.3.2 Asynchronous Single Write Operation on a Nonmultiplexed Device

Figure 9-25 shows an asynchronous single write operation on a nonmultiplexed device.

Figure 9-25. Asynchronous Single Write on an Address/Data-Nonmultiplexed Device



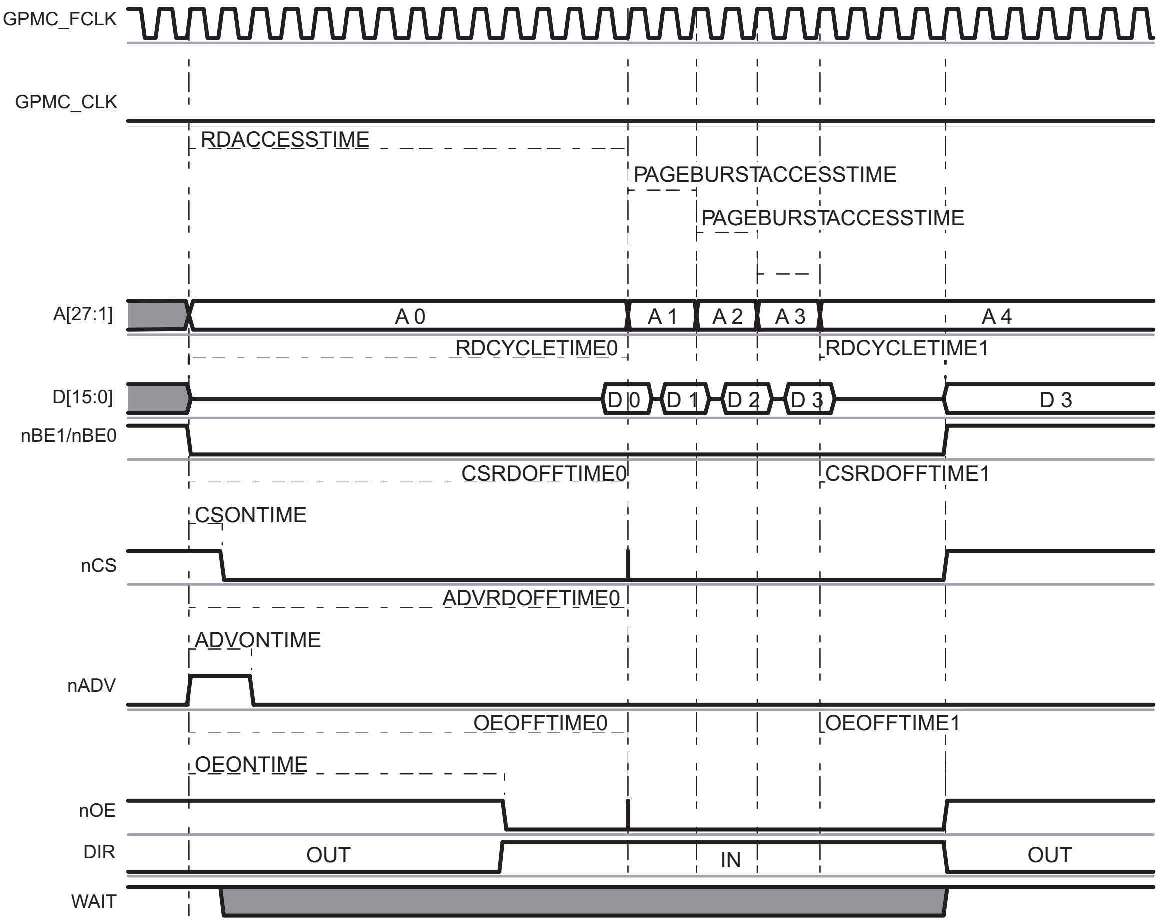
The 27-bit address is driven onto the address bus A[27:1] and the 16-bit data is driven onto the data bus D[15:0].

CSn, ADVn, WEn and DIR signals are controlled in the same way as address/data multiplexed accesses, see Section 9.1.3.3.10.1.1.3.

9.1.3.3.10.3.3 Asynchronous Multiple (Page Mode) Read Operation on a Nonmultiplexed Device

Figure 9-26 shows an asynchronous multiple read operation on a Nonmultiplexed Device, in which two word32 host read accesses to the GPMC are split into one multiple (page mode of 4 word16) read access to the attached device.

Figure 9-26. Asynchronous Multiple (Page Mode) Read



The WAIT signal is active low.

CSn, ADVn, OEn and DIR signals are controlled in the same way as address/data multiplexed accesses, see Section 9.1.3.3.10.1.1.2.

When RDACCESSTIME completes, control-signal timings are frozen during the multiple data transactions, corresponding to PAGEBURSTACCESSTIME multiplied by the number of remaining data transactions.

Read data is latched at GPMC_CONFIG5_i[20-16] RDACCESSTIME completion time. The end of the access is defined by the GPMC_CONFIG5_i[4-0] RDCYCLETIME parameter.

During consecutive accesses, the GPMC increments the address after each data read completes.

Delay between successive read data in the page is controlled by the GPMC_CONFIG5_i[27-24] PAGEBURSTACCESSTIME parameter. Depending on the device page length, the GPMC can control device page crossing during a burst request and insert initial RDACCESSTIME latency. Note that page crossing is only possible with a new burst access, meaning a new initial access phase is initiated.

Total access time RDCYCLETIME corresponds to RDACCESSTIME plus the address hold time starting from the CSn deassertion.

- The read cycle time is defined in the GPMC_CONFIG5_i[4-0] RDCYCLETIME field.
- In Figure 9-26, the RDCYCLETIME programmed value equals RDCYCLETIME0 (before paged accesses) + RDCYCLETIME1 (after paged accesses).

9.1.3.3.10.3.4 Synchronous Operations on a Nonmultiplexed Device

All information for this section is equivalent to similar operations for address/data- or AAD-multiplexed accesses. The only difference resides in the address phase. See Section 9.1.4.2.

9.1.3.3.10.4 Page and Burst Support

Each chip-select can be configured to process system single or burst requests into successive single accesses or asynchronous page/synchronous burst accesses, with appropriate access size adaptation.

Depending on the external device page or burst capability, read and write accesses can be independently configured through the GPMC. The GPMC_CONFIG1_i[30] READMULTIPLE and GPMC_CONFIG1_i[28] WRITMULTIPLE bits are associated with the READTYPE and WRITETYPE parameters.

- Asynchronous write page mode is not supported.
- 8-bit wide device support is limited to nonburstable devices (READMULTIPLE and WRITEMULTIPLE are ignored).
- Not applicable to NAND device interfacing.

9.1.3.3.10.5 System Burst Versus External Device Burst Support

The device system can issue the following requests to the GPMC:

- Byte, 16-bit word, 32-bit word requests (byte enable controlled). This is always a single request from the interconnect point of view.
- Incrementing fixed-length bursts of two words, four words, and eight words
- Wrapped (critical word access first) fixed-length burst of two, four, or eight words

To process a system request with the optimal protocol, the READMULTIPLE (and READTYPE) and WRITEMULTIPLE (and WRITETYPE) parameters must be set according to the burstable capability (synchronous or asynchronous) of the attached device.

The GPMC access engine issues only fixed-length burst. The maximum length that can be issued is defined per CS by the GPMC_CONFIG1_i[24-23] ATTACHEDDEVICEPAGELENGTH field. When the ATTACHEDDEVICEPAGELENGTH value is less than the system burst request length (including the appropriate access size adaptation according to the device width), the GPMC splits the system burst request into multiple bursts. Within the specified 4-, 8-, or 16-word value, the ATTACHEDDEVICEPAGELENGTH field value must correspond to the maximum-length burst supported by the memory device configured in fixed-length burst mode (as opposed to continuous burst mode).

To get optimal performance from memory devices that natively support 16 Word16-length-wrapping burst capability (critical word access first), the ATTACHEDDEVICEPAGELENGTH parameter must be set to 16 words and the GPMC_CONFIG1_i[31] WRAPBURST bit must be set to 1. Similarly DEVICEPAGELENGTH is set to 4 and 8 for memories supporting respectively 4 and 8 Word16-length-wrapping burst.

When the memory device does not offer (or is not configured to offer) native 16 Word16-length-wrapping burst, the WRAPBURST parameter must be cleared, and the GPMC access engine emulates the wrapping burst by issuing the appropriate burst sequences according to the ATTACHEDDEVICEPAGELENGTH value.

When the memory device does not support native-wrapping burst, there is usually no difference in behavior between a fixed burst length mode and a continuous burst mode configuration (except for a potential power increase from a memory-speculative data prefetch in a continuous burst read). However, even though continuous burst mode is compatible with GPMC behavior, because the GPMC access engine issues only fixed-length burst and does not benefit from continuous burst mode, it is best to configure the memory device in fixed-length burst mode.

The memory device maximum-length burst (configured in fixed-length burst wrap or nonwrap mode) usually corresponds to the memory device data buffer size. Memory devices with a minimum of 16 half-word buffers are the most appropriate (especially with wrap support), but memory devices with smaller buffer size (4 or 8) are also supported, assuming that the GPMC_CONFIG1_i[24-23] ATTACHEDDEVICEPAGELENGTH field is set accordingly to 4 or 8 words.

The device system issues only requests with addresses or starting addresses for nonwrapping burst requests; that is, the request size boundary is aligned. In case of an eight-word-wrapping burst, the wrapping address always occurs on the eight-words boundary. As a consequence, all words requested must be available from the memory data buffer when the buffer size is equal to or greater than the ATTACHEDDEVICEPAGELENGTH value. This usually means that data can be read from or written to the buffer at a constant rate (number of cycles between data) without wait states between data accesses. If the memory does not behave this way (nonzero wait state burstable memory), wait-pin monitoring must be enabled to dynamically control data-access completion within the burst.

When the system burst request length is less than the ATTACHEDDEVICEPAGELENGTH value, the GPMC proceeds with the required accesses.

9.1.3.3.11 pSRAM Access Specificities

pSRAM devices are SRAM-pin-compatible low-power memories that contain a self-refreshed DRAM memory array. The GPMC_CONFIG1_i[[11-10] DEVICETYPE field shall be cleared to 0b00.

The pSRAM devices uses the NOR protocol. It support the following operations:

- Asynchronous single read
- Asynchronous page read
- Asynchronous single write
- Synchronous single read and write
- Synchronous burst read
- Synchronous burst write (not supported by NOR Flash memory)

pSRAM devices must be powered up and initialized in a predefined manner according to the specifications of the attached device.

pSRAM devices can be programmed to use either mode: fixed or variable latency. pSRAM devices can either automatically schedule autorefresh operations, which force the GPMC to use its WAIT signal capability when read or write operations occur during an internal self-refresh operation, or pSRAM devices automatically include the autorefresh operation in the access time. These devices do not require additional WAIT signal capability or a minimum CSn high pulse width between consecutive accesses to ensure that the correct internal refresh operation is scheduled.

9.1.3.3.12 NAND Access Description

NAND (8-bit and 16-bit) memory devices using a standard NAND asynchronous address/data-multiplexing scheme can be supported on any chip-select with the appropriate asynchronous configuration settings

As for any other type of memory compatible with the GPMC interface, accesses to a chip-select allocated to a NAND device can be interleaved with accesses to chip-selects allocated to other external devices. This interleaved capability limits the system to *chip enable don't care* NAND devices, because the chip-select allocated to the NAND device must be de-asserted if accesses to other chip-selects are requested.

9.1.3.3.12.1 NAND Memory Device in Byte or 16-Bit Word Stream Mode

NAND devices require correct command and address programming before data array read or write accesses. The GPMC does not include specific hardware to translate a random address system request into a NAND-specific multiphase access. In that sense, GPMC NAND support, as opposed to random memory-map device support, is data-stream-oriented (byte or 16-bit word).

The GPMC NAND programming model relies on a software driver for address and command formatting with the correct data address pointer value according to the block and page structure. Because of NAND structure and protocol interface diversity, the GPMC does not support automatic command and address phase programming, and software drivers must access the NAND device ID to ensure that correct command and address formatting are used for the identified device.

NAND device data read and write accesses are achieved through an asynchronous read or write access. The associated chip-select signal timing control must be programmed according to the NAND device timing specification.

Any chip-select region can be qualified as a NAND region to constrain the ADVn_ALE signal as Address Latch Enable (ALE active high, default state value at low) during address program access, and the BE0n_CLE signal as Command Latch Enable (CLE active high, default state value at low) during command program access. GPMC address lines are not used (the previous value is not changed) during NAND access.

9.1.3.3.12.1.1 Chip-Select Configuration for NAND Interfacing in Byte or Word Stream Mode

The GPMC_CONFIG7_i register associated with a NAND device region interfaced in byte or word stream mode can be initialized with a minimum size of 16 Mbytes, because any address location in the chip-select memory region can be used to access a NAND data array. The NAND Flash protocol specifies an address sequence where address bits are passed through the data bus in a series of write accesses with the ALE pin asserted. After this address phase, all operations are streamed and the system requests address is irrelevant.

To allow correct command, address, and data-access controls, the GPMC_CONFIG1_i register associated with a NAND device region must be initialized in asynchronous read and write modes with the parameters shown in Table 9-11. Failure to comply with these settings corrupts the NAND interface protocol.

The GPMC_CONFIG1_i to GPMC_CONFIG4_i register associated with a NAND device region must be initialized with the correct control-signal timing value according to the NAND device timing parameters.

Table 9-11. Chip-Select Configuration for NAND Interfacing

Bit Field	Register	Value	Comments
WRAPBURST	GPMC_CONFIG1_i	0	No wrap
READMULTIPLE	GPMC_CONFIG1_i	0	Single access
READTYPE	GPMC_CONFIG1_i	0	Asynchronous mode
WRITEMULTIPLE	GPMC_CONFIG1_i	0	Single access
WRITETYPE	GPMC_CONFIG1_i	0	Asynchronous mode
CLKACTIVATIONTIME	GPMC_CONFIG1_i	0b00	
ATTACHEDDEVICEPAGELENGTH	GPMC_CONFIG1_i	Don't care	Single-access mode
WAITREADMONITORING	GPMC_CONFIG1_i	0	Wait not monitored by GPMC access engine
WAITWRITEMONITORING	GPMC_CONFIG1_i	0	Wait not monitored by GPMC access engine
WAITMONITORINGTIME	GPMC_CONFIG1_i	Don't care	Wait not monitored by GPMC access engine
WAITPINSELECT	GPMC_CONFIG1_i		Select which wait is monitored by edge detectors
DEVICESIZE	GPMC_CONFIG1_i	0b00 or 0b01	8- or 16-bit interface
DEVICETYPE	GPMC_CONFIG1_i	0b10	NAND device in stream mode
MUXADDDATA	GPMC_CONFIG1_i	0b00	Nonmultiplexed mode
TIMEPARAGRANULARITY	GPMC_CONFIG1_i	0	Timing achieved with best GPMC clock granularity
GPMCFCLKDIVIDER	GPMC_CONFIG1_i	Don't care	Asynchronous mode

9.1.3.3.12.1.2 NAND Device Command and Address Phase Control

NAND devices require multiple address programming phases. The MPU software driver is responsible for issuing the correct number of command and address program accesses, according to the device command set and the device address-mapping scheme.

NAND device-command and address-phase programming is achieved through write requests to the GPMC_NAND_COMMAND_i and GPMC_NAND_ADDR_i register locations with the correct command and address values. These locations are mapped in the associated chip-select register region. The associated chip-select signal timing control must be programmed according to the NAND device timing specification.

Command and address values are not latched during the access and cannot be read back at the register location.

- Only write accesses must be issued to these locations, but the GPMC does not discard any read access. Accessing a NAND device with OEn and CLE or ALE asserted (read access) can produce undefined results.
- Write accesses to the GPMC_NAND_COMMAND_i register location and to the GPMC_NAND_ADDR_i register location must be posted for faster operations. The GPMC_CONFIG[0] NANDFORCEPOSTEDWRITE bit enables write accesses to these locations as posted, even if they are defined as nonposted.

A write buffer is used to store write transaction information before the external device is accessed:

- Up to eight consecutive posted write accesses can be accepted and stored in the write buffer.
- For nonposted write, the pipeline is one deep.
- A GPMC_STS[0] EMPTYWRITEBUFFERSTS bit stores the empty status of the write buffer.

The GPMC_NAND_COMMAND_i and GPMC_NAND_ADDR_i registers are 32-bit word locations, which means any 32-bit word or 16-bit word access is split into 4- or 2-byte accesses if an 8-bit wide NAND device is attached. For multiple-command phase or multiple-address phase, the software driver can use 32-bit word or 16-bit word access to these registers, but it must account for the splitting and little-endian ordering scheme. When only one byte command or address phase is required, only byte write access to a GPMC_NAND_COMMAND_i and GPMC_NAND_ADDR_i can be used, and any of the four byte locations of the registers are valid.

The same applies to GPMC_NAND_COMMAND_i and GPMC_NAND_ADDR_i 32-bit word write access to a 16-bit wide NAND device (split into two 16-bit word accesses). In the case of a 16-bit word write access, the MSByte of the 16-bit word value must be set according to the NAND device requirement (usually 0). Either 16-bit word location or any one of the four byte locations of the registers is valid

9.1.3.3.12.1.3 Command Latch Cycle

Writing data at the GPMC_NAND_COMMAND_i location places the data as the NAND command value on the bus, using a regular asynchronous write access.

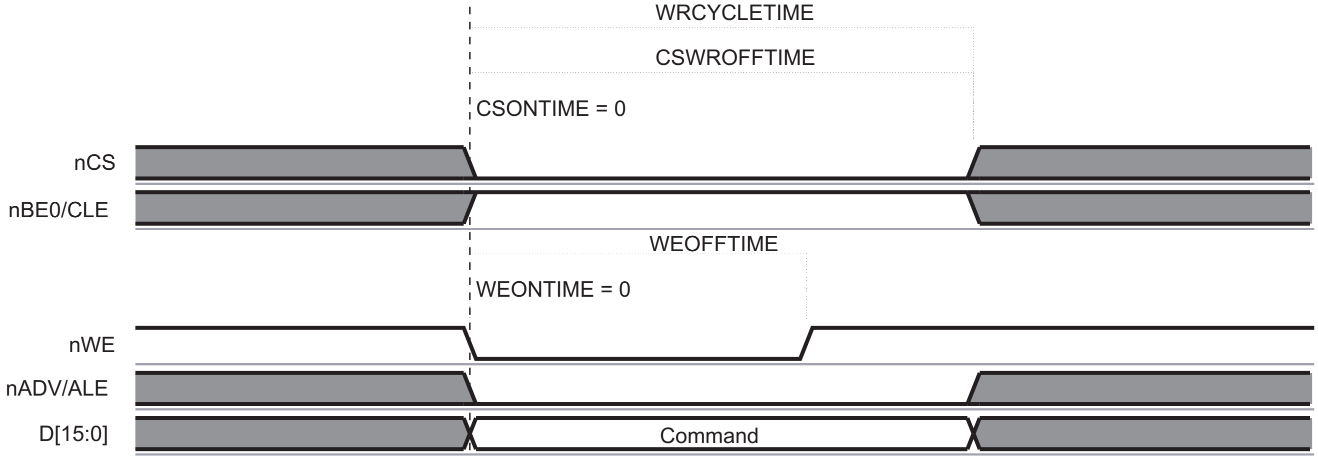
- CSn[i] is controlled by the CSONTIME and CSWROFFTIME timing parameters.
- CLE is controlled by the ADVONTIME and ADVWROFFTIME timing parameters.
- WE is controlled by the WEONTIME and WEOFFTIME timing parameters.
- ALE and REn (OEn) are maintained inactive.

Figure 9-27 shows the NAND command latch cycle.

CLE is shared with the BE0n output signal and has an inverted polarity from BE0n. The NAND qualifier deals with this. During the asynchronous NAND data access cycle, BE0n (also BE1n) must not toggle, because it is shared with CLE.

NAND Flash memories do not use byte enable signals at all.

Figure 9-27. NAND Command Latch Cycle



9.1.3.3.12.1.4 Address Latch Cycle

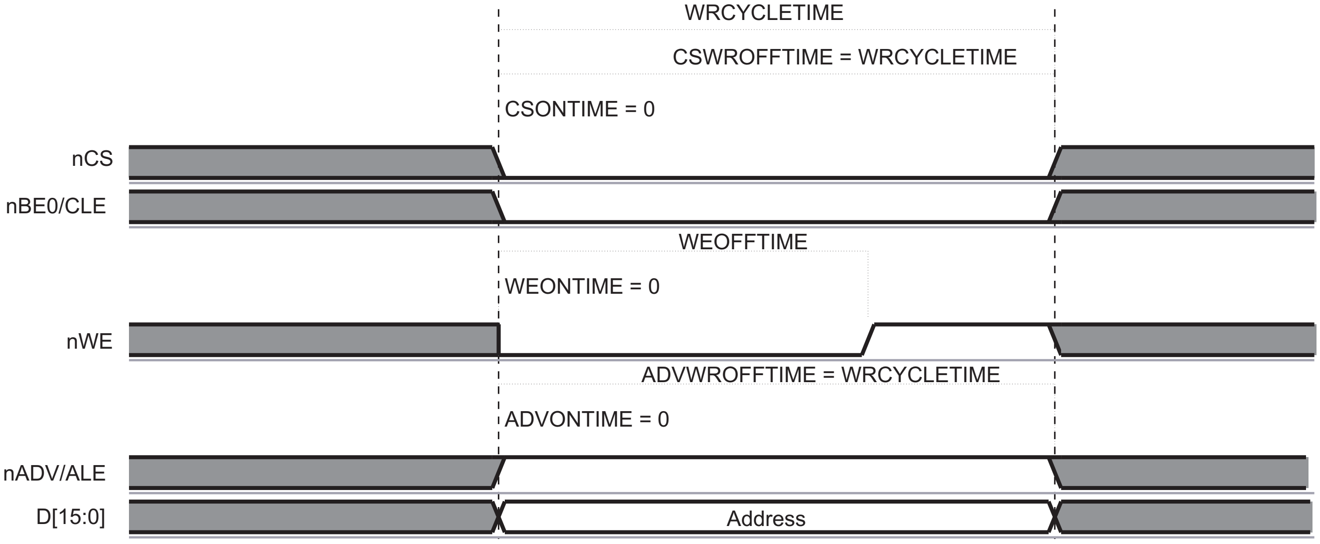
Writing data at the GPMC_NAND_ADDR_i location places the data as the NAND partial address value on the bus, using a regular asynchronous write access.

- CSn is controlled by the CSONTIME and CSWROFFTIME timing parameters.
- ALE is controlled by the ADVONTIME and ADVWROFFTIME timing parameters.
- WEn is controlled by the WEONTIME and WEOFFTIME timing parameters.
- CLE and REn (OEn) are maintained inactive.

Figure 9-28 shows the NAND address latch cycle.

ALE is shared with the ADVn output signal and has an inverted polarity from ADVn. The NAND qualifier deals with this. During the asynchronous NAND data access cycle, ALE is kept stable.

Figure 9-28. NAND Address Latch Cycle



9.1.3.3.12.1.5 NAND Device Data Read and Write Phase Control in Stream Mode

NAND device data read and write accesses are achieved through a read or write request to the chip-select-associated memory region at any address location in the region or through a read or write request to the GPMC_NAND_DATA_i location mapped in the chip-select-associated control register region. GPMC_NAND_DATA_i is not a true register, but an address location to enable REn or WEn signal control. The associated chip-select signal timing control must be programmed according to the NAND device timing specification.

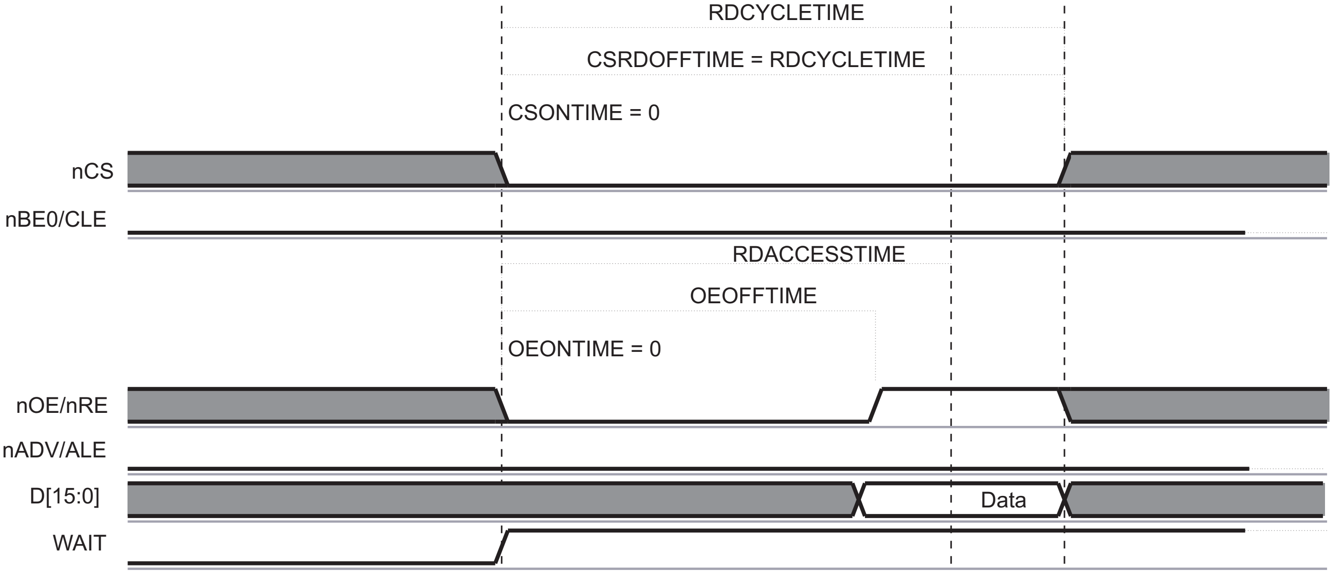
Reading data from the GPMC_NAND_DATA_i location or from any location in the associated chip-select memory region activates an asynchronous read access.

- CSn is controlled by the CSONTIME and CSRDOFFTIME timing parameters.
- REn is controlled by the OEONTIME and OEOFFTIME timing parameters.
- To take advantage of REn high-to-data invalid minimum timing value, the RDACCESSTIME can be set so that data are effectively captured after REn deassertion. This allows optimization of NAND read access cycle time completion. For optimal timing parameter settings, see the NAND device and the device IC timing parameters.

ALE, CLE, and WEn are maintained inactive.

Figure 9-29 shows the NAND data read cycle.

Figure 9-29. NAND Data Read Cycle

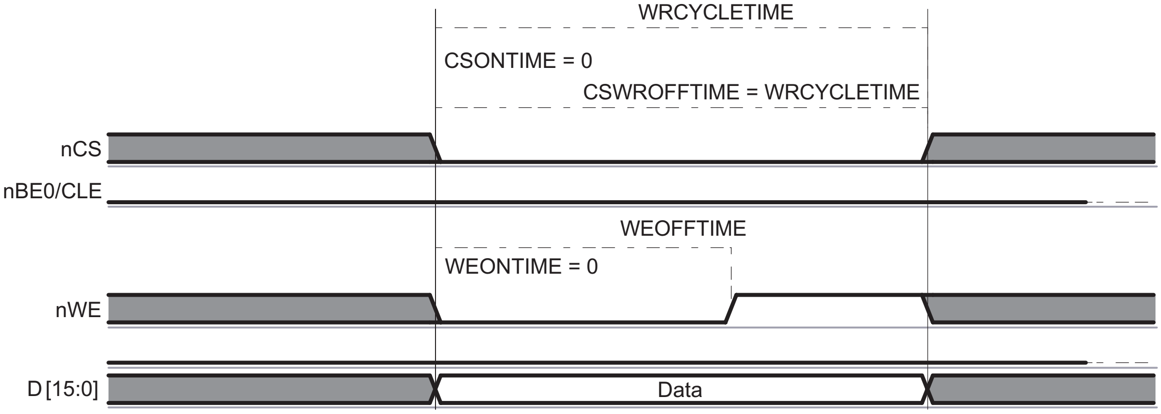


Writing data to the GPMC_NAND_DATA_i location or to any location in the associated chip-select memory region activates an asynchronous write access.

- CSn is controlled by the CSONTIME and CSWROFFTIME timing parameters.
- WEn is controlled by the WEONTIME and WEOFFTIME timing parameters.
- ALE, CLE, and REn (OEn) are maintained inactive.

Figure 9-30 shows the NAND data write cycle.

Figure 9-30. NAND Data Write Cycle



9.1.3.3.12.1.6 *NAND Device General Chip-Select Timing Control Requirement*

For most NAND devices, read data access time is dominated by CSn-to-data-valid timing and has faster REn-to-data-valid timing. Successive accesses with CSn deassertions between accesses are affected by this timing constraint. Because accesses to a NAND device can be interleaved with other chip-select accesses, there is no certainty that CSn always stays low between two accesses to the same chip-select. Moreover, an CSn deassertion time between the same chip-select NAND accesses is likely to be required as follows: the CSn deassertion requires programming CYCLETIME and RDACCESSTIME according to the CSn-to-data-valid critical timing.

To get full performance from NAND read and write accesses, the prefetch engine can dynamically reduce RDCYCLETIME, WRCYCLETIME, RDACCESSTIME, WRACCESSTIME, CSRDOFFTIME, CSWROFFTIME, ADVRDOFFTIME, ADVWROFFTIME, OEOFFTIME, and WEOFFTIME on back-to-back NAND accesses (to the same memory) and suppress the minimum CSn high pulse width between accesses. For more information about optimal prefetch engine access, see Section 9.1.3.3.12.4.

Some NAND devices require minimum write-to-read idle time, especially for device-status read accesses following status-read command programming (write access). If such write-to-read transactions are used, a minimum CSn high pulse width must be set. For this, CYCLE2CYCLESAMECSEN and CYCLE2CYCLEDELAY must be set according to the appropriate timing requirement to prevent any timing violation.

NAND devices usually have an important REn high to data bus in tristate mode. This requires a bus turnaround setting (BUSTURNAROUND = 1), so that the next access to a different chip-select is delayed until the BUSTURNAROUND delay completes. Back-to-back NAND read accesses to the same NAND Flash are not affected by the programmed bus turnaround delay.

9.1.3.3.12.1.7 *Read and Write Access Size Adaptation*

9.1.3.3.12.1.7.1 *8-Bit Wide NAND Device*

Host 16-bit word and 32-bit word read and write access requests to a chip-select associated with an 8-bit wide NAND device are split into successive read and write byte accesses to the NAND memory device. Byte access is ordered according to little-endian organization. A NAND 8-bit wide device must be interfaced on the D0D7 interface bus lane. GPMC data accesses are justified on this bus lane when the chip-select is associated with an 8-bit wide NAND device.

9.1.3.3.12.1.7.2 *16-Bit Wide NAND Device*

Host 32-bit word read and write access requests to a chip-select associated with a 16-bit wide NAND device are split into successive read and write 16-bit word accesses to the NAND memory device. 16-bit word access is ordered according to little-endian organization.

Host byte read and write access requests to a 16-bit wide NAND device are completed as 16-bit accesses on the device itself, because there is no byte-addressing capability on 16-bit wide NAND devices. This means that the NAND device address pointer is incremented on a 16-bit word basis and not on a byte basis. For a read access, only the requested byte is given back to the host, but the remaining byte is not stored or saved by the GPMC, and the next byte or 16-bit word read access gets the next 16-bit word NAND location. For a write access, the invalid byte part of the 16-bit word is driven to FF, and the next byte or 16-bit word write access programs the next 16-bit word NAND location.

Generally, byte access to a 16-bit wide NAND device should be avoided, especially when ECC calculation is enabled. 8-bit or 16-bit ECC-based computations are corrupted by a byte read to a 16-bit wide NAND device, because the nonrequested byte is considered invalid on a read access (not captured on the external data bus; FF is fed to the ECC engine) and is set to FF on a write access.

Host requests (read/write) issued in the chip-select memory region are translated in successive single or split accesses (read/write) to the attached device. Therefore, incrementing 32-bit burst requests are translated in multiple 32-bit sequential accesses following the access adaptation of the 32-bit to 8- or 16-bit device.

9.1.3.3.12.2 NAND Device-Ready Pin

The NAND memory device provides a ready pin to indicate data availability after a block/page opening and to indicate that data programming is complete. The ready pin can be connected to one of the WAIT GPMC input pins; data read accesses must not be tried when the ready pin is sampled inactive (device is not ready) even if the associated chip-select WAITREADMONITORING bit field is set. The duration of the NAND device busy state after the block/page opening is so long (up to 50 µs) that accesses occurring when the ready pin is sampled inactive can stall GPMC access and eventually cause a system time-out.

If a read access to a NAND flash is done using the wait monitoring mode, the device is blocked during a page opening, and so is the GPMC. If the correct settings are used, other chip-selects can be used while the memory processes the page opening command.

To avoid a time-out caused by a block/page opening delay in NAND flash, disable the wait pin monitoring for read and write accesses (that is, set the GPMC_CONFIG1_i[[21] WAITWRITEMONITORING and GPMC_CONFIG1_i[[22] WAITREADMONITORING bits to 0 and use one of the following methods instead:

- Use software to poll the WAITnSTS bit (n = 0 to 1) of the GPMC_STS register.
- Configure an interrupt that is generated on the WAIT signal change (through the GPMC_IRQEN [11-8]bits).

Even if the READWAITMONITORING bit is not set, the external memory nR/B pin status is captured in the programmed WAIT bit in the GPMC_STS register.

The READWAITMONITORING bit method must be used for other memories than NAND flash, if they require the use of a WAIT signal.

9.1.3.3.12.2.1 Ready Pin Monitored by Software Polling

The ready signal state can be monitored through the GPMC_STS WAITxSTS bit (x = 0 or 1). The software must monitor the ready pin only when the signal is declared valid. Refer to the NAND device timing parameters to set the correct software temporization to monitor ready only after the invalid window is complete from the last read command written to the NAND device.

9.1.3.3.12.2.2 Ready Pin Monitored by Hardware Interrupt

Each gpmc_wait input pin can generate an interrupt when a wait-to-no-wait transition is detected. Depending on whether the GPMC_CONFIG WAITxPINPOLARITY bits (x = 0 or 1) is active low or active high, the wait-to-no-wait transition is a low-to-high external WAIT signal transition or a high-to-low external WAIT signal transition, respectively.

The wait transition pin detector must be cleared before any transition detection. This is done by writing 1 to the WAITxEDGEDETECTIONSTS bit (x = 0 or 1) of the GPMC_IRQSTS register according to the gpmc_wait pin used for the NAND device-ready signal monitoring. To detect a wait-to-no-wait transition, the transition detector requires a wait active time detection of a minimum of two GPMC_FCLK cycles. Software must incorporate precautions to clear the wait transition pin detector before wait (busy) time completes.

A wait-to-no-wait transition detection can issue a GPMC interrupt if the WAITxEDGEDETECTIONENABLE bit in the GPMC_IRQEN register is set and if the WAITxEDGEDETECTIONSTS bit field in the GPMC_IRQSTS register is set.

The WAITMONITORINGTIME field does not affect wait-to-no-wait transition time detection.

It is also possible to poll the WAITxEDGEDETECTIONSTS bit field in the GPMC_IRQSTS register according to the gpmc_wait pin used for the NAND device ready signal monitoring.

9.1.3.3.12.3 ECC Calculator

The General Purpose Memory Controller includes an Error Code Correction (ECC) calculator circuitry that enables on the fly ECC calculation during data read or data program (that is, write) operations. The page size supported by the ECC calculator in one calculation/context is 512 bytes.

The user can choose from two different algorithms with different error correction capabilities through the GPMC_ECC_CONFIG[16] ECCALGORITHM bit:

- Hamming code for 1-bit error code correction on 8- or 16-bit NAND Flash organized with page size greater than 512 bytes
- BCH (Bose-Chaudhurl-Hocquenghem) code for 4- to 16-bit error correction

The GPMC does not directly handle the error code correction itself. During writes, the GPMC computes parity bits. During reads, the GPMC provides enough information for the processor to correct errors without reading the data buffer all over again.

The Hamming code ECC is based on a 2-dimensional (row and column) bit parity accumulation. This parity accumulation is either accomplished on the programmed number of bytes or 16-bit words read from the memory device, or written to the memory device in stream mode.

Because the ECC engine includes only one accumulation context, it can be allocated to only one chip-select at a time through the GPMC_ECC_CONFIG[3-1] ECCCS bit field. Even if two CS use different ECC algorithms, one the Hamming code and the other a BCH code, they must define separate ECC contexts because some of the ECC registers are common to all types of algorithms.

9.1.3.3.12.3.1 Hamming Code

All references to Error Code Correction (ECC) in this subsection refer to the 1-bit error correction Hamming code.

The ECC is based on a two-dimensional (row and column) bit parity accumulation known as Hamming Code. The parity accumulation is done for a programmed number of bytes or 16-bit word read from the memory device or written to the memory device in stream mode.

There is no automatic error detection or correction, and it is the software NAND driver responsibility to read the multiple ECC calculation results, compare them to the expected code value, and take the appropriate corrective actions according to the error handling strategy (ECC storage in spare byte, error correction on read, block invalidation).

The ECC engine includes a single accumulation context. It can be allocated to a single designated chip-select at a time and parallel computations on different chip-selects are not possible. Since it is allocated to a single chip-select, the ECC computation is not affected by interleaved GPMC accesses to other chip-selects and devices. The ECC accumulation is sequentially processed in the order of data read from or written to the memory on the designated chip-select. The ECC engine does not differentiate read accesses from write accesses and does not differentiate data from command or status information. It is the software responsibility to make sure only relevant data are passed to the NAND flash memory while the ECC computation engine is active.

The starting NAND page location must be programmed first, followed by an ECC accumulation context reset with an ECC enabling, if required. The NAND device accesses discussed in the following sections must be limited to data read or write until the specified number of ECC calculations is completed.

9.1.3.3.12.3.1.1 ECC Result Register and ECC Computation Accumulation Size

The GPMC includes up to nine ECC result registers (GPMC_ECCj_RESULT, j = 1 to 9) to store ECC computation results when the specified number of bytes or 16-bit words has been computed.

The ECC result registers are used sequentially; one ECC result is stored in one ECC result register on the list, the next ECC result is stored in the next ECC result register on the list, and so forth, until the last ECC computation. The value of the GPMC_ECCj_RESULT register value is valid only when the programmed number of bytes or 16-bit words has been accumulated, which means that the same number of bytes or 16-bit words has been read from or written to the NAND device in sequence.

The GPMC_ECC_CTRL[3-0] ECCPOINTER field must be set to the correct value to select the ECC result register to be used first in the list for the incoming ECC computation process. The ECCPointer can be read to determine which ECC register is used in the next ECC result storage for the ongoing ECC computation. The value of the GPMC_ECCj_RESULT register (j = 1 to 9) can be considered valid when ECCPOINTER equals j + 1. When the GPMC_ECCj_RESULT (where j = 9) is updated, ECCPOINTER is frozen at 10, and ECC computing is stopped (ECCENABLE = 0).

The ECC accumulator must be reset before any ECC computation accumulation process. The GPMC_ECC_CTRL[8] ECCCLR bit must be set to 1 (nonpersistent bit) to clear the accumulator and all ECC result registers.

For each ECC result (each register, j = 1 to 9), the number of bytes or 16-bit words used for ECC computing accumulation can be selected from between two programmable values.

The ECCjRESULTSIZE bits (j = 1 to 9) in the GPMC_ECC_SIZE_CONFIG register select which programmable size value (ECCSIZE0 or ECCSIZE1) must be used for this ECC result (stored in GPMC_ECCj_RESULT register).

The ECCSIZE0 and ECCSIZE1 fields allow selection of the number of bytes or 16-bit words used for ECC computation accumulation. Any even values from 2 to 512 are allowed.

Flexibility in the number of ECCs computed and the number of bytes or 16-bit words used in the successive ECC computations enables different NAND page error-correction strategies. Usually based on 256 or 512 bytes and on 128 or 256 16-bit word, the number of ECC results required is a function of the NAND device page size. Specific ECC accumulation size can be used when computing the ECC on the NAND spare byte.

For example, with a 2 Kbyte data page 8-bit wide NAND device, eight ECCs accumulated on 256 bytes can be computed and added to one extra ECC computed on the 24 spare bytes area where the eight ECC results used for comparison and correction with the computed data page ECC are stored. The GPMC then provides nine GPMC_ECCj_RESULT registers (j= 1 to 9) to store the results. In this case, ECCSIZE0 is set to 256, and ECCSIZE1 is set to 24; the ECC[1-8]RESULTSIZE bits are cleared to 0, and the ECC9RESULTSIZE bit is set to 1.

9.1.3.3.12.3.1.2 ECC Enabling

The GPMC_ECC_CONFIG[3-1] ECCCS field selects the allocated chip-select. The GPMC_ECC_CONFIG[0] ECCENABLE bit enables ECC computation on the next detected read or write access to the selected chip-select.

The ECCPOINTER, ECCCLR, ECCSIZE, ECCjRESULTSIZE (where j = 1 to 9), ECC16B, and ECCCS fields must not be changed or cleared while an ECC computation is in progress.

The ECC accumulator and ECC result register must not be changed or cleared while an ECC computation is in progress.

Table 9-12 describes the ECC enable settings.

Table 9-12. ECC Enable Settings

Bit Field	Register	Value	Comments
ECCCS	GPMC_ECC_CONFIG	0-3h	Selects the chip-select where ECC is computed
ECC16B	GPMC_ECC_CONFIG	0/1	Selects column number for ECC calculation
ECCCLR	GPMC_ECC_CTRL	0-7h	Clears all ECC result registers
ECCPOINTER	GPMC_ECC_CTRL	0-7h	A write to this bit field selects the ECC result register where the first ECC computation is stored. Set to 1 by default.
ECCSIZE1	GPMC_ECC_SIZE_CONFIG	0-FFh	Defines ECCSIZE1
ECCSIZE0	GPMC_ECC_SIZE_CONFIG	0-FFh	Defines ECCSIZE0
ECCjRESULTSIZE (j from 1 to 9)	GPMC_ECC_SIZE_CONFIG	0/1	Selects the size of ECCn result register
ECCENABLE	GPMC_ECC_CONFIG	1	Enables the ECC computation

9.1.3.3.12.3.1.3 ECC Computation

The ECC algorithm is a multiple parity bit accumulation computed on the odd and even bit streams extracted from the byte or Word 16 streams. The parity accumulation is split into row and column accumulations, as shown in Figure 9-31 and Figure 9-32. The intermediate row and column parities are used to compute the upper level row and column parities. Only the final computation of each parity bit is used for ECC comparison and correction.

P1o = bit7 XOR bit5 XOR bit3 XOR bit1 on each byte of the data stream

P1e = bit6 XOR bit4 XOR bit2 XOR bit0 on each byte of the data stream

P2o = bit7 XOR bit6 XOR bit3 XOR bit2 on each byte of the data stream

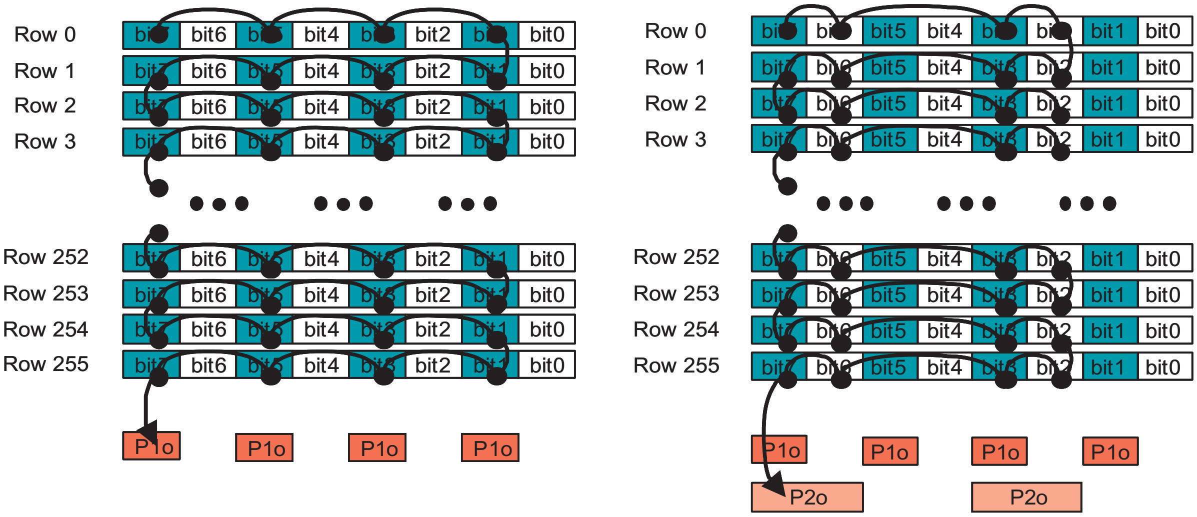
P2e = bit5 XOR bit4 XOR bit1 XOR bit0 on each byte of the data stream

P4o = bit7 XOR bit6 XOR bit5 XOR bit4 on each byte of the data stream

P4e = bit3 XOR bit2 XOR bit1 XOR bit0 on each byte of the data stream

Each column parity bit is XORed with the previous accumulated value.

Figure 9-31. Hamming Code Accumulation Algorithm (1 of 2)



For line parities, the bits of each new data are XORed together, and line parity bits are computed as:

P8e = row0 XOR row2 XOR row4 XOR ... XOR row254

P8o = row1 XOR row3 XOR row5 XOR ... XOR row255

P16e = row0 XOR row1 XOR row4 XOR row5 XOR ... XOR row252 XOR row 253

P16o = row2 XOR row3 XOR row6 XOR row7 XOR ... XOR row254 XOR row 255

Unused parity bits in the result registers are cleared to 0.

Figure 9-32. Hamming Code Accumulation Algorithm (2 of 2)

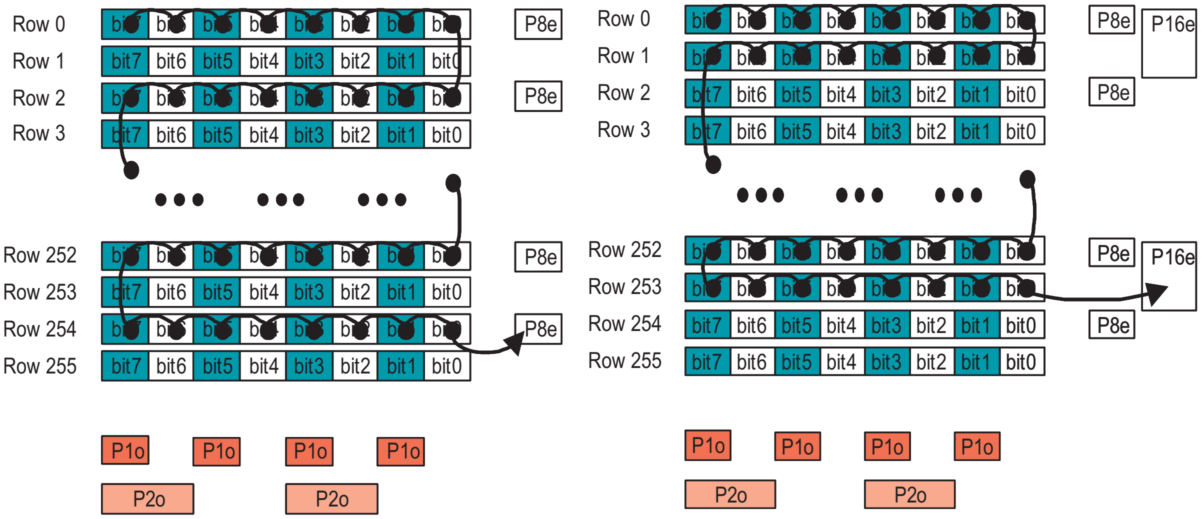


Figure 9-33 shows ECC computation for a 256-byte data stream (read or write). The result includes six column parity bits (P1o-P2o-P4o for odd parities, and P1e-P2e-P4e for even parities) and sixteen row parity bits (P8o-P16o-P32o--P1024o for odd parities, and P8e-P16e-P32e--P1024e for even parities).

Figure 9-33. ECC Computation for a 256-Byte Data Stream (Read or Write)

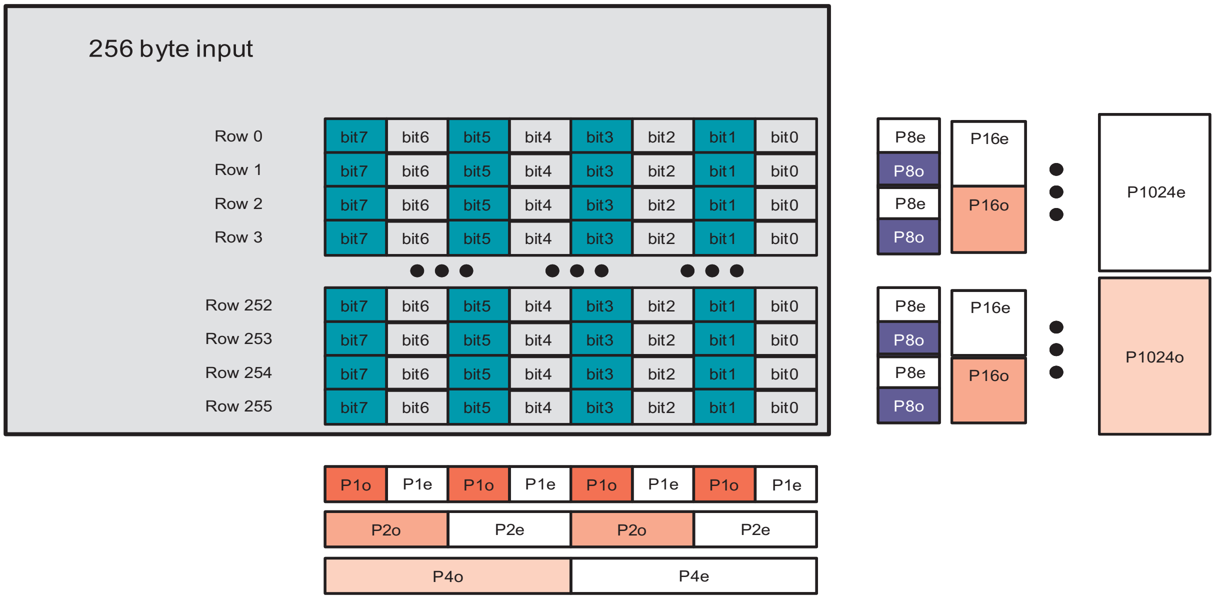
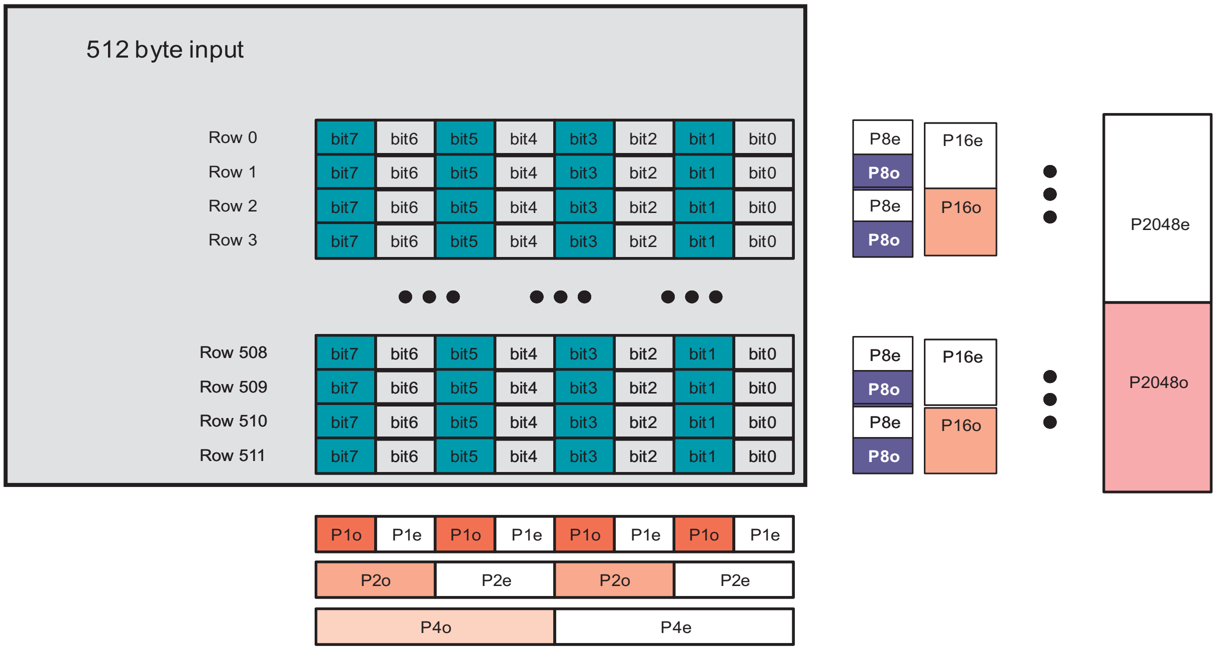


Figure 9-34 shows ECC computation for a 512-byte data stream (read or write). The result includes six column parity bits (P1o-P2o-P4o for odd parities, and P1e-P2e-P4e for even parities) and eighteen row parity bits (P8o-P16o-P32o--P1024o- - P2048o for odd parities, and P8e-P16e-P32e--P1024e- P2048e for even parities).

For a 2 Kbytes page, four 512 bytes ECC calculations plus one for the spare area are required. Results are stored in the GPMC_ECCj_RESULT registers (j = 1 to 9).

Figure 9-34. ECC Computation for a 512-Byte Data Stream (Read or Write)



9.1.3.3.12.3.1.4 ECC Comparison and Correction

To detect an error, the computed ECC result must be XORed with the parity value stored in the spare area of the accessed page.

- If the result of this logical XOR is all 0s, no error is detected and the read data is correct.
- If every second bit in the parity result is a 1, one bit is corrupted and is located at bit address (P2048o, P1024o, P512o, P256o, P128o, P64o, P32o, P16o, P8o, P4o, P2o, P1o). The software must correct the corresponding bit.
- If only one bit in the parity result is 1, it is an ECC error and the read data is correct.

9.1.3.3.12.3.1.5 ECC Calculation Based on 8-Bit Word

The 8-bit based ECC computation is used for 8-bit wide NAND device interfacing.

The 8-bit based ECC computation can be used for 16-bit wide NAND device interfacing to get backward compatibility on the error-handling strategy used with 8-bit wide NAND devices. In this case, the 16-bit wide data read from or written to the NAND device is fragmented into 2 bytes. According to little-endian access, the least significant bit (LSB) of the 16-bit wide data is ordered first in the byte stream used for 8-bit based ECC computation.

9.1.3.3.12.3.1.6 ECC Calculation Based on 16-Bit Word

ECC computation based on a 16-bit word is used for 16-bit wide NAND device interfacing. This ECC computation is not supported when interfacing an 8-bit wide NAND device, and the GPMC_ECC_CONFIG[7] ECC16B bit must be cleared to 0 when interfacing an 8-bit wide NAND device.

The parity computation based on 16-bit words affects the row and column parity mapping. The main difference is that the odd and even parity bits P8o and P8e are computed on rows for an 8-bit based ECC while there are computed on columns for a 16-bit based ECC. Figure 9-35 and Figure 9-36.

Figure 9-35. 128 Word16 ECC Computation

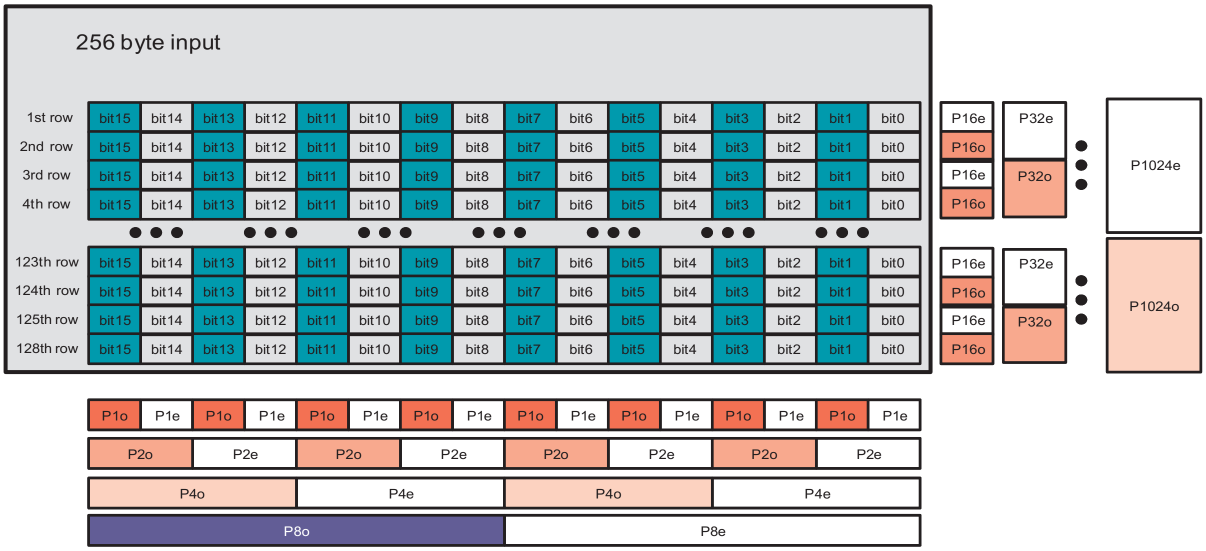
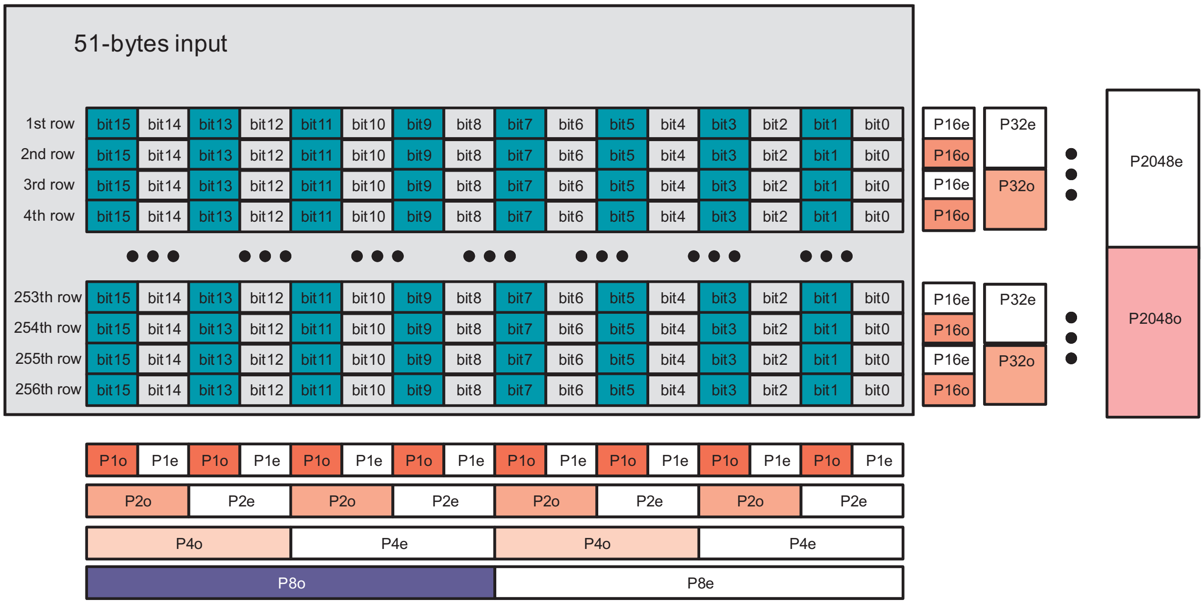


Figure 9-36. 256 Word16 ECC Computation



9.1.3.3.12.3.2 BCH Code (Bose-Chaudhurl-Hocquenghem)

All references to Error Code Correction (ECC) in this subsection refer to the 4- to 16-bit error correction BCH code.

9.1.3.3.12.3.2.1 Requirements

Read and write accesses to a NAND flash take place by whole pages, in a predetermined sequence: first the data byte page itself, then some spare bytes, including the BCH ECC (and other information). The NAND IC can cache a full page, including spares, for read and write accesses.

Typical page write sequence:

- Sequential write to NAND cache of main data + spare data, for a page. ECC is calculated on the fly. Calculated ECC may be inserted on the fly in the spares, or replaced by dummy accesses.
- When the calculated ECC is replaced by dummy accesses, it must be written to the cache in a second, separate phase. The ECC module is disabled during that time.
- NAND writes its cache line (page) to the array

Typical page read sequence:

- Sequential read of a page. ECC is calculated on the fly.
- ECC module buffers status determines the presence of errors.
- Accesses to several memories may be interleaved by the GPMC, but only one of those memories can be a NAND using the BCH engine at a time; in other words, only one BCH calculation (for example, for a single page) can be on-going at any time. Note also that the sequential nature of NAND accesses guarantees that the data is always written / read out in the same order. BCH-relevant accesses are selected by the GPMCs chip-select.
- Each page may hold up to 4 Kbytes of data, spare bytes not included. This means up to 8 x 512-byte BCH messages. Since all the data is written / read out first, followed by the BCH ECC, this means that the BCH engine must be able to hold 8 104-bit remainders or syndromes (or smaller, 52-bit ones) at the same time.

The BCH module has the capacity to store all remainders internally. After the page start, an internal counter is used to detect the 512-byte sector boundaries. On those boundaries, the current remainder is stored and the divider reset for the next calculation. At the end of the page, the BCH module contains all remainders.

- NAND access cycles hold 8 or 16 bits of data each (1 or 2 bytes); Each NAND cycle takes at least 4 cycles of the GPMCs internal clock. This means the NAND flash timing parameters must define a RDCYCLETIME and a WRCYCLETIME of at least 4 clock cycles after optimization when using the BCH calculator.
- The spare area is assumed to be large enough to hold the BCH ECC, that is, to have at least a message of 13 bytes available per 512-byte sector of data. The zone of unused spare area by the ECC may or may not be protected by the same ECC scheme, by extending the BCH message beyond 512 bytes (maximum codeword is 1023-byte long, ECC included, which leaves a lot of space to cover some spares bytes).

9.1.3.3.12.3.2.2 Memory-Mapping of the BCH Codeword

BCH encoding considers a block of data to protect as a polynomial message M(x). In our standard case, 512 bytes of data (that is, 2 bits = 4096 bits) are seen as a polynomial of degree 2 - 1 = 4095, with parameters ranging from M0 to M4095. For 512 bytes of data, 52 bits are required for 4-bit error correction, and 104 bits are required for 8-bit error correction and 207 bits are required for 16-bit error correction. The ECC is a remainder polynomial R(x) of degree 103 (or 51, depending on the selected mode). The complete codeword C(x) is the concatenation of M(x) and R(x) as shown in Table 9-13.

Table 9-13. Flattened BCH Codeword Mapping (512 Bytes + 104 Bits)

	Message M(x)	ECC R(x)				
Bit number	M4095	...	M0	R103	...	R0

If the message is extended by the addition of spare bytes to be protected by the same ECC, the principle is still valid. For example, a 3-byte extension of the message gives a polynomial message M(x) of degree ((512 + 3) × 8) - 1 = 4119, for a total of 3 + 13 = 16 spare bytes of spare, all protected as part of the same codeword.

The message and the ECC bits are manipulated and mapped in the GPMC byte-oriented system. The ECC bits are stored in:

- GPMC_BCH_RESULT0_i
- GPMC_BCH_RESULT1_i
- GPMC_BCH_RESULT2_i
- GPMC_BCH_RESULT3_i

9.1.3.3.12.3.2.3 Memory Mapping of the Data Message

The data message mapping shall follow the following rules:

- Bit endianness within a byte is little-endian, that is, the bytes LS bit is also the lowest-degree polynomial parameter: a byte b7-b0 (with b0 the LS bit) represents a segment of polynomial b7 * x + b6 * x + ... + b0 * x
- The message is mapped in the NAND starting with the highest-order parameters, that is, in the lowest addresses of a NAND page.
- Byte endianness within the NANDs 16-bit words is big endian. This means that the same message mapped in 8- and 16-bit memories has the same content at the same byte address.

The BCH module has no visibility over actual addresses. The most important point is the sequence of data word the BCH sees. However, the NAND page is always scanned incrementally in read and write accesses, and this produces the mapping patterns described in the following.

Table 9-14 and Table 9-15 show the mapping of the same 512-byte vector (typically a BCH message) in the NAND memory space. Note that the byte 'address' is only an offset modulo 512 (200h), since the same page may contain several contiguous 512-byte sectors (BCH blocks). The LSB and MSB are respectively the bits M0 and M(2^12-1) of the codeword mapping given above. In both cases the data vectors are aligned, that is, their boundaries coincide with the RAMs data word boundaries.

Table 9-14. Aligned Message Byte Mapping in 8-bit NAND

Byte Offset	8-Bit Word
0	(msb) Byte 511 (1FFh)
1h	Byte 510 (1FEh)
⋮	⋮
1FFh	Byte 0 (0) (LSB)

Table 9-15. Aligned Message Byte Mapping in 16-bit NAND

Byte Offset	16-Bit Words MSB	16-Bit Words LSB
0	Byte 510 (1FEh)	(msb) Byte 511 (1FFh)
2h	Byte 508 (1FCh)	Byte 509 (1FDh)
⋮	⋮	⋮
1FEh	Byte 0 (0)	(lsb) Byte 1 (1)

Table 9-16 and Table 9-17 show the mapping in memory of arbitrarily-sized messages, starting on access (byte or 16-bit word) boundaries for more clarity. Note that message may actually start and stop on arbitrary nibbles. A nibble is a 4-bit entity. The unused nibbles are not discarded, and they can still be used by the BCH module, but as part of the next message section (for example, on another sectors ECC).

Table 9-16. Aligned Nibble Mapping of Message in 8-bit NAND

Byte Offset	8-Bit Word	
	4-Bit Most Significant Nibble	4-Bit Less Significant Nibble
1	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-3	Nibble S-4
⋮	⋮	⋮
S/2 - 2	Nibble 3	Nibble 2
S/2 - 1	Nibble 1	Nibble 0 (LSB)

Table 9-17. Misaligned Nibble Mapping of Message in 8-bit NAND

Byte Offset	8-Bit Word	
	4-Bit Most Significant Nibble	4-Bit Less Significant Nibble
1	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-3	Nibble S-4
⋮	⋮	⋮
(S+1)/2 - 2	Nibble 2	Nibble 1
(S+1)/2 - 1	Nibble 0 (LSB)	

Table 9-18. Aligned Nibble Mapping of Message in 16-bit NAND

Byte Offset	16-Bit Word			
	4-Bit Most Significant Nibble		4-Bit Less Significant Nibble	
0	Nibble S-3	Nibble S-4	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
⋮	⋮	⋮	⋮	⋮
S/2 - 4	Nibble 5	Nibble 4	Nibble 7	Nibble 6
S/2 - 2	Nibble 1	Nibble 0 (LSB)	Nibble 3	Nibble 2

Table 9-19. Misaligned Nibble Mapping of Message in 16-bit NAND (1 Unused Nibble)

Byte Offset	16-Bit Word			
	4-Bit Most Significant Nibble		4-Bit Less Significant Nibble	
0	Nibble S-3	Nibble S-4	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
⋮	⋮	⋮	⋮	⋮
(S+1)/2 - 4	Nibble 4	Nibble 3	Nibble 6	Nibble 5
(S+1)/2 - 2	Nibble 0 (LSB)		Nibble 2	Nibble 1

Table 9-20. Misaligned Nibble Mapping of Message in 16-bit NAND (2 Unused Nibble)

Byte Offset	16-Bit Word			
	4-Bit Most Significant Nibble		4-Bit Less Significant Nibble	
0	Nibble S-3	Nibble S-4	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
⋮	⋮	⋮	⋮	⋮
(S+2)/2 - 4	Nibble 3	Nibble 2	Nibble 5	Nibble 4
(S+2)/2 - 2			Nibble 1	Nibble 0 (LSB)

Table 9-21. Misaligned Nibble Mapping of Message in 16-bit NAND (3 Unused Nibble)

Byte Offset	16-Bit Word			
	4-Bit Most Significant Nibble		4-Bit Less Significant Nibble	
0	Nibble S-3	Nibble S-4	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
⋮	⋮	⋮	⋮	⋮
(S+3)/2 - 4	Nibble 2	Nibble 1	Nibble 4	Nibble 3
(S+3)/2 - 2			Nibble 0 (LSB)	

Note that many other cases exist than the ones represented above, for example, where the message does not start on a word boundary.

9.1.3.3.12.3.2.4 Memory Mapping of the ECC

The ECC (or remainder) is presented by the BCH module as a single 104-bit (or 52-bit), little-endian vector. It is up to the software to fetch those 13 bytes (or 6 bytes) from the modules interface, then store them to the NANDs spare area (page write) or to an intermediate buffer for comparison with the stored ECC (page read). There are no constraints on the ECC mapping inside the spare area: it is a softwarecontrolled operation.

However, it is advised to maintain a coherence in the respective formats of the message or the ECC remainder once they have been read out of the NAND. The error correction algorithm works from the complete codeword (concatenated message and remainder) once an error as been detected. The creation of this codeword should be made as straightforward as possible.

There are cases where the same NAND access contains both data and the ECC protecting that data. This is the case when the data/ECC boundary (which can be on any nibble) does not coincide with an access boundary. The ECC is calculated on-the-fly following the write. In that case, the write must also contain part of the ECC because it is impossible to insert the ECC on-the-fly. Instead:

- During the initial page write (BCH encoding), the ECC is replaced by dummy bits. The BCH encoder is by definition turned OFF during the ECC section, so the BCH result is unmodified.
- During a second phase, the ECC is written to the correct location, next to the actual data.
- The completed line buffer is then written to the NAND array.

9.1.3.3.12.3.2.5 Wrapping Modes

For a given wrapping mode, the module automatically goes through a specific number of sections, as data is being fed into the module. For each section, the BCH core can be enabled (in which case the data is fed to the BCH divider) or not (in which case the BCH simply counts to the end of the section). When enabled, the data is added to the ongoing calculation for a given sector number (for example, number 0).

Wrapping modes are described below. To get a better understanding and see the real-life read and write sequences implemented with each mode, see Section 9.1.3.3.12.3.3.

For each mode:

- A sequence describes the mode in pseudo-language, with for each section the size and the buffer used for ECC processing (if ON). The programmable lengths are size, size0 and size1.
- A checksum condition is given. If the checksum condition is not respected for a given mode, the modules behavior is unpredictable. S is the number of sectors in the page; size0 and size1 are the section sizes programmed for the mode, in nibbles.

Note that wrapping modes 8, 9, 10, and 11 insert a 1-nibble padding where the BCH processing is OFF. This is intended for t = 4 ECC, where ECC is 6 bytes long and the ECC area is expected to include (at least) 1 unused nibble to remain byte-aligned.

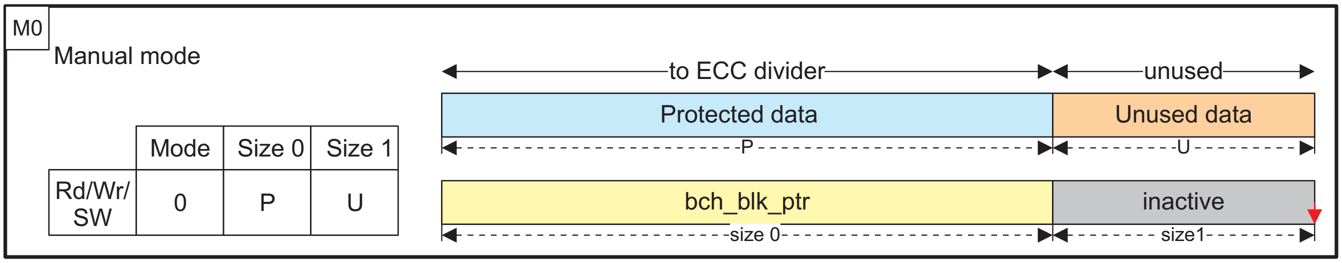
9.1.3.3.12.3.2.6 Manual Mode (0x0)

This mode is intended for short sequences, added manually to a given buffer through the software data port input. A complete page may be built out of several such sequences.

To process an arbitrary sequence of 4-bit nibbles, accesses to the software data port shall be made, containing the appropriate data. If the sequence end does not coincide with an access boundary (for example, to process 5 nibbles = 20 bits in 16-bit access mode) and those nibbles need to be skipped, a number of unused nibbles shall be programmed in size1 (in the same example: 5 nibbles to process + 3 to discard = 8 nibbles = exactly 2 x 16-bit accesses: we must program size0 = 5, size1 = 3).

Figure 9-37 shows the manual mode sequence and mapping. In this figure, size and size0 are the same parameter.

Figure 9-37. Manual Mode Sequence and Mapping



	Mode	Size 0	Size 1
Rd/Wr/ SW	0	P	U

Section processing sequence:

- One time with buffer
  - size0 nibbles of data, processing ON
  - size1 nibbles of unused data, processing OFF

Checksum: size0 + size1 nibbles must fit in a whole number of accesses.

In the following sections, S is the number of sectors in the page.

9.1.3.3.12.3.2.7 Mode 0x1

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = S - (size0 + size1)

9.1.3.3.12.3.2.8 Mode 0xA (10)

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
  - 1 nibble pad spare, processing OFF
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = S - (size0 + 1 + size1)

9.1.3.3.12.3.2.9 Mode 0x2

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = S - (size0 + size1)

9.1.3.3.12.3.2.10 Mode 0x3

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time with buffer 0
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S - size1)

9.1.3.3.12.3.2.11 Mode 0x7

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time with buffer 0
  - size0 nibbles spare, processing ON
- Repeat S times (no buffer used)
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = size0 + (S - size1)

9.1.3.3.12.3.2.12 Mode 0x8

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time with buffer 0
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - 1 nibble padding spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S - (1+size1))

9.1.3.3.12.3.2.13 Mode 0x4

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time (no buffer used)
  - size0 nibbles spare, processing OFF
- Repeat with buffer 0 to S-1
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S - size1)

9.1.3.3.12.3.2.14 Mode 0x9

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time (no buffer used)
  - size0 nibbles spare, processing OFF
- Repeat with buffer 0 to S-1
  - 1 nibble padding spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S - (1+size1))

9.1.3.3.12.3.2.15 Mode 0x5

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = S - (size0 + size1)

9.1.3.3.12.3.2.16 Mode 0xB (11)

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - 1 nibble padding spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = S - (size0 + 1 + size1)

9.1.3.3.12.3.2.17 Mode 0x6

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
- Repeat S times (no buffer used)
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = S - (size0 + size1)

9.1.3.3.12.3.3 Supported NAND Page Mappings and ECC Schemes

The following rules apply throughout the entire mapping description:

- Main data area (sectors) size is hardcoded to 512 bytes.
- Spare area size is programmable.
- All page sections (of main area data bytes, protected spare bytes, unprotected spare bytes, and ECC) are defined as explained in Section 9.1.3.3.12.3.2.3.

Each one of the following sections shows a NAND page mapping example (per-sector spare mappings, pooled spare mapping, per-sector spare mapping, with ECC separated at the end of the page).

In the mapping diagrams, sections that belong to the same BCH codeword have the same color (blue or green); unprotected sections are not covered (orange) by the BCH scheme.

Below each mapping diagram, a write (encoding) and read (decoding: syndrome generation) sequence is given, with the number of the active buffers at each point in time (yellow). In the inactive zones (grey), no computing is taking place but the data counter is still active.

In Figure 9-38 to Figure 9-40, tables on the left summarize the mode, size0, size1 parameters to program for respectively write and read processing of a page, with the given mapping, where:

- P is the size of spare byte section Protected by the ECC (in nibbles)
- U is the size of spare byte section Unprotected by the ECC (in nibbles)
- E is the size of the ECC itself (in nibbles)
- S is the number of Sectors per page (2 in the current diagrams)

Each time the processing of a BCH block is complete (ECC calculation for write/encoding, syndrome generation for read/decoding, indicated by red arrows), the update pointer is pulsed. Note that the processing for block 0 can be the first or the last to complete, depending on the NAND page mapping and operation (read or write). All examples show a page size of 1kByte + spares, that is, S = 2 sectors of 512 bytes. The same principles can be extended to larger pages by adding more sectors.

The actual BCH codeword size is used during the error location work to restrict the search range: by definition, errors can only happen in the codeword that was actually written to the NAND, and not in the mathematical codeword of n = 2 - 1 = 8191 bits. That codeword (higher-order bits) is all-zero and implicit during computations.

The actual BCH codeword size depends on the mode, on the programmed sizes and on the sector number (all sizes in nibbles):

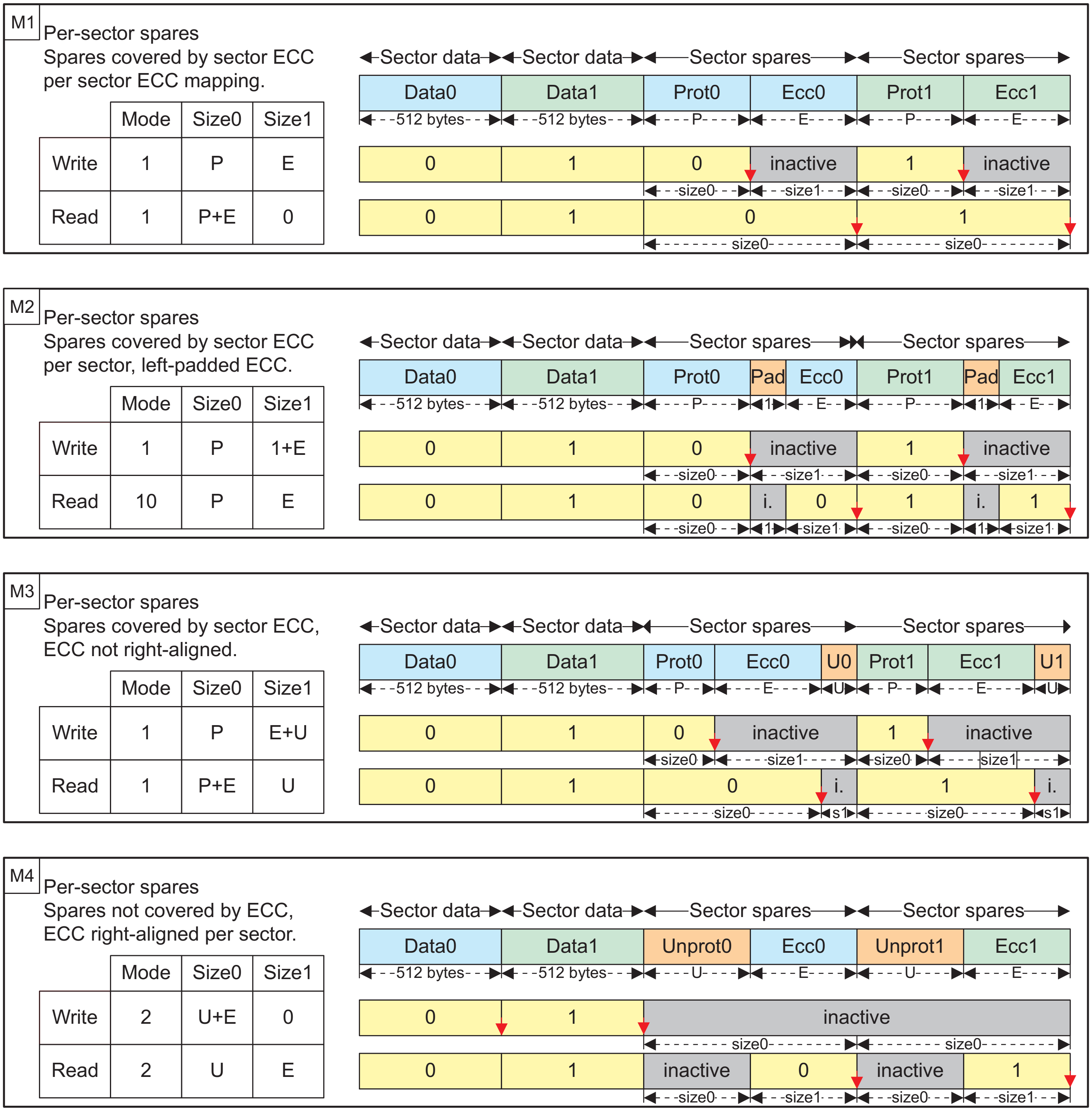
- Spares mapped and protected per sector (Figure 9-38: see M1-M2-M3-M9-M10):
  - all sectors: (512) + P + E
- Spares pooled and protected by sector 0 (Figure 9-38: see M5-M6):
  - sector 0 codeword: (512) + P + E
  - other sectors: (512) + E
- Unprotected spares (Figure 9-38: see M4-M7-M8-M11-M12):
  - all codewords (512) + E

9.1.3.3.12.3.3.1 Per-Sector Spare Mappings

In these schemes (Figure 9-38), each 512-byte sector of the main area has its own dedicated section of the spare area. The spare area of each sector is composed of:

- ECC, which must be located after the data it protects
- other data, which may or may not be protected by the sectors ECC

Figure 9-38. NAND Page Mapping and ECC: Per-Sector Schemes



M1

	Mode	Size0	Size1
Write	1	P	E
Read	1	P+E	0

M2

	Mode	Size0	Size1
Write	1	P	1+E
Read	10	P	E

M3

	Mode	Size0	Size1
Write	1	P	E+U
Read	1	P+E	U

M4

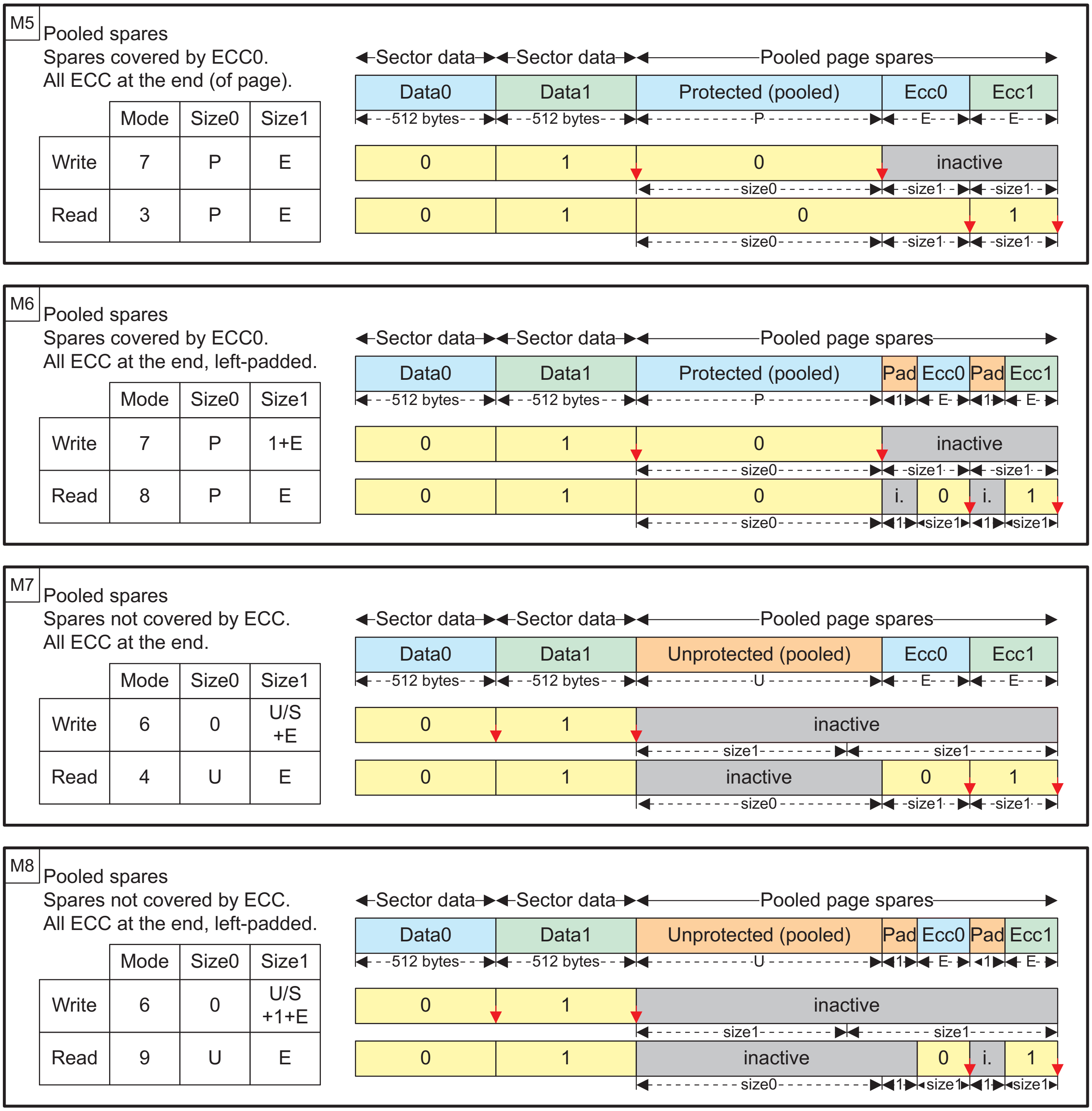
	Mode	Size0	Size1
Write	2	U+E	0
Read	2	U	E

9.1.3.3.12.3.3.2 Pooled Spare Mapping

In these schemes (Figure 9-39), the spare area is pooled for the page.

- The ECC of each sector is aligned at the end of the spare area.
- The non-ECC spare data may or may not be covered by the ECC of sector 0

Figure 9-39. NAND Page Mapping and ECC: Pooled Spare Schemes



M5

	Mode	Size0	Size1
Write	7	P	E
Read	3	P	E

M6

	Mode	Size0	Size1
Write	7	P	1+E
Read	8	P	E

M7

	Mode	Size0	Size1
Write	6	0	U/S +E
Read	4	U	E

M8

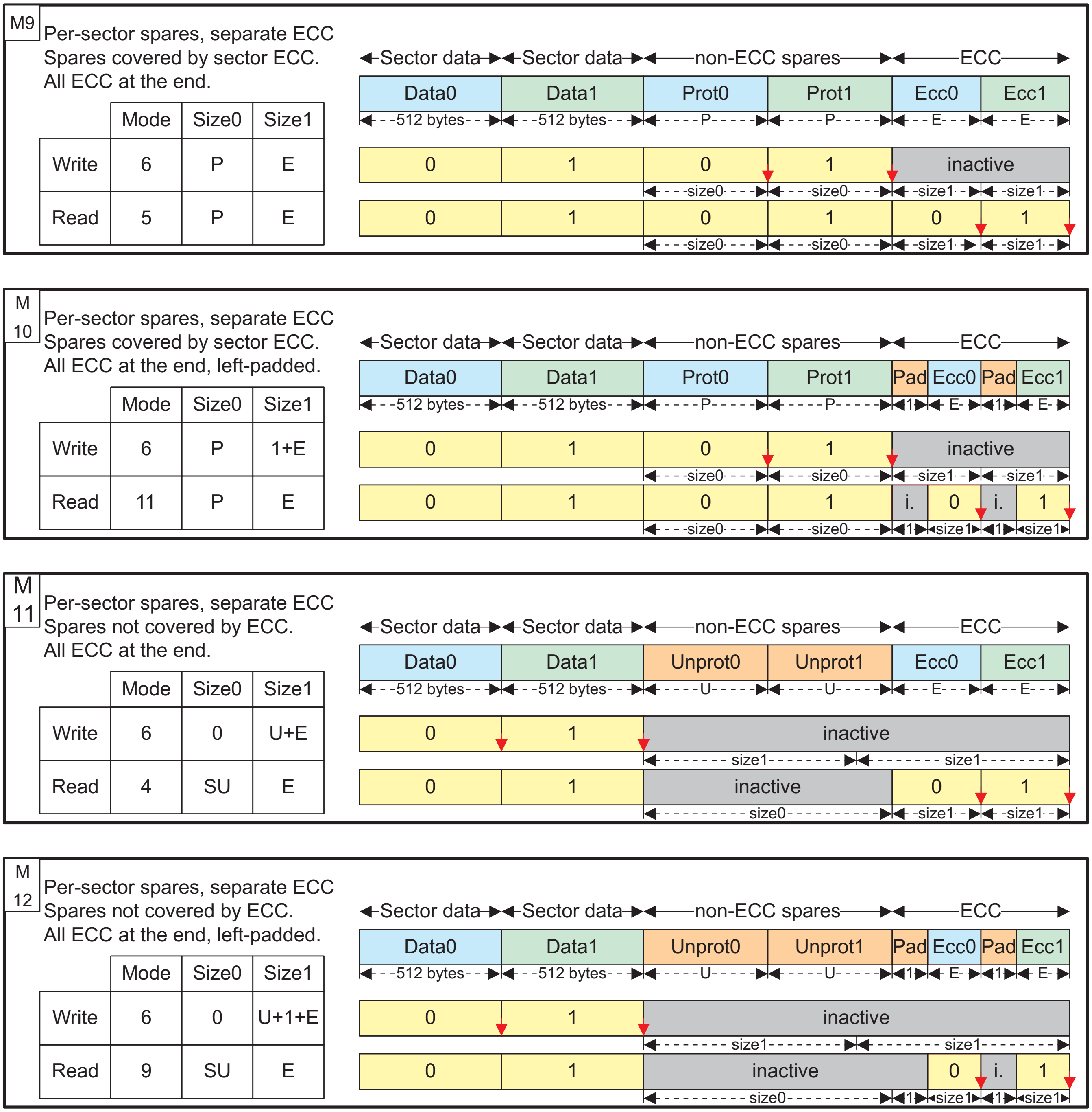
	Mode	Size0	Size1
Write	6	0	U/S +1+E
Read	9	U	E

9.1.3.3.12.3.3.3 Per-Sector Spare Mapping, With ECC Separated at the End of the Page

In these schemes (Figure 9-40), each 512-byte sector of the main area is associated with two sections of the spare area.

- ECC section, all aligned at the end of the page
- other data section, aligned before the ECCs, each of which may or may not be protected by its sectors ECC

Figure 9-40. NAND Page Mapping and ECC: Per-Sector Schemes, with Separate ECC



	Mode	Size0	Size1
Write	6	P	E
Read	5	P	E

	Mode	Size0	Size1
Write	6	P	1+E
Read	11	P	E

	Mode	Size0	Size1
Write	6	0	U+E
Read	4	SU	E

	Mode	Size0	Size1
Write	6	0	U+1+E
Read	9	SU	E

9.1.3.3.12.4 Prefetch and Write-Posting Engine

NAND device data access cycles are usually much slower than the MPU system frequency; such NAND read or write accesses issued by the processor will impact the overall system performance, especially considering long read or write sequences required for NAND page loading or programming. To minimize this effect on system performance, the GPMC includes a prefetch and write-posting engine, which can be used to read from or write to any chip-select location in a buffered manner.

The prefetch and write-posting engine is a simplified embedded-access requester that presents requests to the access engine on a user-defined chip-select target. The access engine interleaves these requests with any request coming from the L3 interface; as a default the prefetch and write-posting engine has the lowest priority.

The prefetch and write-posting engine is dedicated to data-stream access (as opposed to random data access); thus, it is primarily dedicated to NAND support. The engine does not include an address generator; the request is limited to chip-select target identification. It includes a 64-byte FIFO associated with a DMA request synchronization line, for optimal DMA-based use.

The prefetch and write-posting engine uses an embedded 64 bytes (32 16-bit word) FIFO to prefetch data from the NAND device in read mode (prefetch mode) or to store host data to be programmed into the NAND device in write mode (write-posting mode). The FIFO draining and filling (read and write) can be controlled either by the MPU through interrupt synchronization (an interrupt is triggered whenever a programmable threshold is reached) or the sDMA through DMA request synchronization, with a programmable request byte size in both prefetch or posting mode.

The prefetch and write-posting engine includes a single memory pool. Therefore, only one mode, read or write, can be used at any given time. In other words, the prefetch and write-posting engine is a single-context engine that can be allocated to only one chip-select at a time for a read prefetch or a write-posting process.

The engine does not support atomic command and address phase programming and is limited to linear memory read or write access. In consequence, it is limited to NAND data-stream access. The engine relies on the MPU NAND software driver to control block and page opening with the correct data address pointer initialization, before the engine can read from or write to the NAND memory device.

Once started, the engine data reads and writes sequencing is solely based on FIFO location availability and until the total programmed number of bytes is read or written.

Any host-concurrent accesses to a different chip-select are correctly interleaved with ongoing engine accesses. The engine has the lowest priority access so that host accesses to a different chip-select do not suffer a large latency.

A round-robin arbitration scheme can be enabled to ensure minimum bandwidth to the prefetch and write-posting engine in the case of back-to-back direct memory requests to a different chip-select. If the GPMC_PREFETCH_CONFIG1[23] PFPWENROUNDROBIN bit is enabled, the arbitration grants the prefetch and write posting engine access to the GPMC bus for a number of requests programmed in the GPMC_PREFETCH_CONFIG1[19-16] PFPWWEIGHTEDPRIO field.

The prefetch/write-posting engine read or write request is routed to the access engine with the chip-select destination ID. After the required arbitration phase, the access engine processes the request as a single access with the data access size equal to the device size specified in the corresponding chip-select configuration.

The destination chip-select configuration must be set to the NAND protocol-compatible configuration for which address lines are not used (the address bus is not changed from its current value). Selecting a different chip-select configuration can produce undefined behavior.

9.1.3.3.12.4.1 General Facts About the Engine Configuration

The engine can be configured only if the GPMC_PREFETCH_CTRL[0] STARTENGINE bit is de-asserted.

The engine must be correctly configured in prefetch or write-posting mode and must be linked to a NAND chip-select before it can be started. The chip-select is linked using the GPMC_PREFETCH_CONFIG1[26-24] ENGINECSSELECTOR field.

In both prefetch and write-posting modes, the engine respectivelly uses byte or 16-bit word access requests for an 8- or 16-bit wide NAND device attached to the linked chip-select. The FIFOTHRESHOLD and TRANSFERCOUNT fields must be programmed accordingly as a number of bytes or a number of 16-bit word.

When the GPMC_PREFETCH_CONFIG1[7] ENABLEENGINE bit is set, the FIFO entry on the L3 interconnect port side is accessible at any address in the associated chip-select memory region. When the ENABLEENGINE bit is set, any host access to this chip-select is rerouted to the FIFO input. Directly accessing the NAND device linked to this chip-select from the host is still possible through these registers:

- GPMC_NAND_COMMAND_i
- GPMC_NAND_ADDR_i
- GPMC_NAND_DATA_i

The FIFO entry on the L3 interconnect port can be accessed with Byte, 16-bit word, or 32-bit word access size, according to little-endian format, even though the FIFO input is 32-bit wide.

The FIFO control is made easier through the use of interrupts or DMA requests associated with the FIFOTHRESHOLD bit field. The GPMC_PREFETCH_STS[30-24] FIFOPOINTER field monitors the number of available bytes to be read in prefetch mode or the number of free empty slots which can be written in write-posting mode. The GPMC_PREFETCH_STS[13-0] COUNTVALUE field monitors the number of remaining bytes to be read or written by the engine according to the TRANSFERCOUNT value. The FIFOPOINTER and COUNTVALUE bit fields are always expressed as a number of bytes even if a 16-bit wide NAND device is attached to the linked chip-select.

In prefetch mode, when the FIFOPOINTER equals 0, that is, the FIFO is empty, a host read access receives the byte last read from the FIFO as its response. In case of 32-bit word or 16-bit word read accesses, the last byte read from the FIFO is copied the required number of times to fit the requested word size. In write-posting mode, when the FIFOPOINTER equals 0, that is, the FIFO is full, a host write overwrites the last FIFO byte location. There is no underflow or overflow error reporting in the GPMC.

9.1.3.3.12.4.2 Prefetch Mode

The prefetch mode is selected when the GPMC_PREFETCH_CONFIG1[0] ACCESSMODE bit is cleared.

The MPU NAND software driver must issue the block and page opening (READ) command with the correct data address pointer initialization before the engine can be started to read from the NAND memory device. The engine is started by asserting the GPMC_PREFETCH_CTRL[0] STARTENGINE bit. The STARTENGINE bit automatically clears when the prefetch process completes.

If required, the ECC calculator engine must be initialized (i.e., reset, configured, and enabled) before the prefetch engine is started, so that the ECC is correctly computed on all data read by the prefetch engine.

When the GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE bit is cleared, the prefetch engine starts requesting data as soon as the STARTENGINE bit is set. If using this configuration, the host must monitor the NAND device-ready pin so that it only sets the STARTENGINE bit when the NAND device is in a ready state, meaning data is valid for prefetching.

When the SYNCHROMODE bit is set, the prefetch engine starts requesting data when an active to inactive wait signal transition is detected. The transition detector must be cleared before any transition detection; see Section 9.1.3.3.12.2.2. The GPMC_PREFETCH_CONFIG1[5-4] WAITPINSELECTOR field selects which gpmc_wait pin edge detector triggers the prefetch engine in this synchronized mode.

If the STARTENGINE bit is set after the NAND address phase (page opening command), the engine is effectively started only after the actual NAND address phase completion. To prevent GPMC stall during this NAND address phase, set the STARTENGINE bit field before NAND address phase completion when in synchronized mode. The prefetch engine will start when an active to inactive wait signal transition is detected. The STARTENGINE bit is automatically cleared on prefetch process completion.

The prefetch engine issues a read request to fill the FIFO with the amount of data specified by GPMC_PREFETCH_CONFIG2[13-0] TRANSFERCOUNT field.

Table 9-22 describes the prefetch mode configuration.

Table 9-22. Prefetch Mode Configuration

Bit Field	Register	Value	Comments
STARTENGINE	GPMC_PREFETCH_CTRL	0	Prefetch engine can be configured only if STARTENGINE is cleared to 0.
ENGINECSSELECTOR	GPMC_PREFETCH_CONFIG1	0 to 3h	Selects the chip-select associated with a NAND device where the prefetch engine is active.
ACCESSMODE	GPMC_PREFETCH_CONFIG1	0	Selects prefetch mode
FIFOTHRESHOLD	GPMC_PREFETCH_CONFIG1		Selects the maximum number of bytes read or written by the host on DMA or interrupt request
TRANSFERCOUNT	GPMC_PREFETCH_CONFIG1		Selects the number of bytes to be read or written by the engine to the selected chip-select
SYNCHROMODE	GPMC_PREFETCH_CONFIG1	0/1	Selects when the engine starts the access to the chip-select
WAITPINSELECT	GPMC_PREFETCH_CONFIG1	0 to 1	Selects wait pin edge detector (if GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE = 1)
ENABLEOPTIMIZEDACCESS	GPMC_PREFETCH_CONFIG1	0/1	See Section 9.1.3.3.12.4.6
CYCLEOPTIMIZATION	GPMC_PREFETCH_CONFIG1		Number of clock cycle removed to timing parameters
ENABLEENGINE	GPMC_PREFETCH_CONFIG1	1	Engine enabled
STARTENGINE	GPMC_PREFETCH_CONFIG1	1	Starts the prefetch engine

9.1.3.3.12.4.3 FIFO Control in Prefetch Mode

The FIFO can be drained directly by the MPU or by an eDMA channel.

In MPU draining mode, the FIFO status can be monitored through the GPMC_PREFETCH_STS[30-24] FIFOPOINTER field or through the GPMC_PREFETCH_STS[16] FIFOTHRSTS bit. The FIFOPOINTER indicates the current number of available data to be read; FIFOTHRSTS set to 1 indicates that at least FIFOTHRESHOLD bytes are available from the FIFO.

An interrupt can be triggered by the GPMC if the GPMC_IRQEN[0] FIFOEVTEN bit is set. The FIFO interrupt event is logged, and the GPMC_IRQSTS[0] FIFOEVTSTS bit is set. To clear the interrupt, the MPU must read all the available bytes, or at least enough bytes to get below the programmed FIFO threshold, and the FIFOEVTSTS bit must be cleared to enable further interrupt events. The FIFOEVTSTS bit must always be reset prior to asserting the FIFOEVTEN bit to clear any out-of-date logged interrupt event. This interrupt generation must be enabled after enabling the STARTENGINE bit.

Prefetch completion can be monitored through the GPMC_PREFETCH_STS[13-0] COUNTVALUE field. COUNTVALUE indicates the number of currently remaining data to be requested according to the TRANSFERCOUNT value. An interrupt can be triggered by the GPMC when the prefetch process is complete (that is, COUNTVALUE equals 0) if the GPMC_IRQEN[1] TERMINALCOUNTEVTEN bit is set. At prefetch completion, the TERMINALCOUNT interrupt event is also logged, and the GPMC_IRQSTS[1] TERMINALCOUNTSTS bit is set. To clear the interrupt, the MPU must clear the TERMINALCOUNTSTS bit. The TERMINALCOUNTSTS bit must always be cleared prior to asserting the TERMINALCOUNTEVTEN bit to clear any out-of-date logged interrupt event.

NOTE: The COUNTVALUE value is only valid when the prefetch engine is active (started), and an interrupt is only triggered when COUNTVALUE reaches 0, that is, when the prefetch engine automatically goes from an active to an inactive state.

The number of bytes to be prefetched (programmed in TRANSFERCOUNT) must be a multiple of the programmed FIFOTHRESHOLD to trigger the correct number of interrupts allowing a deterministic and transparent FIFO control. If this guideline is respected, the number of ISR accesses is always required and the FIFO is always empty after the last interrupt is trigerred. In other cases, the TERMINALCOUNT interrupt must be used to read the remaining bytes in the FIFO (the number of remaining bytes being lower than the FIFOTHRESHOLD value).

In DMA draining mode, the GPMC_PREFETCH_CONFIG1[2] DMAMODE bit must be set so that the GPMC issues a DMA hardware request when at least FIFOTHRESHOLD bytes are ready to be read from the FIFO. The DMA channel owning this DMA request must be programmed so that the number of bytes programmed in FIFOTHRESHOLD is read from the FIFO during the DMA request process. The DMA request is kept active until this number of bytes has effectively been read from the FIFO, and no other DMA request can be issued until the ongoing active request is complete.

In prefetch mode, the TERMINALCOUNT event is also a source of DMA requests if the number of bytes to be prefetched is not a multiple of FIFOTHRESHOLD, the remaining bytes in the FIFO can be read by the DMA channel using the last DMA request. This assumes that the number of remaining bytes to be read is known and controlled through the DMA channel programming model.

Any potentially active DMA request is cleared when the prefetch engine goes from inactive to active prefetch (the STARTENGINE bit is set to 1). The associated DMA channel must always be enabled by the MPU after setting the STARTENGINE bit so that the out-of-date active DMA request does not trigger spurious DMA transfers.

9.1.3.3.12.4.4 Write-Posting Mode

The write-posting mode is selected when the GPMC_PREFETCH_CONFIG1[0] ACCESSMODE bit is set.

The MPU NAND software driver must issue the correct address pointer initialization command (page program) before the engine can start writing data into the NAND memory device. The engine starts when the GPMC_PREFETCH_CTRL[0] STARTENGINE bit is set to 1. The STARTENGINE bit clears automatically when posting completes. When all data have been written to the NAND memory device, the MPU NAND software driver must issue the second cycle program command and monitor the status for programming process completion (adding ECC handling, if required).

If used, the ECC calculator engine must be started (configured, reset, and enabled) before the posting engine is started so that the ECC parities are properly calculated on all data written by the prefetch engine to the associated chip-select.

In write-posting mode, the GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE bit must be cleared so that posting starts as soon as the STARTENGINE bit is set and the FIFO is not empty.

If the STARTENGINE bit is set after the NAND address phase (page program command), the STARTENGINE setting is effective only after the actual NAND command completion. To prevent GPMC stall during this NAND command phase, set the STARTENGINE bit field before the NAND address completion and ensure that the associated DMA channel is enabled after the NAND address phase.

The posting engine issues a write request when valid data are available from the FIFO and until the programmed GPMC_PREFETCH_CONFIG2[13-0] TRANSFERCOUNT accesses have been completed.

The STARTENGINE bit clears automatically when posting completes. When all data have been written to the NAND memory device, the MPU NAND software driver must issue the second cycle program command and monitor the status for programming process completion. The closing program command phase must only be issued when the full NAND page has been written into the NAND flash write buffer, including the spare area data and the ECC parities, if used.

Table 9-23. Write-Posting Mode Configuration

Bit Field	Register	Value	Comments
STARTENGINE	GPMC_PREFETCH_CTRL	0	Write-posting engine can be configured only if STARTENGINE is cleared to 0.
ENGINECSSELECTOR	GPMC_PREFETCH_CONFIG1	0 to 3h	Selects the chip-select associated with a NAND device where the prefetch engine is active
ACCESSMODE	GPMC_PREFETCH_CONFIG1	1	Selects write-posting mode
FIFOTHRESHOLD	GPMC_PREFETCH_CONFIG1		Selects the maximum number of bytes read or written by the host on DMA or interrupt request
TRANSFERCOUNT	GPMC_PREFETCH_CONFIG2		Selects the number of bytes to be read or written by the engine from/to the selected chip-select
SYNCHROMODE	GPMC_PREFETCH_CONFIG1	0	Engine starts the access to chip-select as soon as STARTENGINE is set.
ENABLEOPTIMIZEDACCESS	GPMC_PREFETCH_CONFIG1	0/1	See Section 9.1.3.3.12.4.6
CYCLEOPTIMIZATION	GPMC_PREFETCH_CONFIG		
ENABLEENGINE	GPMC_PREFETCH_CONFIG1	1	Engine enabled
STARTENGINE	GPMC_PREFETCH_CTRL	1	Starts the prefetch engine

9.1.3.3.12.4.5 FIFO Control in Write-Posting Mode

The FIFO can be filled directly by the MPU or by an sDMA channel.

In MPU filling mode, the FIFO status can be monitored through the FIFOPOINTER or through the GPMC_PREFETCH_STS[16] FIFOTHRSTS bit. FIFOPOINTER indicates the current number of available free byte places in the FIFO, and the FIFOTHRSTS bit, when set, indicates that at least FIFOTHRESHOLD free byte places are available in the FIFO.

An interrupt can be issued by the GPMC if the GPMC_IRQEN[0] FIFOEVTEN bit is set. When the interrupt is fired, the GPMC_IRQSTS[0] FIFOEVTSTS bit is set. To clear the interrupt, the MPU must write enough bytes to fill the FIFO, or enough bytes to get below the programmed threshold, and the FIFOEVTSTS bit must be cleared to get further interrupt events. The FIFOEVTSTS bit must always be cleared prior to asserting the FIFOEVTEN bit to clear any out-of-date logged interrupt event. This interrupt must be enabled after enabling the STARTENGINE bit

The posting completion can be monitored through the GPMC_PREFETCH_STS[13-0] COUNTVALUE field. COUNTVALUE indicates the current number of remaining data to be written based on the TRANSFERCOUNT value. An interrupt is issued by the GPMC when the write-posting process completes (that is, COUNTVALUE equal to 0) if the GPMC_IRQEN[1] TERMINALCOUNTEVTEN bit is set. When the interrupt is fired, the GPMC_IRQSTS[1] TERMINALCOUNTSTS bit is set. To clear the interrupt, the MPU must clear the TERMINALCOUNTSTS bit. The TERMINALCOUNTSTS bit must always be cleared prior to asserting the TERMINALCOUNTEVTEN bit to clear any out-of-date logged interrupt event.

NOTE: The COUNTVALUE value is only valid if the write-posting engine is active and started, and an interrupt is only issued when COUNTVALUE reaches 0, that is, when the posting engine automatically goes from active to inactive.

In DMA filling mode, the DMAMode bit field in the GPMC_PREFETCH_CONFIG1[2] DMAMODE bit must be set so that the GPMC issues a DMA hardware request when at least FIFOTHRESHOLD bytes-free places are available in the FIFO. The DMA channel owning this DMA request must be programmed so that a number of bytes equal to the value programmed in the FIFOTHRESHOLD bit field are written into the FIFO during the DMA access. The DMA request remains active until the associated number of bytes has effectively been written into the FIFO, and no other DMA request can be issued until the ongoing active request has been completed.

Any potentially active DMA request is cleared when the prefetch engine goes from inactive to active prefetch (STARTENGINE set to 1). The associated DMA channel must always be enabled by the MPU after setting the STARTENGINE bit so that an out-of-date active DMA request does not trigger spurious DMA transfers.

In write-posting mode, the DMA or the MPU fill the FIFO with no consideration to the associated byte enables. Any byte stored in the FIFO is written into the memory device.

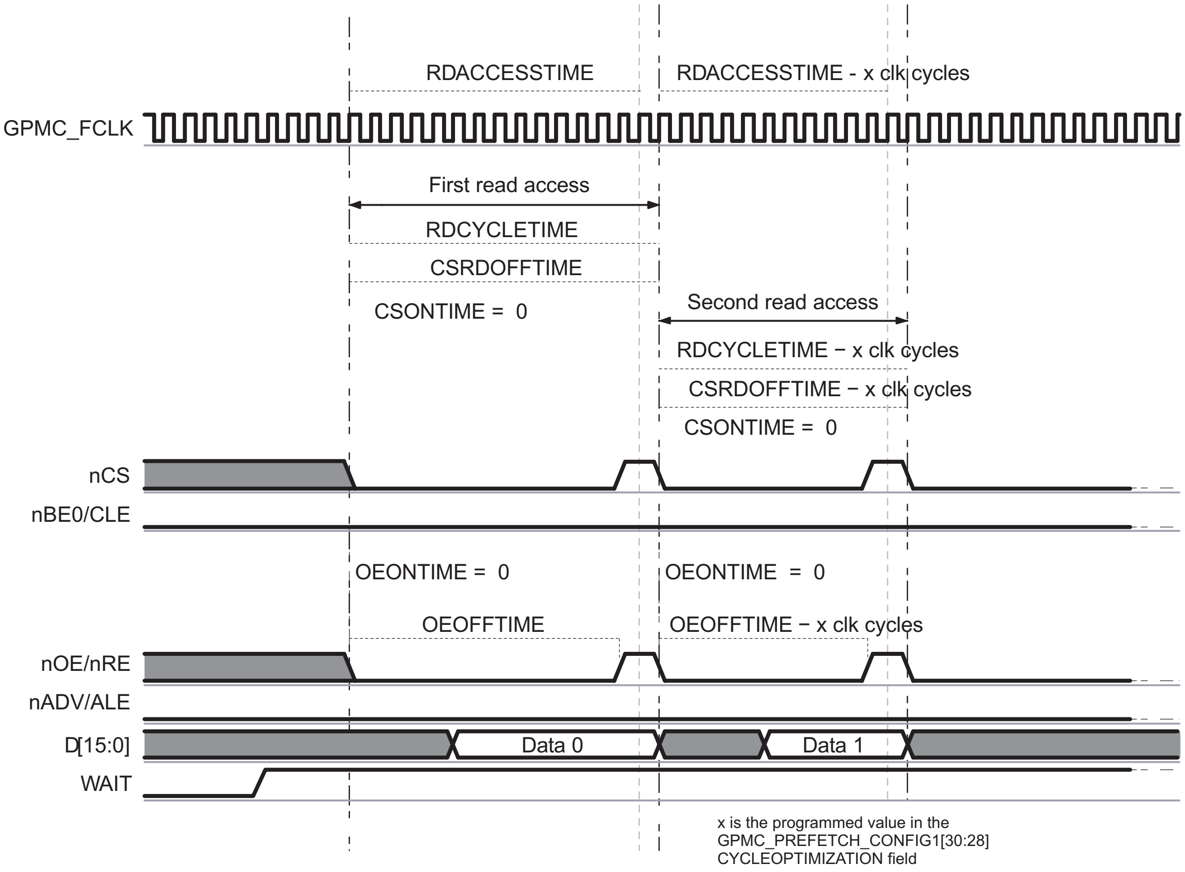
9.1.3.3.12.4.6 Optimizing NAND Access Using the Prefetch and Write-Posting Engine

Access time to a NAND memory device can be optimized for back-to-back accesses if the associated CSn signal is not deasserted between accesses. The GPMC access engine can track prefetch engine accesses to optimize the access timing parameter programmed for the allocated chip-select, if no accesses to other chip-selects (that is, interleaved accesses) occur. Similarly, the access engine also eliminates the CYCLE2CYCLEDELAY even if CYCLE2CYCLESAMECSEN is set. This capability is limited to the prefetch and write-posting engine accesses, and MPU accesses to a NAND memory device (through the defined chip-select memory region or through the GPMC_NAND_DATA_i are never optimized.

The GPMC_PREFETCH_CONFIG1[27] ENABLEOPTIMIZEDACCESS bit must be set to enable optimized accesses. To optimize access time, the GPMC_PREFETCH_CONFIG1[30-28] CYCLEOPTIMIZATION field defines the number of GPMC_FCLK cycles to be suppressed from the RDCYCLETIME, WRCYCLETIME, RDACCESSTIME, WRACCESSTIME, CSOFFTIME, ADVOFFTIME, OEOFFTIME, and WEOFFTIME timing parameters.

Figure 9-41, in the case of back-to-back accesses to the NAND flash through the prefetch engine, CYCLE2CYCLESAMECSEN is forced to 0 when using optimized accesses. The first access uses the regular timing settings for this chip-select. All accesses after this one use settings reduced by x clock cycles, x being defined by the GPMC_PREFETCH_CONFIG1[30-28] CYCLEOPTIMIZATION field.

Figure 9-41. NAND Read Cycle Optimization Timing Description



9.1.3.3.12.4.7 Interleaved Accesses Between Prefetch and Write-Posting Engine and Other Chip-Selects

Any on-going read or write access from the prefetch and write-posting engine is completed before an access to any other chip-select can be initiated. As a default, the arbiter uses a fixed-priority algorithm, and the prefetch and write-posting engine has the lowest priority. The maximum latency added to access starting time in this case equals the RDCYCLETIME or WRCYCLETIME (optimized or not) plus the requested BUSTURNAROUND delay for bus turnaround completion programmed for the chip-select to which the NAND device is connected to.

Alternatively, a round-robin arbitration can be used to prioritize accesses to the external bus. This arbitration scheme is enabled by setting the GPMC_PREFETCH_CONFIG1[23] PFPWENROUNDROBIN bit. When a request to another chip-select is received while the prefetch and write-posting engine is active, priority is given to the new request. The request processed thereafter is the prefetch and write-posting engine request, even if another interconnect request is passed in the mean time. The engine keeps control of the bus for an additional number of requests programmed in the GPMC_PREFETCH_CONFIG1[19-16] PFPWWEIGHTEDPRIO bit field. Control is then passed to the direct interconnect request.

As an example, the round-robin arbitration scheme is selected with PFPWWEIGHTEDPRIO set to 2h. Considering the prefetch and write-posting engine and the interconnect interface are always requesting access to the external interface, the GPMC grants priority to the direct interconnect access for one request. The GPMC then grants priority to the engine for three requests, and finnaly back to the direct interconnect access, until the arbiter is reset when one of the two initiators stops initiating requests.

9.1.4 GPMC High-Level Programming Model Overview

The high-level programming model introduces a top-down approach for users that need to configure the GPMC module. Figure 9-42 shows a programming model top-level diagram for the GPMC. Each block of the diagram is described in one of the following subsections through a set of registers to configure. Table 9-24 and Table 9-25 list each step in the model.

Figure 9-42. Programming Model Top-Level Diagram

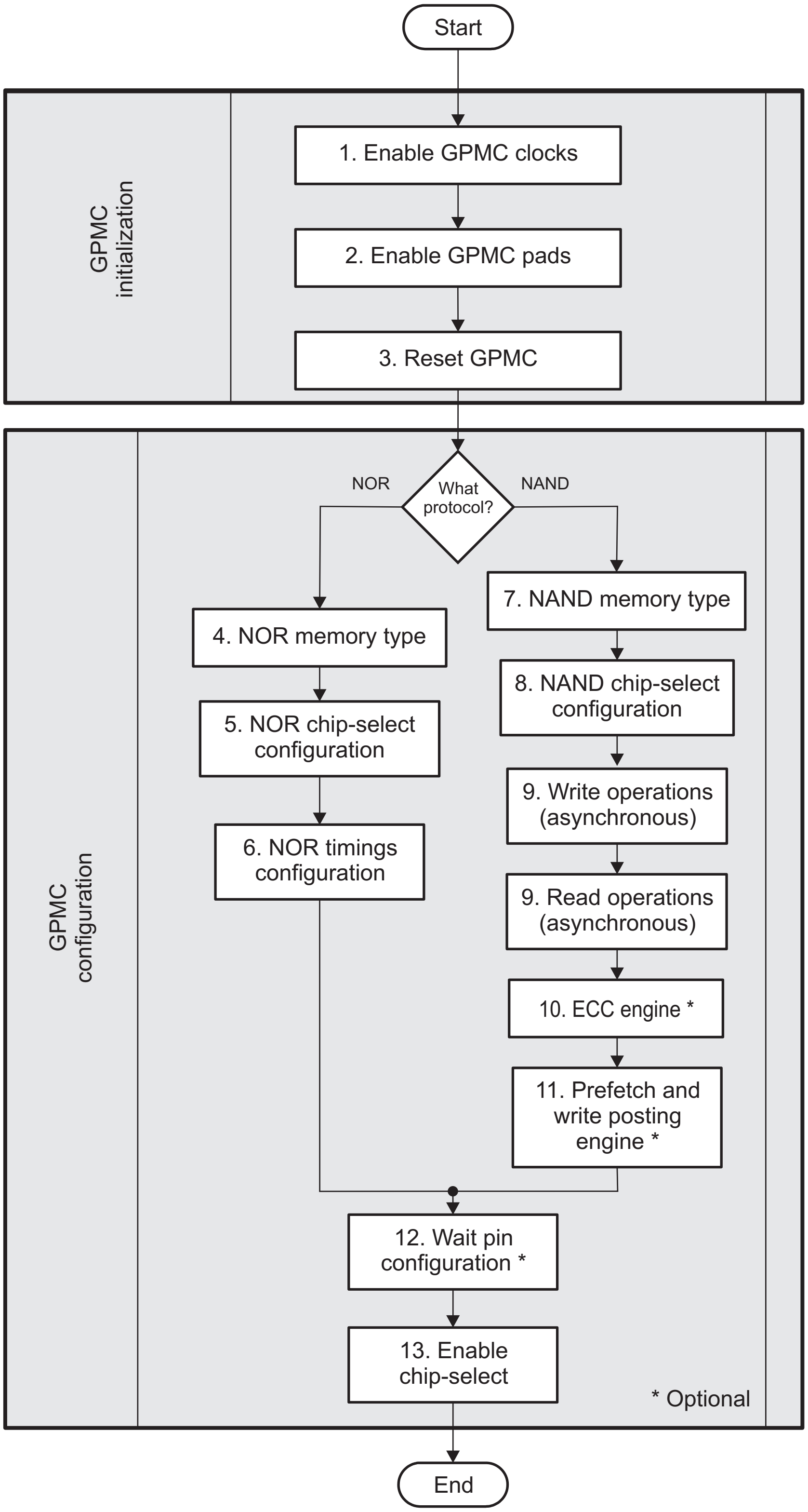


Table 9-24. GPMC Configuration in NOR Mode

Step	Description
NOR Memory Type	See Table 9-27
NOR Chip-Select Configuration	See Table 9-28
NOR Timings Configuration	See Table 9-29
Wait Pin Configuration	See Table 9-30
Enable Chip-Select	See Table 9-31

Table 9-25. GPMC Configuration in NAND Mode

Step	Description
NAND Memory Type	See Table 9-32
NAND Chip-Select Configuration	See Table 9-33
Write Operations (Asynchronous)	See Table 9-34
Read Operations (Asynchronous)	See Table 9-34
ECC Engine	See Table 9-35
Prefetch and Write-Posting Engine	See Table 9-36
Wait Pin Configuration	See Table 9-37
Enable Chip-Select	See Table 9-38

9.1.4.1 GPMC Initialization

Table 9-26 describes the settings required to reset the GPMC.

Table 9-26. Reset GPMC

Sub-process Name	Register / Bitfield	Value
Start a software reset	GPMC_SYSCONFIG[1] SOFTRESET	1
Wait until	GPMC_SYSSTS[0] RESETDONE	1

9.1.4.2 GPMC Configuration in NOR Mode

This section gives a generic configuration for parameters related to the NOR memory connected to the GPMC. Table 9-27 through Table 9-31 list the steps to configure the GPMC in NOR mode.

NOTE: In the tables of this section, 'x' in Value column stands for 'depends on configuration'.

Table 9-27. NOR Memory Type

Sub-process Name	Register / Bitfield	Value
Set the NOR protocol	GPMC_CONFIG1_i[11-10] DEVICETYPE	0
Set a device size	GPMC_CONFIG1_i[13-12] DEVICESIZE	x
Select an address and data multiplexing protocol	GPMC_CONFIG1_i[9] MUXADDDATA	x
Set the attached device page length	GPMC_CONFIG1_i[24-23] ATTACHEDDEVICEPAGELENGTH	x
Set the wrapping burst capabilities	GPMC_CONFIG1_i[31] WRAPBURST	x
Select a timing signals latencies factor	GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY	x
Select an output clock frequency	GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER	x
Choose an output clock activation time	GPMC_CONFIG1_i[26-25] CLKACTIVATIONTIME	x
Set a single or multiple access for read operations	GPMC_CONFIG1_i[30] READMULTIPLE	x
Set a synchronous or asynchronous mode for read operations	GPMC_CONFIG1_i[29] READTYPE	x
Set a single or multiple access for write operations	GPMC_CONFIG1_i[28] WRITEMULTIPLE	x
Set a synchronous or asynchronous mode for write operations	GPMC_CONFIG1_i[27] WRITETYPE	x

Table 9-28. NOR Chip-Select Configuration

Sub-process Name	Register / Bitfield	Value
Select the chip-select base address	GPMC_CONFIG7_i[5-0] BASEADDRESS	x
Select the chip-select mask address	GPMC_CONFIG7_i[11-8] MASKADDRESS	x

Table 9-29. NOR Timings Configuration

Sub-process Name	Register / Bitfield	Value
Configure adequate timing parameters in various memory modes	See Section 9.1.4.5	

Table 9-30. WAIT Pin Configuration

Sub-process Name	Register / Bitfield	Value
Enable or disable wait pin monitoring for read operations	GPMC_CONFIG1_i[22] WAITREADMONITORING	x
Enable or disable wait pin monitoring for write operations	GPMC_CONFIG1_i[21] WAITWRITEMONITORING	x
Select a wait pin monitoring time	GPMC_CONFIG1_i[19-18] WAITMONITORINGTIME	x
Choose the input wait pin for the chip-select	GPMC_CONFIG1_i[17-16] WAITPINSELECT	x

Table 9-31. Enable Chip-Select

Sub-process Name	Register / Bitfield	Value
When all parameters are configured, enable the chip-select	GPMC_CONFIG7_i[6] CSVALID	x

9.1.4.3 GPMC Configuration in NAND Mode

This section gives a generic configuration for parameters related to NAND memory connected to the GPMC.

Table 9-32. NAND Memory Type

Sub-process Name	Register / Bitfield	Value
Set the NAND protocol	GPMC_CONFIG1_i[11-10] DEVICETYPE	2h
Set a device size	GPMC_CONFIG1_i[13-12] DEVICESIZE	x
Set the address and data multiplexing protocol to non-multiplexed attached device	GPMC_CONFIG1_i[9] MUXADDDATA	0
Select a timing signals latencies factor	GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY	x
Set a synchronous or asynchronous mode and a single or multiple access for read and write operations	See Section 9.1.4.4	x

Table 9-33. NAND Chip-Select Configuration

Sub-process Name	Register / Bitfield	Value
Select the chip-select base address	GPMC_CONFIG7_i[5-0] BASEADDRESS	x
Select the chip-select minimum granularity (16M bytes)	GPMC_CONFIG7_i[11-8] MASKADDRESS	x

Table 9-34. Asynchronous Read and Write Operations

Sub-process Name	Register / Bitfield	Value
Configure adequate timing parameters in asynchronous modes	See Section 9.1.4.5	

Table 9-35. ECC Engine

Sub-process Name	Register / Bitfield	Value
Select the ECC result register where the first ECC computation is stored (Only applies to Hamming)	GPMC_ECC_CONTROL[3-0] ECCPOINTER	x
Clear all ECC result registers	GPMC_ECC_CONTROL[8] ECCCLEAR	Write 1 to clear
Define ECCSIZE0 and ECCSIZE1	GPMC_ECC_SIZE_CONFIG[19-12] ECCSIZE0 and GPMC_ECC_SIZE_CONFIG[29-22] ECCSIZE1	x
Select the size of each of the 9 result registers (size specified by ECCSIZE0 or ECCSIZE1)	GPMC_ECC_SIZE_CONFIG[j-1] ECCjRESULTSIZE where j = 1 to 9	x
Select the chip-select where ECC is computed	GPMC_ECC_SIZE_CONFIG[3-1] ECCCS	x
Select the Hamming code or BCH code ECC algorithm in use	GPMC_ECC_SIZE_CONFIG[16] ECCALGORITHM	x
Select word size for ECC calculation	GPMC_ECC_SIZE_CONFIG[7] ECC16B	x
If the BCH code is used, Set an error correction capability and Select a number of sectors to process	GPMC_ECC_SIZE_CONFIG[13-12] ECCBCHTSEL and GPMC_ECC_SIZE_CONFIG[6-4] ECCTOPSECTOR	x
Enable the ECC computation	GPMC_ECC_SIZE_CONFIG[0] ECCENABLE	1

Table 9-36. Prefetch and Write-Posting Engine

Sub-process Name	Register / Bitfield	Value
Disable the engine before configuration	GPMC_PREFETCH_CONTROL[0] STARTENGINE	0
Select the chip-select associated with a NAND device where the prefetch engine is active	GPMC_PREFETCH_CONFIG1[26-24] ENGINECSSELECTOR	x
Select access direction through prefetch engine, read or write.	GPMC_PREFETCH_CONFIG1[0] ACCESSMODE	x
Select the threshold used to issue a DMA request	GPMC_PREFETCH_CONFIG1[14-8] FIFOTHRESHOLD	x
Select either DMA synchronized mode or SW manual mode.	GPMC_PREFETCH_CONFIG1[2] DMAMODE	x
Select if the engine immediately starts accessing the memory upon STARTENGINE assertion or if hardware synchronization based on a WAIT signal is used.	GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE	x
Select which wait pin edge detector should start the engine in synchronized mode	GPMC_PREFETCH_CONFIG1[5-4] WAITPINSELECTOR	x
Enter a number of clock cycles removed to timing parameters (For all back-to-back accesses to the NAND flash but not the first one)	GPMC_PREFETCH_CONFIG1[30-28] CYCLEOPTIMIZATION	x
Enable the prefetch postwite engine	GPMC_PREFETCH_CONFIG1[7] ENABLEENGINE	1
Select the number of bytes to be read or written by the engine to the selected chip-select	GPMC_PREFETCH_CONFIG2[13-0] TRANSFERCOUNT	x
Start the prefetch engine	GPMC_PREFETCH_CONTROL[0] STARTENGINE	1

Table 9-37. WAIT Pin Configuration

Sub-process Name	Register / Bitfield	Value
Selects when the engine starts the access to CS	GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE	x
Select which wait pin edge detector should start the engine in synchronized mode	GPMC_PREFETCH_CONFIG1[5-4] WAITPINSELECTOR	x

Table 9-38. Enable Chip-Select

Sub-process Name	Register / Bitfield	Value
When all parameters are configured, enable the chip-select	GPMC_CONFIG7_i[6] CSVALID	x

9.1.4.4 Set Memory Access

This section details the bit field to configure to set the GPMC in various memory modes.

Table 9-39. Mode Parameters Check List Table

Register	Bit	Bit Field Name	Asynchronous				Synchronous			
			Single Read Access	Single Write Access	Multiple Read (Page) Access	Multiple Write (Page) Access	Single Read Access	Single Write Access	Multiple Read (Burst) Access	Multiple Write (Burst) Access
GPMC_CONFIG1_i	30	READMULTIPLE	0	-	1	N/S	0	-	1	-
GPMC_CONFIG1_i	29	READTYPE	0	-	0	N/S	1	-	1	-
GPMC_CONFIG1_i	28	WRITEMULTIPLE	-	0		N/S	-	0	-	1
GPMC_CONFIG1_i	27	WRITETYPE	-	0		N/S	-	1	-	1

Table 9-40. Access Type Parameters Check List Table

Register	Bit	Bit Field Name	Access Type		
			Non-Mux	Address/Data Mux	AAD Mux
GPMC_CONFIG1_i	9-8	MUXADDDATA	0	2h	1

9.1.4.5 GPMC Timing Parameters

Figure 9-43 shows a programming model diagram for the NOR interfacing timing parameters.

Table 9-41 lists bit fields to configure adequate timing parameter in various memory modes.

Figure 9-43. NOR Interfacing Timing Parameters Diagram

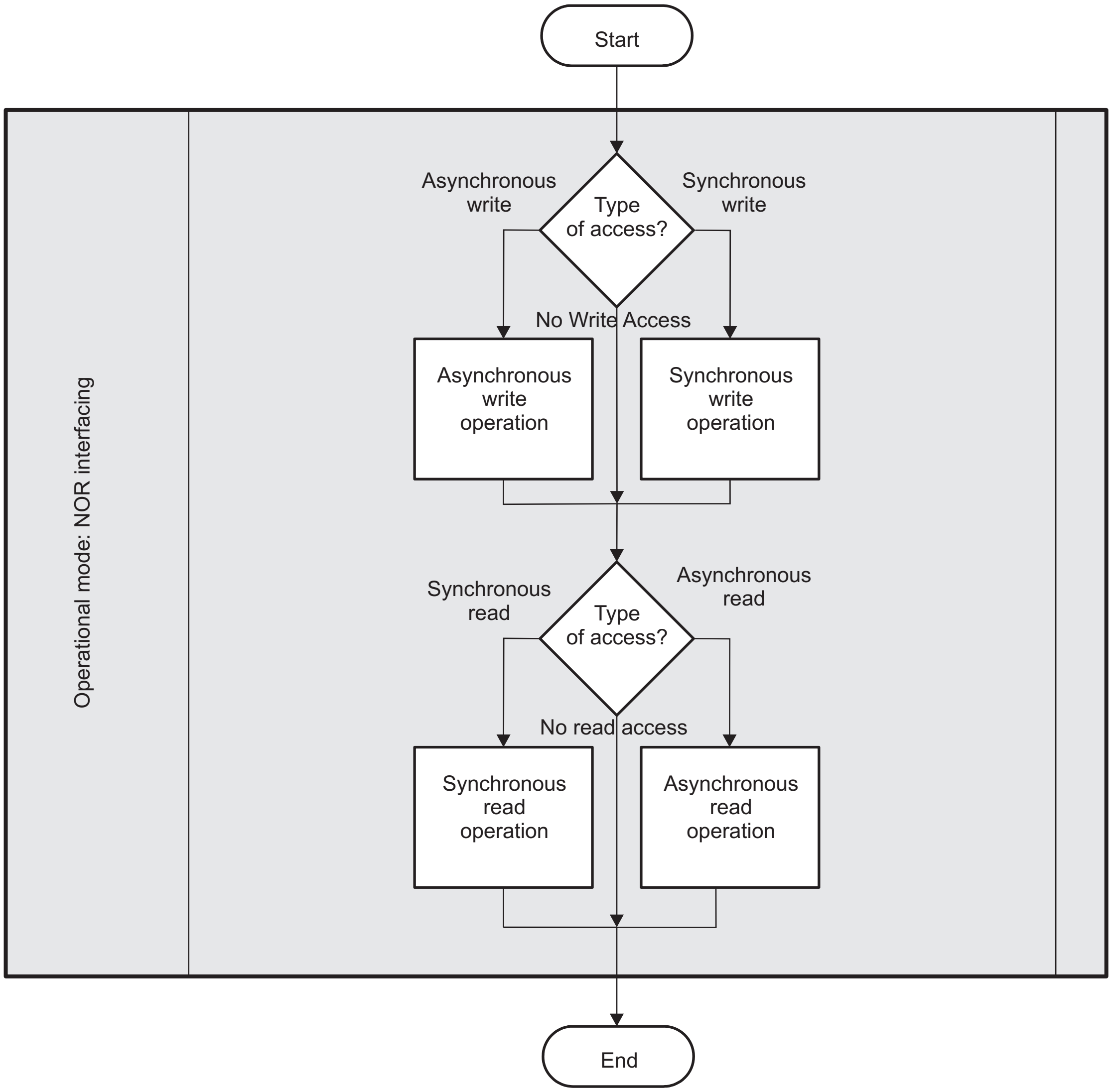


Table 9-41. Timing Parameters

Register	Bit	Bit Field Name	Asynchronous			Synchronous				Access Type		
			Single Read Access	Single Write Access	Multiple Read (Page) access	Single Read Access	Single Write Access	Multiple Read (Burst) Access	Multiple Write (Burst) Access	Non-multiplexed	Address /Data-multiplexed	AAD-multiplexed
GPMC_CONFIG1_i	9-8	MUXADDDATA	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG1_i	29	READTYPE	y		y	y		y		y	y	y
GPMC_CONFIG1_i	30	READMULTIPLE	y		y	y		y		y	y	y
GPMC_CONFIG1_i	27	WRITETYPE		y			y		y	y	y	y
GPMC_CONFIG1_i	28	WRITEMULTIPLE		y			y		y	y	y	y
GPMC_CONFIG1_i	31	WRAPBURST						y	y	y	y	y
GPMC_CONFIG1_i	26-25	CLKACTIVATIONTIME				y	y	y	y	y	y	y
GPMC_CONFIG1_i	19-18	WAITMONITORINGTIME	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG1_i	4	TIMEPARAGRANULARITY	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG2_i	20-16	CSWROFFTIME		y			y		y	y	y	y
GPMC_CONFIG2_i	12-8	CSRDOFFTIME	y		y	y		y		y	y	y
GPMC_CONFIG2_i	7	CSEXTRADELAY	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG2_i	3-0	CSONTIME	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG3_i	30-28	ADVAADMUXWROFFTIME		y			y		y			y
GPMC_CONFIG3_i	26-24	ADVAADMUXRDOFFTIME	y		y	y		y				y
GPMC_CONFIG3_i	6-4	ADVAADMUXONTIME	y	y	y	y	y	y	y			y
GPMC_CONFIG3_i	20-16	ADVWROFFTIME		y			y		y	y	y	y
GPMC_CONFIG3_i	12-8	ADVRDOFFTIME	y		y	y		y		y	y	y
GPMC_CONFIG3_i	7	ADVEXTRADELAY	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG3_i	3-0	ADVONTIME	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG4_i	15-13	OEAADMUXOFFTIME	y	y	y	y	y	y	y			y
GPMC_CONFIG4_i	6-4	OEAADMUXONTIME	y	y	y	y	y	y	y			y
GPMC_CONFIG4_i	28-24	WEOFFTIME		y			y		y	y	y	y
GPMC_CONFIG4_i	23	WEEXTRADELAY		y			y		y	y	y	y
GPMC_CONFIG4_i	19-16	WEONTIME		y			y		y	y	y	y
GPMC_CONFIG4_i	12-8	OEOFFTIME	y		y	y		y		y	y	y
GPMC_CONFIG4_i	7	OEEXTRADELAY	y		y	y		y		y	y	y
GPMC_CONFIG4_i	3-0	OEONTIME	y		y	y		y		y	y	y
GPMC_CONFIG5_i	27-24	PAGEBURSTACCESSTIME			y			y	y	y	y	y
GPMC_CONFIG5_i	20-16	RDACCESSTIME	y		y	y		y		y	y	y
GPMC_CONFIG5_i	12-8	WRCYCLETIME		y			y		y	y	y	y
GPMC_CONFIG5_i	4-0	RDCYCLETIME	y		y	y		y		y	y	y
GPMC_CONFIG6_i	28-24	WRACCESSTIME		y			y		y	y	y	y
GPMC_CONFIG6_i	19-16	WRDATAONADMUXBUS		y			y		y		y	y

Table 9-41. Timing Parameters (continued)

Register	Bit	Bit Field Name	Asynchronous			Synchronous				Access Type		
			Single Read Access	Single Write Access	Multiple Read (Page) access	Single Read Access	Single Write Access	Multiple Read (Burst) Access	Multiple Write (Burst) Access	Non-multiplexed	Address /Data-multiplexed	AAD-multiplexed
GPMC_CONFIG6_i	11-8	CYCLE2CYCLEDELAY	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG6_i	7	CYCLE2CYCLESAMECSEN	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG6_i	6	CYCLE2CYCLEDIFFCSEN	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG6_i	3-0	BUSTURNAROUND	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG7_i	6	CSVALID	y	y	y	y	y	y	y	y	y	y

9.1.5 Use Cases

9.1.5.1 How to Set GPMC Timing Parameters for Typical Accesses

9.1.5.1.1 External Memory Attached to the GPMC Module

As discussed in the introduction to this chapter, the GPMC module supports the following external memory types:

- Asynchronous or synchronous, 8-bit or 16-bit-width memory or device
- 16-bit address/data-multiplexed or not multiplexed NOR flash device
- 8- or 16-bit NAND flash device

The following examples show how to calculate GPMC timing parameters by showing a typical parameter setup for the access to be performed.

The example is based on a 512-Mb multiplexed NOR flash memory with the following characteristics:

- Type: NOR flash (address/data-multiplexed mode)
- Size: 512M bits
- Data Bus: 16 bits wide
- Speed: 104 MHz clock frequency
- Read access time: 80 ns

9.1.5.1.2 Typical GPMC Setup

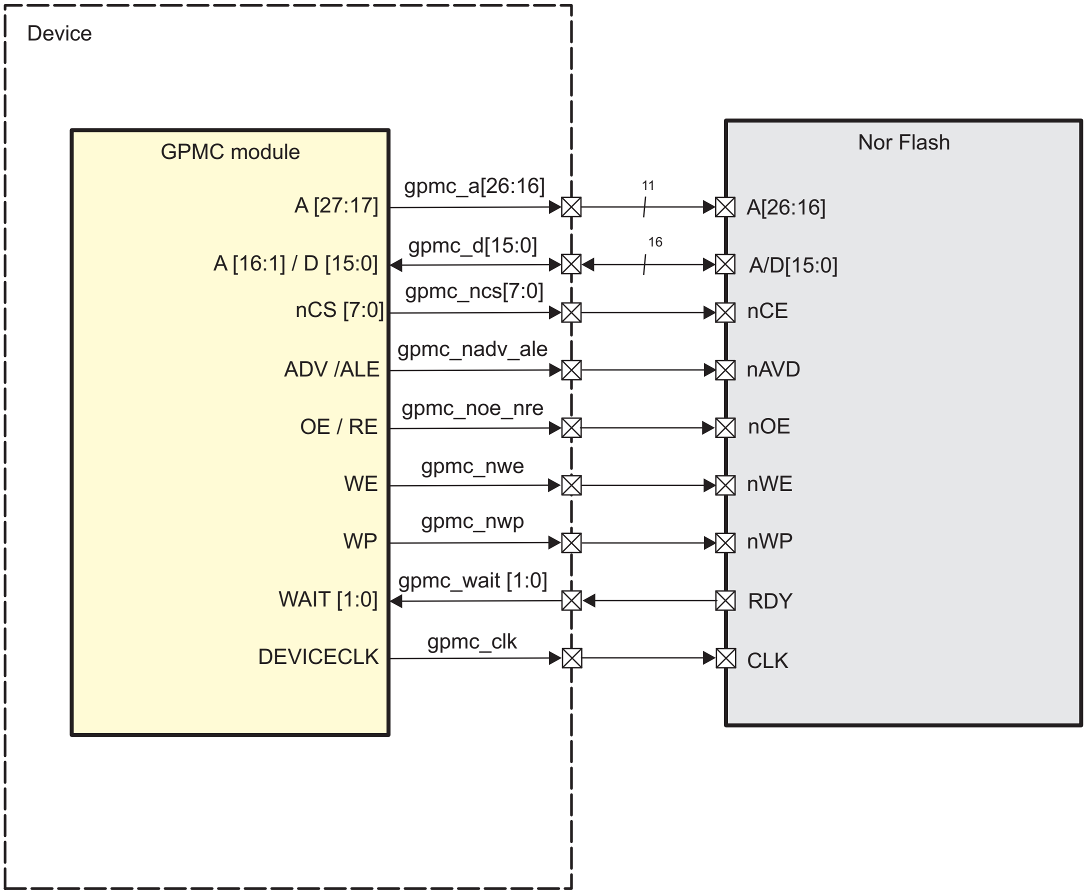
Table 9-42 lists some of the I/Os of the GPMC module.

Table 9-42. GPMC Signals

Signal Name	I/O	Description
GPMC_FCLK	Internal	Functional and interface clock. Acts as the time reference.
GPMC_CLK	O	External clock provided to the external device for synchronous operations
GPMC_A[27:17]	O	Address
GPMC_AD[15: 0]	I/O	Data-multiplexed with addresses A[16:1] on memory side
GPMC_CSxn	O	Chip-select (where x = 0, or 1)
GPMC_ADVn_ALE	O	Address valid enable
GPMC_OE_REn	O	Output enable (read access only)
GPMC_WEn	O	Write enable (write access only)
GPMC_WAIT[1:0]	I	Ready signal from memory device. Indicates when valid burst data is ready to be read

Figure 9-44 shows the typical connection between the GPMC module and an attached NOR Flash memory.

Figure 9-44. GPMC Connection to an External NOR Flash Memory



The following sections demonstrate how to calculate GPMC parameters for three access types:

- Synchronous burst read
- Asynchronous read
- Asynchronous single write

9.1.5.1.3 GPMC Configuration for Synchronous Burst Read Access

The clock runs at 104 MHz (f = 104 MHz; T = 9, 615 ns).

Table 9-43 shows the timing parameters (on the memory side) that determine the parameters on the GPMC side.

Table 9-44 shows how to calculate timings for the GPMC using the memory parameters.

Figure 9-45 shows the synchronous burst read access.

Table 9-43. Useful Timing Parameters on the Memory Side

AC Read Characteristics on the Memory Side	Description	Duration (ns)
tCES	CSn setup time to clock	0
tACS	Address setup time to clock	3
tIACC	Synchronous access time	80
tBACC	Burst access time valid clock to output delay	5,2
tCEZ	Chip-select to High-Impedance	7
tOEZ	Output enable to High-Impedance	7
tAVC	ADVn setup time	6
tAVD	AVDn pulse	6
tACH	Address hold time from clock	3

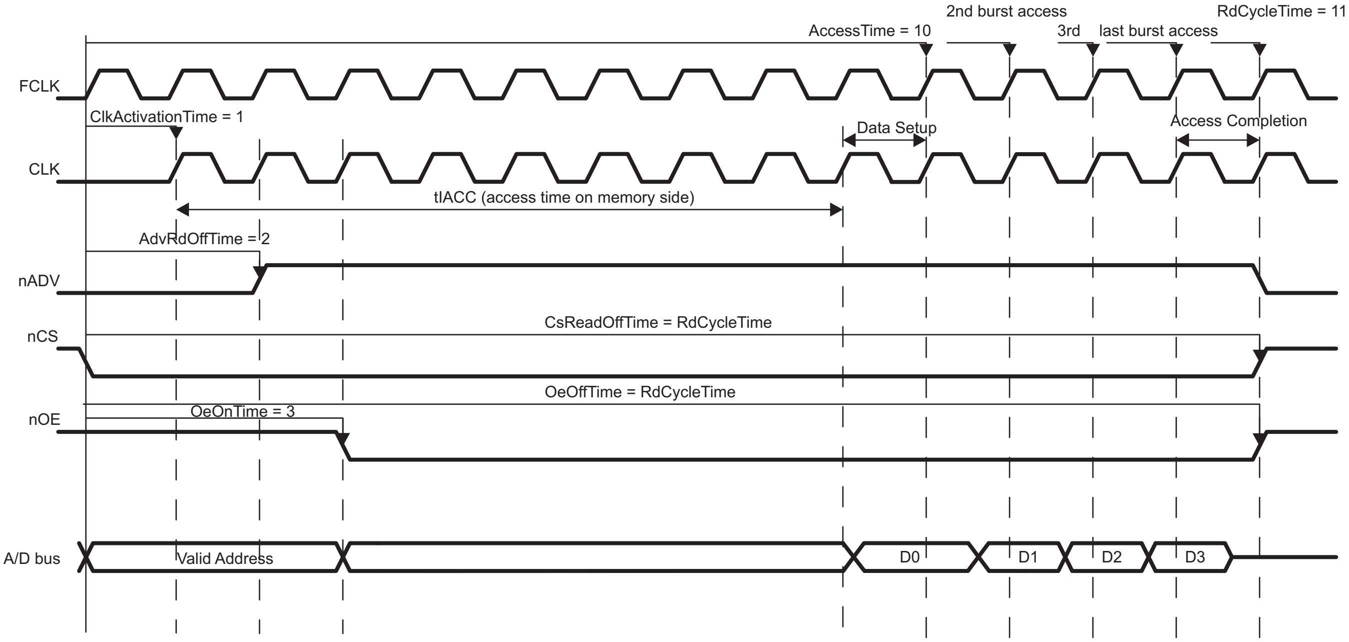
The following terms, which describe the timing interface between the controller and its attached device, are used to calculate the timing parameters on the GPMC side:

- Read Access time (GPMC side): Time required to activate the clock + read access time requested on the memory side + data setup time required for optimal capture of a burst of data
- Data setup time (GPMC side): Ensures a good capture of a burst of data (as opposed to taking a burst of data out). One word of data is processed in one clock cycle (T = 9,615 ns). The read access time between 2 bursts of data is tBACC = 5,2 ns. Therefore, data setup time is a clock period - tBACC = 4,415 ns of data setup.
- Access completion (GPMC side): (Different from page burst access time) Time required between the last burst access and access completion: CSn/OEn hold time (CSn and OEn must be released at the end of an access. These signals are held to allow the access to complete).
- Read cycle time (GPMC side): Read Access time + access completion
- Write cycle time for burst access: Not supported for NOR flash memory

Table 9-44. Calculating GPMC Timing Parameters

Parameter Name on GPMC Side	Formula	Duration (ns)	Number of Clock Cycles (F = 104 MHz)	GPMC Register Configurations
GPMC FCLK Divider	-	-	-	GPMCFCLKDIVIDER = 0
ClkActivationTime	min (tCES, tACS)	3	1	CLKACTIVATIONTIME = 1
RdAccessTime	roundmax (ClkActivationTime + tIACC + DataSetupTime)	94,03: (9,615 + 80 + 4,415)	10 : roundmax (94,03 / 9,615)	ACCESSTIME = Ah
PageBurstAccessTime	roundmax (tBACC)	roundmax (5,2)	1	PAGEBURSTACCESSTIME = 1
RdCycleTime	AccessTime + max (tCEZ, tOEZ)	101, 03: (94, 03 + 7)	11	RDCYCLETIME = Bh
CsOnTime	tCES	0	0	CSONTIME = 0
CsReadOffTime	RdCycleTime	-	11	CSRDOFFTIME = Bh
AdvOnTime	tAVC	0	0	ADVONTIME = 0
AdvRdOffTime	tAVD + tAVC	12	2	ADVRDOFFTIME = 2h
OeOnTime	(ClkActivationTime + tACH) < OeOnTime < (ClkActivationTime + tIACC)	-	3, for instance	OEONTIME = 3h
OeOffTime	RdCycleTime	-	11	OEOFFTIME = Bh

Figure 9-45. Synchronous Burst Read Access (Timing Parameters in Clock Cycles)



9.1.5.1.4 GPMC Configuration for Asynchronous Read Access

The clock runs at 104 MHz (f = 104 MHz; T = 9, 615 ns).

Table 9-45 shows the timing parameters (on the memory side) that determine the parameters on the GPMC side.

Table 9-46 shows how to calculate timings for the GPMC using the memory parameters.

Figure 9-46 shows the asynchronous read access.

Table 9-45. AC Characteristics for Asynchronous Read Access

AC Read Characteristics on the Memory Side	Description	Duration (ns)
tCE	Read Access time from CSn low	80
tAAVDS	Address setup time to rising edge of ADVn	3
tAVDP	ADVn low time	6
tCAS	CSn setup time to ADVn	0
tOE	Output enable to output valid	6
tOEZ	Output enable to High-Impedance	7

Use the following formula to calculate the RdCycleTime parameter for this typical access:

RdCycleTime = RdAccessTime + AccessCompletion = RdAccessTime + 1 clock cycle + tOEZ

- First, on the memory side, the external memory makes the data available to the output bus. This is the memory-side read access time defined in Table 9-46: the number of clock cycles between the address capture (ADVn rising edge) and the data valid on the output bus.

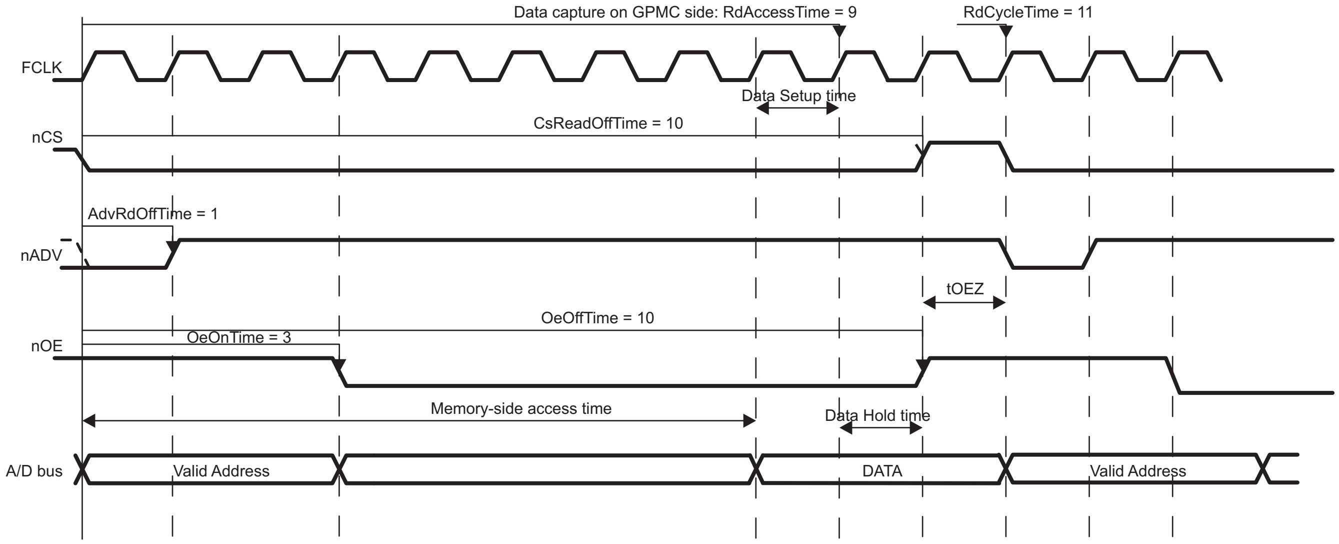
The GPMC requires some hold time to allow the data to be captured correctly and the access to be finished.

- To read the data correctly, the GPMC must be configure to meet the data setup time requirement of the memory; the GPMC module captures the data on the next rising edge. This is access time on the GPMC side.
- There must also be a data hold time for correctly reading the data (checking that there is no OEn/CSn deassertion while reading the data). This data hold time is 1 clock cycle (that is, AccessTime + 1).
- To complete the access, OEn/CSn signals are driven to high-impedance. AccessTime + 1 + tOEZ is the read cycle time.
- Addresses can now be relatched and a new read cycle begun.

Table 9-46. GPMC Timing Parameters for Asynchronous Read Access

Parameter Name on GPMC side	Formula	Duration (ns)	Number of Clock Cycles (F = 104 MHz)	GPMC Register Configurations
ClkActivationTime	n/a (asynchronous mode)			
AccessTime	round max (tCE)	80	9	ACCESSTIME = 9h
PageBurstAccessTime	n/a (single access)			
RdCycleTime	AccessTime + 1cycle + tOEZ	96, 615	11	RDCYCLETIME = Bh
CsOnTime	tCAS	0	0	CSONTIME = 0
CsReadOffTime	AccessTime + 1 cycle	89, 615	10	CSRDOFFTIME = Ah
AdvOnTime	tAAVDS	3	1	ADVONTIME = 1
AdvRdOffTime	tAAVDS + tAVDP	9	1	ADVRDOFFTIME = 1
OeOnTime	OeOnTime ≥ AdvRdOffTime (multiplexed mode)	-	3, for instance	OEONTIME = 3h
OeOffTime	AccessTime + 1cycle	89, 615	10	OEOFFTIME = Ah

Figure 9-46. Asynchronous Single Read Access (Timing Parameters in Clock Cycles)



9.1.5.1.5 GPMC Configuration for Asynchronous Single Write Access

The clock runs at 104 MHz: (f = 104 MHz; T = 9, 615 ns).

Table 9-47 shows how to calculate timings for the GPMC using the memory parameters.

Table 9-48 shows the timing parameters (on the memory side) that determine the parameters on the GPMC side.

Figure 9-47 shows the synchronous burst write access.

Table 9-47. AC Characteristics for Asynchronous Single Write (Memory Side)

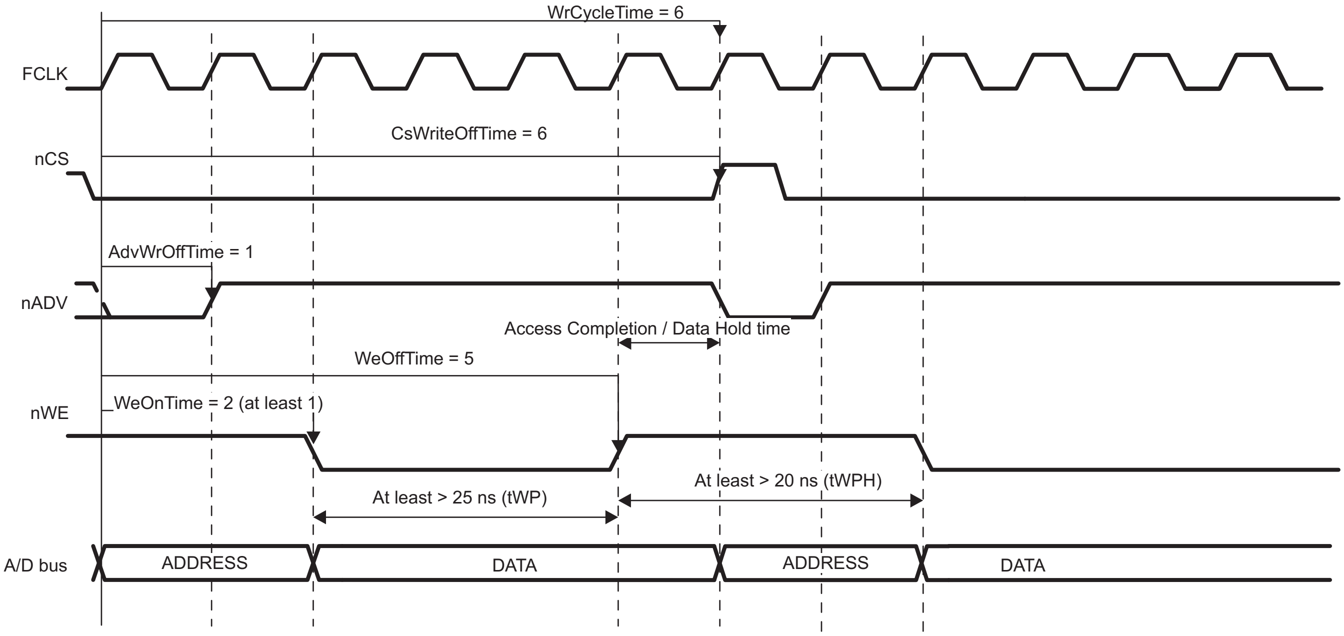
AC Characteristics on the Memory Side	Description	Duration (ns)
tWC	Write cycle time	60
tAVDP	ADVn low time	6
tWP	Write pulse width	25
tWPH	Write pulse width high	20
tCS	CSn setup time to WEn	3
tCAS	CSn setup time to ADVn	0
tAVSC	ADVn setup time	3

For asynchronous single write access, write cycle time is WrCycleTime = WeOffTime + AccessCompletion = WeOffTime + 1. For the AccesCompletion, the GPMC requires 1 cycle of data hold time (CSn de-assertion).

Table 9-48. GPMC Timing Parameters for Asynchronous Single Write

Parameter Name on GPMC side	Formula	Duration (ns)	Number of Clock Cycles (F = 104 MHz)	GPMC Register Configurations
ClkActivationTime	n/a (asynchronous mode)			
AccessTime	Applicable only to WAITMONITORING (the value is the same as for read access)			
PageBurstAccessTime	n/a (single access)			
WrCycleTime	WeOffTime + AccessCompletion	57, 615	6	WRCYCLETIME = 6h
CsOnTime	tCAS	0	0	CSONTIME = 0
CsWrOffTime	WeOffTime + 1	57, 615	6	CSWROFFTIME = 6h
AdvOnTime	tAVSC	3	1	ADVONTIME = 1
AdvWrOffTime	tAVSC + tAVDP	9	1	ADVWROFFTIME = 1
WeOnTime	tCS	3	1	WEONTIME = 1
WeOffTime	tCS + tWP + tWPH	48	5	WEOFFTIME = 5h

Figure 9-47. Asynchronous Single Write Access (Timing Parameters in Clock Cycles)



9.1.5.2 How to Choose a Suitable Memory to Use With the GPMC

This section is intended to help the user select a suitable memory device to interface with the GPMC controller.

9.1.5.2.1 Supported Memories or Devices

NAND flash and NOR flash architectures are the two flash technologies. The GPMC supports various types of external memory or device, basically any one that supports NAND or NOR protocols:

- 8- and 16-bit width asynchronous or synchronous memory or device (8-bit: non burst device only)
- 16-bit address and data multiplexed NOR flash devices (pSRAM)
- 8- and 16-bit NAND flash device

9.1.5.2.2 NAND Interface Protocol

NAND flash architecture, introduced in 1989, is a flash technology. NAND is a page-oriented memory device, that is, read and write accesses are done by pages. NAND achieves great density by sharing common areas of the storage transistor, which creates strings of serially connected transistors (in NOR devices, each transistor stands alone). Thanks to its high density NAND is best suited to devices requiring high capacity data storage, such as pictures, music, or data files. NAND non-volatility, makes of it a good storage solution for many applications where mobility, low power, and speed are key factors. Low pin count and simple interface are other advantages of NAND.

Table 9-49 summarizes the NAND interface signals level applied to external device or memories.

Table 9-49. NAND Interface Bus Operations Summary

Bus Operation	CLE	ALE	CEn	WEn	REn	WPn
Read (cmd input)	H	L	L	RE	H	x
Read (add input)	L	H	L	RE	H	x
Write (cmd input)	H	L	L	RE	H	H
Write (add input)	L	H	L	RE	H	H
Data input	L	L	L	RE	H	H
Data output	L	L	L	H	FE	x
Busy (during read)	x	x	H	H	H	x
Busy (during program)	x	x	x	x	x	H
Busy (during erase)	x	x	x	x	x	H
Write protect	x	x	x	x	x	L
Stand-by	x	x	H	x	x	H/L

9.1.5.2.3 NOR Interface Protocol

NOR flash architecture, introduced in 1988, is a flash technology. Unlike NAND, which is a sequential access device, NOR is directly addressable; i.e., it is designed to be a random access device. NOR is best suited to devices used to store and run code or firmware, usually in small capacities. While NOR has fast read capabilities it has slow write and erase functions compared to NAND architecture.

Table 9-50 summarizes the NOR interface signals level applied to external device or memories.

Table 9-50. NOR Interface Bus Operations Summary

Bus Operation	CLK	ADVn	CSn	OEn	WEn	WAIT	DQ[15:0]
Read (asynchronous)	x	L	L	L	H	Asserted	Output
Read (synchronous)	Running	L	L	L	H	Driven	Output
Read (burst suspend)	Halted	x	L	H	H	Active	Output
Write	x	L	L	H	L	Asserted	Input

Table 9-50. NOR Interface Bus Operations Summary (continued)

Bus Operation	CLK	ADVn	CSn	OEn	WEn	WAIT	DQ[15:0]
Output disable	x	x	L	H	H	Asserted	High-Z
Standby	x	x	H	x	x	High-Z	High-Z

9.1.5.2.4 Other Technologies

Other supported device type interact with the GPMC through the NOR interface protocol.

OneNAND Flash is a high-density and low-power memory device. It is based on single- or multi-level-cell NAND core with SRAM and logic, and interfaces as a synchronous NOR Flash, plus has synchronous write capability. It reads faster than conventional NAND and writes faster than conventionnal NOR flash. Hence, it is appropriate for both mass storage and code storage.

pSRAM stands for pseudo-static random access memory. pSRAM is a low-power memory device for mobile applications. pSRAM is based on the DRAM cell with internal refresh and address control features, and interfaces as a synchronous NOR Flash, plus has synchronous write capability.

9.1.5.2.5 Supported Protocols

The GPMC supports the following interface protocols when communicating with external memory or external devices:

- Asynchronous read/write access
- Asynchronous read page access (4-8-16 Word16)
- Synchronous read/write access
- Synchronous read burst access without wrap capability (4-8-16 Word16)
- Synchronous read burst access with wrap capability (4-8-16 Word16)

9.1.5.2.6 GPMC Features and Settings

This section lists GPMC features and settings:

- Supported device type: up to four NAND or NOR protocol external memories or devices
- Operating Voltage: 3.3V
- Maximum GPMC addressing capability: 512 MBytes divided into eight chip-selects
- Maximum supported memory size: 256 MBytes (must be a power-of-2)
- Minimum supported memory size: 16 MBytes (must be a power-of-2). Aliasing occurs when addressing smaller memories.
- Data path to external memory or device: 8- and 16-bit wide
- Burst and page access: burst of 4-8-16 Word16
- Supports bus keeping
- Supports bus turn around

9.1.6 GPMC Registers

Table 9-51 lists the memory-mapped registers for the GPMC. All register offset addresses not listed in Table 9-51 should be considered as reserved locations and the register contents should not be modified.

Table 9-51. GPMC Registers

Offset	Acronym	Register Name	Section
0h	GPMC_REVISION		Section 9.1.6.1
10h	GPMC_SYSCONFIG		Section 9.1.6.2
14h	GPMC_SYSSTATUS		Section 9.1.6.3
18h	GPMC_IRQSTATUS		Section 9.1.6.4
1Ch	GPMC_IRQENABLE		Section 9.1.6.5
40h	GPMC_TIMEOUT_CONTROL		Section 9.1.6.6
44h	GPMC_ERR_ADDRESS		Section 9.1.6.7
48h	GPMC_ERR_TYPE		Section 9.1.6.8
50h	GPMC_CONFIG		Section 9.1.6.9
54h	GPMC_STATUS		Section 9.1.6.10

Table 9-51. GPMC Registers (continued)

Offset	Acronym	Register Name	Section
60h	GPMC_CONFIG1_0		Section 9.1.6.11
64h	GPMC_CONFIG2_0		Section 9.1.6.12
68h	GPMC_CONFIG3_0		Section 9.1.6.13
6Ch	GPMC_CONFIG4_0		Section 9.1.6.14
70h	GPMC_CONFIG5_0		Section 9.1.6.15
74h	GPMC_CONFIG6_0		Section 9.1.6.16
78h	GPMC_CONFIG7_0		Section 9.1.6.17
7Ch	GPMC_NAND_COMMAND_0		Section 9.1.6.18
80h	GPMC_NAND_ADDRESS_0		Section 9.1.6.19
84h	GPMC_NAND_DATA_0		Section 9.1.6.20
0h	GPMC_CONFIG1_1		Section 9.1.6.11
0h	GPMC_CONFIG2_1		Section 9.1.6.12
0h	GPMC_CONFIG3_1		Section 9.1.6.13
0h	GPMC_CONFIG4_1		Section 9.1.6.14
0h	GPMC_CONFIG5_1		Section 9.1.6.15
0h	GPMC_CONFIG6_1		Section 9.1.6.16
0h	GPMC_CONFIG7_1		Section 9.1.6.17
0h	GPMC_NAND_COMMAND_1		Section 9.1.6.18
0h	GPMC_NAND_ADDRESS_1		Section 9.1.6.19
0h	GPMC_NAND_DATA_1		Section 9.1.6.20
0h	GPMC_CONFIG1_2		Section 9.1.6.11
0h	GPMC_CONFIG2_2		Section 9.1.6.12
0h	GPMC_CONFIG3_2		Section 9.1.6.13
0h	GPMC_CONFIG4_2		Section 9.1.6.14
0h	GPMC_CONFIG5_2		Section 9.1.6.15
0h	GPMC_CONFIG6_2		Section 9.1.6.16
0h	GPMC_CONFIG7_2		Section 9.1.6.17
0h	GPMC_NAND_COMMAND_2		Section 9.1.6.18
0h	GPMC_NAND_ADDRESS_2		Section 9.1.6.19
0h	GPMC_NAND_DATA_2		Section 9.1.6.20
0h	GPMC_CONFIG1_3		Section 9.1.6.11
0h	GPMC_CONFIG2_3		Section 9.1.6.12
0h	GPMC_CONFIG3_3		Section 9.1.6.13
0h	GPMC_CONFIG4_3		Section 9.1.6.14
0h	GPMC_CONFIG5_3		Section 9.1.6.15
0h	GPMC_CONFIG6_3		Section 9.1.6.16
0h	GPMC_CONFIG7_3		Section 9.1.6.17
0h	GPMC_NAND_COMMAND_3		Section 9.1.6.18
0h	GPMC_NAND_ADDRESS_3		Section 9.1.6.19
0h	GPMC_NAND_DATA_3		Section 9.1.6.20
0h	GPMC_CONFIG1_4		Section 9.1.6.11
0h	GPMC_CONFIG2_4		Section 9.1.6.12
0h	GPMC_CONFIG3_4		Section 9.1.6.13
0h	GPMC_CONFIG4_4		Section 9.1.6.14
0h	GPMC_CONFIG5_4		Section 9.1.6.15
0h	GPMC_CONFIG6_4		Section 9.1.6.16
0h	GPMC_CONFIG7_4		Section 9.1.6.17

Table 9-51. GPMC Registers (continued)

Offset	Acronym	Register Name	Section
0h	GPMC_NAND_COMMAND_4		Section 9.1.6.18
0h	GPMC_NAND_ADDRESS_4		Section 9.1.6.19
0h	GPMC_NAND_DATA_4		Section 9.1.6.20
0h	GPMC_CONFIG1_5		Section 9.1.6.11
0h	GPMC_CONFIG2_5		Section 9.1.6.12
0h	GPMC_CONFIG3_5		Section 9.1.6.13
0h	GPMC_CONFIG4_5		Section 9.1.6.14
0h	GPMC_CONFIG5_5		Section 9.1.6.15
0h	GPMC_CONFIG6_5		Section 9.1.6.16
0h	GPMC_CONFIG7_5		Section 9.1.6.17
0h	GPMC_NAND_COMMAND_5		Section 9.1.6.18
0h	GPMC_NAND_ADDRESS_5		Section 9.1.6.19
0h	GPMC_NAND_DATA_5		Section 9.1.6.20
0h	GPMC_CONFIG1_6		Section 9.1.6.11
0h	GPMC_CONFIG2_6		Section 9.1.6.12
0h	GPMC_CONFIG3_6		Section 9.1.6.13
0h	GPMC_CONFIG4_6		Section 9.1.6.14
0h	GPMC_CONFIG5_6		Section 9.1.6.15
0h	GPMC_CONFIG6_6		Section 9.1.6.16
0h	GPMC_CONFIG7_6		Section 9.1.6.17
0h	GPMC_NAND_COMMAND_6		Section 9.1.6.18
0h	GPMC_NAND_ADDRESS_6		Section 9.1.6.19
0h	GPMC_NAND_DATA_6		Section 9.1.6.20
1E0h	GPMC_PREFETCH_CONFIG1		Section 9.1.6.21
1E4h	GPMC_PREFETCH_CONFIG2		Section 9.1.6.22
1ECh	GPMC_PREFETCH_CONTROL		Section 9.1.6.23
1F0h	GPMC_PREFETCH_STATUS		Section 9.1.6.24
1F4h	GPMC_ECC_CONFIG		Section 9.1.6.25
1F8h	GPMC_ECC_CONTROL		Section 9.1.6.26
1FCh	GPMC_ECC_SIZE_CONFIG		Section 9.1.6.27
200h	GPMC_ECC1_RESULT		Section 9.1.6.28
204h	GPMC_ECC2_RESULT		Section 9.1.6.29
208h	GPMC_ECC3_RESULT		Section 9.1.6.30
20Ch	GPMC_ECC4_RESULT		Section 9.1.6.31
210h	GPMC_ECC5_RESULT		Section 9.1.6.32
214h	GPMC_ECC6_RESULT		Section 9.1.6.33
218h	GPMC_ECC7_RESULT		Section 9.1.6.34
21Ch	GPMC_ECC8_RESULT		Section 9.1.6.35
220h	GPMC_ECC9_RESULT		Section 9.1.6.36
240h	GPMC_BCH_RESULT0_0		Section 9.1.6.37
244h	GPMC_BCH_RESULT1_0		Section 9.1.6.38
248h	GPMC_BCH_RESULT2_0		Section 9.1.6.39
24Ch	GPMC_BCH_RESULT3_0		Section 9.1.6.40
250h	GPMC_BCH_RESULT0_1		Section 9.1.6.41
254h	GPMC_BCH_RESULT1_1		Section 9.1.6.42
258h	GPMC_BCH_RESULT2_1		Section 9.1.6.43
25Ch	GPMC_BCH_RESULT3_1		Section 9.1.6.44

Table 9-51. GPMC Registers (continued)

Offset	Acronym	Register Name	Section
260h	GPMC_BCH_RESULT0_2		Section 9.1.6.45
264h	GPMC_BCH_RESULT1_2		Section 9.1.6.46
268h	GPMC_BCH_RESULT2_2		Section 9.1.6.47
26Ch	GPMC_BCH_RESULT3_2		Section 9.1.6.48
270h	GPMC_BCH_RESULT0_3		Section 9.1.6.49
274h	GPMC_BCH_RESULT1_3		Section 9.1.6.50
278h	GPMC_BCH_RESULT2_3		Section 9.1.6.51
27Ch	GPMC_BCH_RESULT3_3		Section 9.1.6.52
280h	GPMC_BCH_RESULT0_4		Section 9.1.6.53
284h	GPMC_BCH_RESULT1_4		Section 9.1.6.54
288h	GPMC_BCH_RESULT2_4		Section 9.1.6.55
28Ch	GPMC_BCH_RESULT3_4		Section 9.1.6.56
290h	GPMC_BCH_RESULT0_5		Section 9.1.6.57
294h	GPMC_BCH_RESULT1_5		Section 9.1.6.58
298h	GPMC_BCH_RESULT2_5		Section 9.1.6.59
29Ch	GPMC_BCH_RESULT3_5		Section 9.1.6.60
2A0h	GPMC_BCH_RESULT0_6		Section 9.1.6.61
2A4h	GPMC_BCH_RESULT1_6		Section 9.1.6.62
2A8h	GPMC_BCH_RESULT2_6		Section 9.1.6.63
2ACh	GPMC_BCH_RESULT3_6		Section 9.1.6.64
2D0h	GPMC_BCH_SWDATA		Section 9.1.6.65
300h	GPMC_BCH_RESULT4_0		Section 9.1.6.66
304h	GPMC_BCH_RESULT5_0		Section 9.1.6.67
308h	GPMC_BCH_RESULT6_0		Section 9.1.6.68
310h	GPMC_BCH_RESULT4_1		Section 9.1.6.69
314h	GPMC_BCH_RESULT5_1		Section 9.1.6.70
318h	GPMC_BCH_RESULT6_1		Section 9.1.6.71
320h	GPMC_BCH_RESULT4_2		Section 9.1.6.72
324h	GPMC_BCH_RESULT5_2		Section 9.1.6.73
328h	GPMC_BCH_RESULT6_2		Section 9.1.6.74
330h	GPMC_BCH_RESULT4_3		Section 9.1.6.75
334h	GPMC_BCH_RESULT5_3		Section 9.1.6.76
338h	GPMC_BCH_RESULT6_3		Section 9.1.6.77
340h	GPMC_BCH_RESULT4_4		Section 9.1.6.78
344h	GPMC_BCH_RESULT5_4		Section 9.1.6.79
348h	GPMC_BCH_RESULT6_4		Section 9.1.6.80
350h	GPMC_BCH_RESULT4_5		Section 9.1.6.81
354h	GPMC_BCH_RESULT5_5		Section 9.1.6.82
358h	GPMC_BCH_RESULT6_5		Section 9.1.6.83
360h	GPMC_BCH_RESULT4_6		Section 9.1.6.84
364h	GPMC_BCH_RESULT5_6		Section 9.1.6.85
368h	GPMC_BCH_RESULT6_6		Section 9.1.6.86
4B0h	GPMC_BCH_RESULT0_7		Section 9.1.6.87
4B4h	GPMC_BCH_RESULT1_7		Section 9.1.6.88
4B8h	GPMC_BCH_RESULT2_7		Section 9.1.6.89
4BCh	GPMC_BCH_RESULT3_7		Section 9.1.6.90
570h	GPMC_BCH_RESULT4_7		Section 9.1.6.91

Table 9-51. GPMC Registers (continued)

Offset	Acronym	Register Name	Section
574h	GPMC_BCH_RESULT5_7		Section 9.1.6.92
578h	GPMC_BCH_RESULT6_7		Section 9.1.6.93

9.1.6.1 GPMC_REVISION Register (Offset = 0h) [reset = 0h]

GPMC_REVISION is shown in Figure 9-48 and described in Table 9-52.

Return to Summary Table.

The GPMC_REVISION register contains the IP revision code.

Figure 9-48. GPMC_REVISION Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								REV							
R-0h																								R-0h							

Table 9-52. GPMC_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	REV	R	0h	IP revision. Major revision is [7:4]. Minor revision is [3:0]. Examples: 10h for revision 1.0, 21h for revision 2.1.

9.1.6.2 GPMC_SYSCONFIG Register (Offset = 10h) [reset = 0h]

GPMC_SYSCONFIG is shown in Figure 9-49 and described in Table 9-53.

Return to Summary Table.

The GPMC_SYSCONFIG register controls the various parameters of the OCP interface.

Figure 9-49. GPMC_SYSCONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED			SIDLEMODE		RESERVED	SOFTRESET	AUTOIDLE
R-0h			R/W-0h		R-0h	R/W-0h	R/W-0h

Table 9-53. GPMC_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4-3	SIDLEMODE	R/W	0h	Idle mode 0h (R/W) = Force-idle. An idle request is acknowledged unconditionally 1h (R/W) = No-idle. An idle request is never acknowledged 2h (R/W) = Smart-idle. Acknowledgment to an idle request is given based on the internal activity of the module 3h (R/W) = Reserved
2	RESERVED	R	0h	
1	SOFTRESET	R/W	0h	Software reset (Set 1 to this bit triggers a module reset. This bit is automatically reset by hardware. During reads, it always returns 0). 0h (R/W) = Normal mode 1h (R/W) = The module is reset
0	AUTOIDLE	R/W	0h	Internal OCP clock gating strategy. 0h (R/W) = Interface clock is free-running 1h (R/W) = Automatic interface clock gating strategy is applied based on the Interconnect activity.

9.1.6.3 GPMC_SYSSTATUS Register (Offset = 14h) [reset = 0h]

GPMC_SYSSTATUS is shown in Figure 9-50 and described in Table 9-54.

Return to Summary Table.

The GPMC_SYSSTATUS register provides status information about the module, excluding the interrupt status information.

Figure 9-50. GPMC_SYSSTATUS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="7">RESERVED</td><td>RESETDONE</td></tr>
<tr><td colspan="7">R-0h</td><td>R-0h</td></tr>
</table>

Table 9-54. GPMC_SYSSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	RESETDONE	R	0h	Internal reset monitoring 0h (R) = Internal module reset in on-going 1h (R) = Reset completed

9.1.6.4 GPMC_IRQSTATUS Register (Offset = 18h) [reset = 0h]

GPMC_IRQSTATUS is shown in Figure 9-51 and described in Table 9-55.

Return to Summary Table.

The GPMC_IRQSTATUS interrupt status register regroups all the status of the module internal events that can generate an interrupt.

Figure 9-51. GPMC_IRQSTATUS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="6">RESERVED</td><td>WAIT1EDGEDETECTIONSTATUS</td><td>WAIT0EDGEDETECTIONSTATUS</td></tr>
<tr><td colspan="6">R-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>TERMINALCOUNTSTATUS</td><td>FIFOEVENTSTATUS</td></tr>
<tr><td colspan="6">R-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 9-55. GPMC_IRQSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9	WAIT1EDGEDETECTIONSTATUS	R/W	0h	Status of the Wait1 Edge Detection interrupt 0h (W) = WAIT1EDGEDETECTIONSTATUS bit unchanged 0h (R) = A transition on WAIT1 input pin has not been detected 1h (R) = A transition on WAIT1 input pin has been detected 1h (W) = WAIT1EDGEDETECTIONSTATUS bit is reset
8	WAIT0EDGEDETECTIONSTATUS	R/W	0h	Status of the Wait0 Edge Detection interrupt 0h (W) = WAIT0EDGEDETECTIONSTATUS bit unchanged 0h (R) = A transition on WAIT0 input pin has not been detected 1h (R) = A transition on WAIT0 input pin has been detected 1h (W) = WAIT0EDGEDETECTIONSTATUS bit is reset
7-2	RESERVED	R	0h	
1	TERMINALCOUNTSTATUS	R/W	0h	Status of the TerminalCountEvent interrupt 0h (R/W) = TERMINALCOUNTSTATUS bit unchanged 1h (R/W) = Indicates that CountValue is equal to 0
0	FIFOEVENTSTATUS	R/W	0h	Status of the FIFOEvent interrupt 0h (W) = FIFOEVENTSTATUS bit unchanged 0h (R) = Indicates than less than GPMC_PREFETCH_STATUS[16] FIFOTHRESHOLDSTATUS bytes are available in prefetch mode and less than FIFOTHRESHOLD bytes free places are available in write-posting mode. 1h (W) = FIFOEVENTSTATUS bit is reset 1h (R) = Indicates than at least GPMC_PREFETCH_STATUS[16] FIFOTHRESHOLDSTATUS bytes are available in prefetch mode and at least FIFOTHRESHOLD bytes free places are available in write-posting mode.

9.1.6.5 GPMC_IRQENABLE Register (Offset = 1Ch) [reset = 0h]

GPMC_IRQENABLE is shown in Figure 9-52 and described in Table 9-56.

Return to Summary Table.

The GPMC_IRQENABLE interrupt enable register allows to mask/unmask the module internal sources of interrupt, on a event-by-event basis.

Figure 9-52. GPMC_IRQENABLE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED						WAIT1EDGEDETECTIONENABLE	WAIT0EDGEDETECTIONENABLE
R-0h						R/W-0h	R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						TERMINALCOUNTEVENTENABLE	FIFOEVENTENABLE
R-0h						R/W-0h	R/W-0h

Table 9-56. GPMC_IRQENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9	WAIT1EDGEDETECTIONENABLE	R/W	0h	Enables the Wait1 Edge Detection interrupt 0h (R/W) = Wait1EdgeDetection interrupt is masked 1h (R/W) = Wait1EdgeDetection event generates an interrupt if occurs
8	WAIT0EDGEDETECTIONENABLE	R/W	0h	Enables the Wait0 Edge Detection interrupt 0h (R/W) = Wait0EdgeDetection interrupt is masked 1h (R/W) = Wait0EdgeDetection event generates an interrupt if occurs
7-2	RESERVED	R	0h	
1	TERMINALCOUNTEVENTENABLE	R/W	0h	Enables TerminalCountEvent interrupt issuing in pre-fetch or write posting mode 0h (R/W) = TerminalCountEvent interrupt is masked 1h (R/W) = TerminalCountEvent interrupt is not masked
0	FIFOEVENTENABLE	R/W	0h	Enables the FIFOEvent interrupt 0h (R/W) = FIFOEvent interrupt is masked 1h (R/W) = FIFOEvent interrupt is not masked

9.1.6.6 GPMC_TIMEOUT_CONTROL Register (Offset = 40h) [reset = 0h]

GPMC_TIMEOUT_CONTROL is shown in Figure 9-53 and described in Table 9-57.

Return to Summary Table.

The GPMC_TIMEOUT_CONTROL register allows the user to set the start value of the timeout counter

Figure 9-53. GPMC_TIMEOUT_CONTROL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED			TIMEOUTSTARTVALUE				
R-0h			R/W-0h				
7	6	5	4	3	2	1	0
TIMEOUTSTARTVALUE				RESERVED			TIMEOUTENABLE
R/W-0h				R-0h			R/W-0h

Table 9-57. GPMC_TIMEOUT_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12-4	TIMEOUTSTARTVALUE	R/W	0h	Start value of the time-out counter (000 corresponds to 0 GPMC.FCLK cycle, 1h corresponds to 1 GPMC.FCLK cycle, and 1FFh corresponds to 511 GPMC.FCLK cycles)
3-1	RESERVED	R	0h	
0	TIMEOUTENABLE	R/W	0h	Enable bit of the TimeOut feature 0h (R/W) = TimeOut feature is disabled 1h (R/W) = TimeOut feature is enabled

9.1.6.7 GPMC_ERR_ADDRESS Register (Offset = 44h) [reset = 0h]

GPMC_ERR_ADDRESS is shown in Figure 9-54 and described in Table 9-58.

Return to Summary Table.

The GPMC_ERR_ADDRESS register stores the address of the illegal access when an error occurs.

Figure 9-54. GPMC_ERR_ADDRESS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>RESERVED</td><td colspan="7">ILLEGALADD</td></tr>
<tr><td>R-0h</td><td colspan="7">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">ILLEGALADD</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">ILLEGALADD</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">ILLEGALADD</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 9-58. GPMC_ERR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-0	ILLEGALADD	R/W	0h	Address of illegal access: A30 (0 for memory region, 1 for GPMC register region) and A29 to A0 (1GByte maximum)

9.1.6.8 GPMC_ERR_TYPE Register (Offset = 48h) [reset = 0h]

GPMC_ERR_TYPE is shown in Figure 9-55 and described in Table 9-59.

Return to Summary Table.

The GPMC_ERR_TYPE register stores the type of error when an error occurs.

Figure 9-55. GPMC_ERR_TYPE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED					ILLEGALMCMD		
R-0h					R/W-0h		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED			ERRORNOTSUPPADD	ERRORNOTSUPPMCMD	ERRORTIMEOUT	RESERVED	ERRORVALID
R-0h			R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h

Table 9-59. GPMC_ERR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10-8	ILLEGALMCMD	R/W	0h	System Command of the transaction that caused the error
7-5	RESERVED	R	0h	
4	ERRORNOTSUPPADD	R/W	0h	Not supported Address error 0h (R/W) = No error occurs 1h (R/W) = The error is due to a non supported Address
3	ERRORNOTSUPPMCMD	R/W	0h	Not supported Command error 0h (R/W) = No error occurs 1h (R/W) = The error is due to a non supported Command
2	ERRORTIMEOUT	R/W	0h	Time-out error 0h (R/W) = No error occurs 1h (R/W) = The error is due to a time out
1	RESERVED	R	0h	
0	ERRORVALID	R/W	0h	Error validity status - Must be explicitly cleared with a write 1 transaction 0h (R/W) = All error fields no longer valid 1h (R/W) = Error detected and logged in the other error fields

9.1.6.9 GPMC_CONFIG Register (Offset = 50h) [reset = 0h]

GPMC_CONFIG is shown in Figure 9-56 and described in Table 9-60.

Return to Summary Table.

The configuration register allows global configuration of the GPMC.

Figure 9-56. GPMC_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED						WAIT1PINPOLARITY	WAIT0PINPOLARITY
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED			WRITEPROTECT	RESERVED		LIMITEDADDRESS	NANDFORCEPOSTEDWRITE
R-0h			R/W-0h	R-0h		R/W-0h	R/W-0h

Table 9-60. GPMC_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9	WAIT1PINPOLARITY	R/W	0h	Selects the polarity of input pin WAIT1. 0h (R/W) = WAIT1 active low 1h (R/W) = WAIT1 active high
8	WAIT0PINPOLARITY	R/W	0h	Selects the polarity of input pin WAIT0. 0h (R/W) = WAIT0 active low 1h (R/W) = WAIT0 active high
7-5	RESERVED	R	0h	
4	WRITEPROTECT	R/W	0h	Controls the WP output pin level. 0h (R/W) = WP output pin is low 1h (R/W) = WP output pin is high
3-2	RESERVED	R	0h	
1	LIMITEDADDRESS	R/W	0h	Limited Address device support. 0h (R/W) = No effect. GPMC controls all addresses. 1h (R/W) = A26-A11 are not modified during an external memory access.
0	NANDFORCEPOSTEDWRITE	R/W	0h	0h (R/W) = Disables Force Posted Write 1h (R/W) = Enables Force Posted Write

9.1.6.10 GPMC_STATUS Register (Offset = 54h) [reset = 0h]

GPMC_STATUS is shown in Figure 9-57 and described in Table 9-61.

Return to Summary Table.

The status register provides global status bits of the GPMC.

Figure 9-57. GPMC_STATUS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED						WAIT1STATUS	WAIT0STATUS
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED							EMPTYWRITEBUFFERSTATUS
R-0h							R/W-0h

Table 9-61. GPMC_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9	WAIT1STATUS	R/W	0h	Is a copy of input pin WAIT1. (Reset value is WAIT1 input pin sampled at IC reset) 0h (R/W) = WAIT1 asserted (inactive state) 1h (R/W) = WAIT1 de-asserted
8	WAIT0STATUS	R/W	0h	Is a copy of input pin WAIT0. (Reset value is WAIT0 input pin sampled at IC reset) 0h (R/W) = WAIT0 asserted (inactive state) 1h (R/W) = WAIT0 de-asserted
7-1	RESERVED	R	0h	
0	EMPTYWRITEBUFFERSTATUS	R/W	0h	Stores the empty status of the write buffer 0h (R/W) = Write Buffer is not empty 1h (R/W) = Write Buffer is empty

9.1.6.11 GPMC_CONFIG1_0 to GPMC_CONFIG1_6 Register (Offset = 60h + [i * 0h], where i = 0 to 6) [reset = 0h]

GPMC_CONFIG1_0 to GPMC_CONFIG1_6 is shown in Figure 9-58 and described in Table 9-62.

Return to Summary Table.

The configuration 1 register sets signal control parameters per chip select.

Figure 9-58. GPMC_CONFIG1_0 to GPMC_CONFIG1_6 Register

31	30	29	28	27	26	25	24
WRAPBURST	READMULTIPLE	READTYPE	WRITEMULTIPLE	WRITETYPE	CLKACTIVATIONTIME		ATTACHEDDEVICEPAGELENGTH
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h

23	22	21	20	19	18	17	16
ATTACHEDDEVICEPAGELENGTH	WAITREADMONITORING	WAITWRITEMONITORING	RESERVED	WAITMONITORINGTIME		WAITPINSELECT	
R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h		R/W-0h	

15	14	13	12	11	10	9	8
RESERVED		DEVICESIZE		DEVICETYPE		MUXADDDATA	
R-0h		R/W-0h		R/W-0h		R/W-0h	

7	6	5	4	3	2	1	0
RESERVED			TIMEPARAGRANULARITY	RESERVED		GPMCFCLKDIVIDER	
R-0h			R/W-0h	R-0h		R/W-0h	

Table 9-62. GPMC_CONFIG1_0 to GPMC_CONFIG1_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	WRAPBURST	R/W	0h	Enables the wrapping burst capability. Must be set if the attached device is configured in wrapping burst 0h (R/W) = Synchronous wrapping burst not supported 1h (R/W) = Synchronous wrapping burst supported
30	READMULTIPLE	R/W	0h	Selects the read single or multiple access 0h (R/W) = single access 1h (R/W) = multiple access (burst if synchronous, page if asynchronous)
29	READTYPE	R/W	0h	Selects the read mode operation 0h (R/W) = Read Asynchronous 1h (R/W) = Read Synchronous
28	WRITEMULTIPLE	R/W	0h	Selects the write single or multiple access 0h (R/W) = Single access 1h (R/W) = Multiple access (burst if synchronous, considered as single if asynchronous)
27	WRITETYPE	R/W	0h	Selects the write mode operation 0h (R/W) = Write Asynchronous 1h (R/W) = Write Synchronous
26-25	CLKACTIVATIONTIME	R/W	0h	Output GPMC.CLK activation time 0h (R/W) = First rising edge of GPMC_CLK at start access time 1h (R/W) = First rising edge of GPMC_CLK one GPMC_FCLK cycle after start access time 2h (R/W) = First rising edge of GPMC_CLK two GPMC_FCLK cycles after start access time 3h (R/W) = Reserved

Table 9-62. GPMC_CONFIG1_0 to GPMC_CONFIG1_6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24-23	ATTACHEDDEVICEPAGELENGTH	R/W	0h	Specifies the attached device page (burst) length (1 Word = Interface size) 0h (R/W) = 4 Words 1h (R/W) = 8 Words 2h (R/W) = 16 Words 3h (R/W) = Reserved
22	WAITREADMONITORING	R/W	0h	Selects the Wait monitoring configuration for Read accesses. 0h (R/W) = WAIT pin is not monitored for read accesses 1h (R/W) = WAIT pin is monitored for read accesses
21	WAITWRITEMONITORING	R/W	0h	Selects the Wait monitoring configuration for Write accesses 0h (R/W) = WAIT pin is not monitored for write accesses 1h (R/W) = WAIT pin is monitored for write accesses
20	RESERVED	R	0h	
19-18	WAITMONITORINGTIME	R/W	0h	Selects input pin Wait monitoring time 0h (R/W) = WAIT pin is monitored with valid data 1h (R/W) = WAIT pin is monitored one GPMC_CLK cycle before valid data 2h (R/W) = WAIT pin is monitored two GPMC_CLK cycle before valid data 3h (R/W) = Reserved
17-16	WAITPINSELECT	R/W	0h	Selects the input WAIT pin for this chip select. 0h (R/W) = WAIT input pin is WAIT0 1h (R/W) = WAIT input pin is WAIT1 2h (R/W) = Reserved 3h (R/W) = Reserved
15-14	RESERVED	R	0h	
13-12	DEVICESIZE	R/W	0h	Selects the device size attached (Reset value is SYSBOOT[8] input pin sampled at IC reset for CS[0] (active low) and 01 for CS[1] to CS[6] (active low)). 0h (R/W) = 8 bit 1h (R/W) = 16 bit 2h (R/W) = Reserved 3h (R/W) = Reserved
11-10	DEVICETYPE	R/W	0h	Selects the attached device type 0h (R/W) = NOR Flash like, asynchronous and synchronous devices 1h (R/W) = Reserved 2h (R/W) = NAND Flash like devices, stream mode 3h (R/W) = Reserved
9-8	MUXADDDATA	R/W	0h	Enables the Address and data multiplexed protocol (Reset value is SYSBOOT[11] and SYSBOOT[10] input pins sampled at IC reset for CS[0] (active low) and 0 for CS[1] to CS[6] (active low)). 0h (R/W) = Non-multiplexed attached device 1h (R/W) = AAD-multiplexed protocol device 2h (R/W) = Address and data multiplexed attached device 3h (R/W) = Reserved
7-5	RESERVED	R	0h	
4	TIMEPARAGRANULARITY	R/W	0h	Signals timing latencies scalar factor (Rd/WRCycleTime, AccessTime, PageBurstAccessTime, CSOnTime, CSRd/WrOffTime, ADVOnTime, ADVRd/WrOffTime, OEOnTime, OEOffTime, WEOnTime, WEOffTime, Cycle2CycleDelay, BusTurnAround, TimeOutStartValue) 0h (R/W) = x1 latencies 1h (R/W) = x2 latencies
3-2	RESERVED	R	0h	

Table 9-62. GPMC_CONFIG1_0 to GPMC_CONFIG1_6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GPMCFCLKDIVIDER	R/W	0h	Divides the GPMC.FCLK clock 0h (R/W) = GPMC_CLK frequency = GPMC_FCLK frequency 1h (R/W) = GPMC_CLK frequency = GPMC_FCLK frequency/2 2h (R/W) = GPMC_CLK frequency = GPMC_FCLK frequency/3 3h (R/W) = GPMC_CLK frequency = GPMC_FCLK frequency/4

9.1.6.12 GPMC_CONFIG2_0 to GPMC_CONFIG2_6 Register (Offset = 64h + [i * 0h], where i = 0 to 6) [reset = 0h]

GPMC_CONFIG2_0 to GPMC_CONFIG2_6 is shown in Figure 9-59 and described in Table 9-63.

Return to Summary Table.

Chip-select signal timing parameter configuration.

Figure 9-59. GPMC_CONFIG2_0 to GPMC_CONFIG2_6 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="3">RESERVED</td><td colspan="5">CSWROFFTIME</td></tr>
<tr><td colspan="3">R-0h</td><td colspan="5">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="3">RESERVED</td><td colspan="5">CSRDOFFTIME</td></tr>
<tr><td colspan="3">R-0h</td><td colspan="5">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>CSEXTRADELAY</td><td colspan="3">RESERVED</td><td colspan="4">CSONTIME</td></tr>
<tr><td>R/W-0h</td><td colspan="3">R-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 9-63. GPMC_CONFIG2_0 to GPMC_CONFIG2_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R	0h	
20-16	CSWROFFTIME	R/W	0h	CS# de-assertion time from start cycle time for write accesses 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle 1Fh (R/W) = 31 GPMC_FCLK cycles
15-13	RESERVED	R	0h	
12-8	CSRDOFFTIME	R/W	0h	CS# de-assertion time from start cycle time for read accesses 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle 1Fh (R/W) = 31 GPMC_FCLK cycles
7	CSEXTRADELAY	R/W	0h	CS# Add Extra Half GPMC.FCLK cycle 0h (R/W) = CS i Timing control signal is not delayed 1h (R/W) = CS i Timing control signal is delayed of half GPMC_FCLK clock cycle
6-4	RESERVED	R	0h	
3-0	CSONTIME	R/W	0h	CS# assertion time from start cycle time 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle 1Fh (R/W) = 15 GPMC_FCLK cycles

9.1.6.13 GPMC_CONFIG3_0 to GPMC_CONFIG3_6 Register (Offset = 68h + [i * 0h], where i = 0 to 6) [reset = 0h]

GPMC_CONFIG3_0 to GPMC_CONFIG3_6 is shown in Figure 9-60 and described in Table 9-64.

Return to Summary Table.

ADV# signal timing parameter configuration.

Figure 9-60. GPMC_CONFIG3_0 to GPMC_CONFIG3_6 Register

31	30	29	28	27	26	25	24
RESERVED	ADVAADMUXWROFFTIME			RESERVED	ADVAADMUXRDOFFTIME		
R-0h	R/W-0h			R-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED			ADVWROFFTIME				
R-0h			R/W-0h				
15	14	13	12	11	10	9	8
RESERVED			ADVRDOFFTIME				
R-0h			R/W-0h				
7	6	5	4	3	2	1	0
ADVEXTRADELAY	ADVAADMUXONTIME			ADVONTIME			
R/W-0h	R/W-0h			R/W-0h			

Table 9-64. GPMC_CONFIG3_0 to GPMC_CONFIG3_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	ADVAADMUXWROFFTIME	R/W	0h	ADV# de-assertion for first address phase when using the AAD-Mux protocol 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle 7h (R/W) = 7 GPMC_FCLK cycles
27	RESERVED	R	0h	
26-24	ADVAADMUXRDOFFTIME	R/W	0h	ADV# assertion for first address phase when using the AAD-Mux protocol 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle 7h (R/W) = 7 GPMC_FCLK cycles
23-21	RESERVED	R	0h	
20-16	ADVWROFFTIME	R/W	0h	ADV# de-assertion time from start cycle time for write accesses 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle 1Fh (R/W) = 31 GPMC_FCLK cycles
15-13	RESERVED	R	0h	
12-8	ADVRDOFFTIME	R/W	0h	ADV# de-assertion time from start cycle time for read accesses 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle 1Fh (R/W) = 31 GPMC_FCLK cycles
7	ADVEXTRADELAY	R/W	0h	ADV# Add Extra Half GPMC.FCLK cycle 0h (R/W) = ADV (active low) Timing control signal is not delayed 1h (R/W) = ADV (active low) Timing control signal is delayed of half GPMC_FCLK clock cycle

Table 9-64. GPMC_CONFIG3_0 to GPMC_CONFIG3_6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6-4	ADVAADMUXONTIME	R/W	0h	ADV# assertion for first address phase when using the AAD-Multiplexed protocol 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle 7h (R/W) = 7 GPMC_FCLK cycles
3-0	ADVONTIME	R/W	0h	ADV# assertion time from start cycle time 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle Fh (R/W) = 15 GPMC_FCLK cycles

9.1.6.14 GPMC_CONFIG4_0 to GPMC_CONFIG4_6 Register (Offset = 6Ch + [i * 0h], where i = 0 to 6) [reset = 0h]

GPMC_CONFIG4_0 to GPMC_CONFIG4_6 is shown in Figure 9-61 and described in Table 9-65.

Return to Summary Table.

WE# and OE# signals timing parameter configuration.

Figure 9-61. GPMC_CONFIG4_0 to GPMC_CONFIG4_6 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="3">RESERVED</td><td colspan="5">WEOFFTIME</td></tr>
<tr><td colspan="3">R-0h</td><td colspan="5">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>WEEXTRADELAY</td><td colspan="3">RESERVED</td><td colspan="4">WEONTIME</td></tr>
<tr><td>R/W-0h</td><td colspan="3">R-0h</td><td colspan="4">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="3">OEAADMUXOFFTIME</td><td colspan="5">OEOFFTIME</td></tr>
<tr><td colspan="3">R/W-0h</td><td colspan="5">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>OEEXTRADELAY</td><td colspan="3">OEAADMUXONTIME</td><td colspan="4">OEONTIME</td></tr>
<tr><td>R/W-0h</td><td colspan="3">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 9-65. GPMC_CONFIG4_0 to GPMC_CONFIG4_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	WEOFFTIME	R/W	0h	WE# de-assertion time from start cycle time 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle 1Fh (R/W) = 31 GPMC_FCLK cycles
23	WEEXTRADELAY	R/W	0h	WE# Add Extra Half GPMC.FCLK cycle 0h (R/W) = WE (active low) Timing control signal is not delayed 1h (R/W) = WE (active low) Timing control signal is delayed of half GPMC_FCLK clock cycle
22-20	RESERVED	R	0h	
19-16	WEONTIME	R/W	0h	WE# assertion time from start cycle time 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle Fh (R/W) = 15 GPMC_FCLK cycles
15-13	OEAADMUXOFFTIME	R/W	0h	OE# de-assertion time for the first address phase in an AAD-Multiplexed access 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle 7h (R/W) = 7 GPMC_FCLK cycles
12-8	OEOFFTIME	R/W	0h	OE# de-assertion time from start cycle time 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle 1Fh (R/W) = 31 GPMC_FCLK cycles
7	OEEXTRADELAY	R/W	0h	OE# Add Extra Half GPMC.FCLK cycle 0h (R/W) = OE (active low) Timing control signal is not delayed 1h (R/W) = OE (active low) Timing control signal is delayed of half GPMC_FCLK clock cycle

Table 9-65. GPMC_CONFIG4_0 to GPMC_CONFIG4_6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6-4	OEAADMUXONTIME	R/W	0h	OE# assertion time for the first address phase in an AAD-Multiplexed access 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle 7h (R/W) = 7 GPMC_FCLK cycles
3-0	OEONTIME	R/W	0h	OE# assertion time from start cycle time 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle Fh (R/W) = 15 GPMC_FCLK cycles

9.1.6.15 GPMC_CONFIG5_0 to GPMC_CONFIG5_6 Register (Offset = 70h + [i * 0h], where i = 0 to 6) [reset = 0h]

GPMC_CONFIG5_0 to GPMC_CONFIG5_6 is shown in Figure 9-62 and described in Table 9-66.

Return to Summary Table.

RdAccessTime and CycleTime timing parameters configuration.

Figure 9-62. GPMC_CONFIG5_0 to GPMC_CONFIG5_6 Register

31	30	29	28	27	26	25	24
RESERVED				PAGEBURSTACCESSTIME			
R-0h				R/W-0h			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED			RDACCESSTIME				
R-0h			R/W-0h				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED			WRCYCLETIME				
R-0h			R/W-0h				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED			RDCYCLETIME				
R-0h			R/W-0h				

Table 9-66. GPMC_CONFIG5_0 to GPMC_CONFIG5_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-24	PAGEBURSTACCESSTIME	R/W	0h	Delay between successive words in a multiple access 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle Fh (R/W) = 15 GPMC_FCLK cycles
23-21	RESERVED	R	0h	
20-16	RDACCESSTIME	R/W	0h	Delay between start cycle time and first data valid 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle 1Fh (R/W) = 31 GPMC_FCLK cycles
15-13	RESERVED	R	0h	
12-8	WRCYCLETIME	R/W	0h	Total write cycle time 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle 1Fh (R/W) = 31 GPMC_FCLK cycles
7-5	RESERVED	R	0h	
4-0	RDCYCLETIME	R/W	0h	Total read cycle time 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle 1Fh (R/W) = 31 GPMC_FCLK cycles

9.1.6.16 GPMC_CONFIG6_0 to GPMC_CONFIG6_6 Register (Offset = 74h + [i * 0h], where i = 0 to 6) [reset = 0F070000h]

GPMC_CONFIG6_0 to GPMC_CONFIG6_6 is shown in Figure 9-63 and described in Table 9-67.

Return to Summary Table.

WrAccessTime, WrDataOnADmuxBus, Cycle2Cycle, and BusTurnAround parameters configuration

Figure 9-63. GPMC_CONFIG6_0 to GPMC_CONFIG6_6 Register

31	30	29	28	27	26	25	24
RESERVED			WRACCESSTIME				
R-0h			R/W-Fh				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				WRDATAONADMUXBUS			
R-0h				R/W-7h			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED				CYCLE2CYCLEDELAY			
R-0h				R/W-0h			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CYCLE2CYCLESAMECSEN	CYCLE2CYCLEDIFFCSEN	RESERVED		BUSTURNAROUND			
R/W-0h	R/W-0h	R-0h		R/W-0h			

Table 9-67. GPMC_CONFIG6_0 to GPMC_CONFIG6_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	Reset value for bit 31 is 1.
28-24	WRACCESSTIME	R/W	Fh	Delay from StartAccessTime to the GPMC.FCLK rising edge corresponding the the GPMC.CLK rising edge used by the attached memory for the first data capture. Reset value is 0xF. 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle 1Fh (R/W) = 31 GPMC_FCLK cycles
23-20	RESERVED	R	0h	
19-16	WRDATAONADMUXBUS	R/W	7h	Specifies on which GPMC.FCLK rising edge the first data of the synchronous burst write is driven in the add/data multiplexed bus. Reset value is 0x7.
15-12	RESERVED	R	0h	
11-8	CYCLE2CYCLEDELAY	R/W	0h	Chip select high pulse delay between two successive accesses 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle Fh (R/W) = 15 GPMC_FCLK cycles
7	CYCLE2CYCLESAMECSEN	R/W	0h	Add Cycle2CycleDelay between two successive accesses to the same chip-select (any access type) 0h (R/W) = No delay between the two accesses 1h (R/W) = Add CYCLE2CYCLEDELAY
6	CYCLE2CYCLEDIFFCSEN	R/W	0h	Add Cycle2CycleDelay between two successive accesses to a different chip-select (any access type) 0h (R/W) = No delay between the two accesses 1h (R/W) = Add CYCLE2CYCLEDELAY
5-4	RESERVED	R	0h	

Table 9-67. GPMC_CONFIG6_0 to GPMC_CONFIG6_6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	BUSTURNAROUND	R/W	0h	Bus turn around latency between two successive accesses to the same chip-select (read to write) or to a different chip-select (read to read and read to write) 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle Fh (R/W) = 15 GPMC_FCLK cycles

9.1.6.17 GPMC_CONFIG7_0 to GPMC_CONFIG7_6 Register (Offset = 78h + [i * 0h], where i = 0 to 6) [reset = 0h]

GPMC_CONFIG7_0 to GPMC_CONFIG7_6 is shown in Figure 9-64 and described in Table 9-68.

Return to Summary Table.

Chip-select address mapping configuration.

Figure 9-64. GPMC_CONFIG7_0 to GPMC_CONFIG7_6 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">MASKADDRESS</td></tr>
<tr><td colspan="4">R-0h</td><td colspan="4">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>RESERVED</td><td>CSVALID</td><td colspan="6">BASEADDRESS</td></tr>
<tr><td>R-0h</td><td>R/W-0h</td><td colspan="6">R/W-0h</td></tr>
</table>

Table 9-68. GPMC_CONFIG7_0 to GPMC_CONFIG7_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	
11-8	MASKADDRESS	R/W	0h	Chip-select mask address. Values not listed must be avoided as they create holes in the chip-select address space. 0h (R/W) = Chip-select size of 256 Mbytes 8h (R/W) = Chip-select size of 128 Mbytes Ch (R/W) = Chip-select size of 64 Mbytes Eh (R/W) = Chip-select size of 32 Mbytes Fh (R/W) = Chip-select size of 16 Mbytes
7	RESERVED	R	0h	
6	CSVALID	R/W	0h	Chip-select enable (reset value is 1 for CS[0] (active low) and 0 for CS[1] to CS[5] (active low)). 0h (R/W) = CS (active low) disabled 1h (R/W) = CS (active low) enabled
5-0	BASEADDRESS	R/W	0h	Chip-select base address. CSi base address where i = 0 to 3 (16 Mbytes minimum granularity). Bits 5 to 0 correspond to A29, A28, A27, A26, A25, and A24.

9.1.6.18 GPMC_NAND_COMMAND_0 to GPMC_NAND_COMMAND_6 Register (Offset = 7Ch + [i * 0h], where i = 0 to 6) [reset = 0h]

GPMC_NAND_COMMAND_0 to GPMC_NAND_COMMAND_6 is shown in Figure 9-65 and described in Table 9-69.

Return to Summary Table.

This register is not a true register, just an address location.

Figure 9-65. GPMC_NAND_COMMAND_0 to GPMC_NAND_COMMAND_6 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">GPMC_NAND_COMMAND</td></tr>
<tr><td colspan="32">W-0h</td></tr>
</table>

Table 9-69. GPMC_NAND_COMMAND_0 to GPMC_NAND_COMMAND_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPMC_NAND_COMMAND	W	0h	Writing data at the GPMC_NAND_COMMAND_n location places the data as the NAND command value on the bus, using a regular asynchronous write access.

9.1.6.19 GPMC_NAND_ADDRESS_0 to GPMC_NAND_ADDRESS_6 Register (Offset = 80h + [i * 0h], where i = 0 to 6) [reset = 0h]

GPMC_NAND_ADDRESS_0 to GPMC_NAND_ADDRESS_6 is shown in Figure 9-66 and described in Table 9-70.

Return to Summary Table.

This register is not a true register, just an address location.

Figure 9-66. GPMC_NAND_ADDRESS_0 to GPMC_NAND_ADDRESS_6 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">GPMC_NAND_ADDRESS</td></tr>
<tr><td colspan="32">W-0h</td></tr>
</table>

Table 9-70. GPMC_NAND_ADDRESS_0 to GPMC_NAND_ADDRESS_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPMC_NAND_ADDRESS	W	0h	Writing data at the GPMC_NAND_ADDRESS_n location places the data as the NAND partial address value on the bus, using a regular asynchronous write access.

9.1.6.20 GPMC_NAND_DATA_0 to GPMC_NAND_DATA_6 Register (Offset = 84h + [i * 0h], where i = 0 to 6) [reset = 0h]

GPMC_NAND_DATA_0 to GPMC_NAND_DATA_6 is shown in Figure 9-67 and described in Table 9-71.

Return to Summary Table.

This register is not a true register, just an address location.

Figure 9-67. GPMC_NAND_DATA_0 to GPMC_NAND_DATA_6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_NAND_DATA																															
R/W-0h																															

Table 9-71. GPMC_NAND_DATA_0 to GPMC_NAND_DATA_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPMC_NAND_DATA	R/W	0h	Reading data from the GPMC_NAND_DATA_n location or from any location in the associated chip-select memory region activates an asynchronous read access.

9.1.6.21 GPMC_PREFETCH_CONFIG1 Register (Offset = 1E0h) [reset = 0h]

GPMC_PREFETCH_CONFIG1 is shown in Figure 9-68 and described in Table 9-72.

Return to Summary Table.

Figure 9-68. GPMC_PREFETCH_CONFIG1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>RESERVED</td><td colspan="3">CYCLEOPTIMIZATION</td><td>ENABLEOPTIMIZEDACCESS</td><td colspan="3">ENGINECSSELECTOR</td></tr>
<tr><td>R-0h</td><td colspan="3">R/W-0h</td><td>R/W-0h</td><td colspan="3">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>PFPWENROUNDROBIN</td><td colspan="3">RESERVED</td><td colspan="4">PFPWWEIGHTEDPRIO</td></tr>
<tr><td>R/W-0h</td><td colspan="3">R-0h</td><td colspan="4">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>RESERVED</td><td colspan="7">FIFOTHRESHOLD</td></tr>
<tr><td>R-0h</td><td colspan="7">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>ENABLEENGINE</td><td>RESERVED</td><td colspan="2">WAITPINSELECTOR</td><td>SYNCHROMODE</td><td>DMAMODE</td><td>RESERVED</td><td>ACCESSMODE</td></tr>
<tr><td>R/W-0h</td><td>R-0h</td><td colspan="2">R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R-0h</td><td>R/W-0h</td></tr>
</table>

Table 9-72. GPMC_PREFETCH_CONFIG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	CYCLEOPTIMIZATION	R/W	0h	Define the number of GPMC.FCLK cycles to be substracted from RdCycleTime, WrCycleTime, AccessTime, CSRdOffTime, CSWrOffTime, ADVRdOffTime, ADVWrOffTime, OEOffTime, WEOffTime 0h (R/W) = 0 GPMC_FCLK cycle 1h (R/W) = 1 GPMC_FCLK cycle 7h (R/W) = 7 GPMC_FCLK cycles
27	ENABLEOPTIMIZEDACCESS	R/W	0h	Enables access cycle optimization 0h (R/W) = Access cycle optimization is disabled 1h (R/W) = Access cycle optimization is enabled
26-24	ENGINECSSELECTOR	R/W	0h	Selects the CS (active low) where Prefetch Postwrite engine is active 0h (R/W) = CS0 (active low) 1h (R/W) = CS1 (active low) 2h (R/W) = CS2 (active low) 3h (R/W) = CS3 (active low) 4h (R/W) = CS4 (active low) 5h (R/W) = CS5 (active low) 6h (R/W) = CS6 (active low)
23	PFPWENROUNDROBIN	R/W	0h	Enables the PFPW RoundRobin arbitration 0h (R/W) = Prefetch Postwrite engine round robin arbitration is disabled 1h (R/W) = Prefetch Postwrite engine round robin arbitration is enabled
22-20	RESERVED	R	0h	
19-16	PFPWWEIGHTEDPRIO	R/W	0h	When an arbitration occurs between a direct memory access and a PFPW engine access, the direct memory access is always serviced. If the PFPWEnRoundRobin is enabled: 0h (R/W) = The next access is granted to the PFPW engine 1h (R/W) = The two next accesses are granted to the PFPW engine Fh (R/W) = The 16 next accesses are granted to the PFPW engine

Table 9-72. GPMC_PREFETCH_CONFIG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	
14-8	FIFOTHRESHOLD	R/W	0h	Selects the maximum number of bytes read from the FIFO or written to the FIFO by the host on a DMA or interrupt request 0h (R/W) = 0 byte 1h (R/W) = 1 byte 40h (R/W) = 64 bytes
7	ENABLEENGINE	R/W	0h	Enables the Prefetch Postwite engine 0h (R/W) = Prefetch Postwrite engine is disabled 1h (R/W) = Prefetch Postwrite engine is enabled
6	RESERVED	R	0h	
5-4	WAITPINSELECTOR	R/W	0h	Select which wait pin edge detector should start the engine in synchronized mode 0h (R/W) = Selects Wait0EdgeDetection 1h (R/W) = Selects Wait1EdgeDetection 2h (R/W) = Reserved 3h (R/W) = Reserved
3	SYNCHROMODE	R/W	0h	Selects when the engine starts the access to CS 0h (R/W) = Engine starts the access to CS as soon as STARTENGINE is set 1h (R/W) = Engine starts the access to CS as soon as STARTENGINE is set AND wait to non wait edge detection on the selected wait pin
2	DMAMODE	R/W	0h	Selects interrupt synchronization or DMA request synchronization 0h (R/W) = Interrupt synchronization is enabled. Only interrupt line will be activated on FIFO threshold crossing. 1h (R/W) = DMA request synchronization is enabled. A DMA request protocol is used.
1	RESERVED	R	0h	
0	ACCESSMODE	R/W	0h	Selects pre-fetch read or write posting accesses 0h (R/W) = Prefetch read mode 1h (R/W) = Write-posting mode

9.1.6.22 GPMC_PREFETCH_CONFIG2 Register (Offset = 1E4h) [reset = 0h]

GPMC_PREFETCH_CONFIG2 is shown in Figure 9-69 and described in Table 9-73.

Return to Summary Table.

Figure 9-69. GPMC_PREFETCH_CONFIG2 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="18">RESERVED</td><td colspan="14">TRANSFERCOUNT</td></tr>
<tr><td colspan="18">R-0h</td><td colspan="14">R/W-0h</td></tr>
</table>

Table 9-73. GPMC_PREFETCH_CONFIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	
13-0	TRANSFERCOUNT	R/W	0h	Selects the number of bytes to be read or written by the engine to the selected CS (active low) 0h (R/W) = 0 byte 1h (R/W) = 1 byte 2000h (R/W) = 8 Kbytes

9.1.6.23 GPMC_PREFETCH_CONTROL Register (Offset = 1ECh) [reset = 0h]

GPMC_PREFETCH_CONTROL is shown in Figure 9-70 and described in Table 9-74.

Return to Summary Table.

Figure 9-70. GPMC_PREFETCH_CONTROL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							STARTENGINE
R-0h							R/W-0h

Table 9-74. GPMC_PREFETCH_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	STARTENGINE	R/W	0h	Resets the FIFO pointer and starts the engine 0h (W) = Stops the engine 0h (R) = Engine is stopped 1h (W) = Resets the FIFO pointer to 0 in prefetch mode and 40h in postwrite mode and starts the engine 1h (R) = Engine is running

9.1.6.24 GPMC_PREFETCH_STATUS Register (Offset = 1F0h) [reset = 0h]

GPMC_PREFETCH_STATUS is shown in Figure 9-71 and described in Table 9-75.

Return to Summary Table.

Figure 9-71. GPMC_PREFETCH_STATUS Register

31	30	29	28	27	26	25	24
RESERVED	FIFOPOINTER						
R-0h	R-0h						

23	22	21	20	19	18	17	16
RESERVED							FIFOTHRESHOLDSTATUS
R-0h							R-0h

15	14	13	12	11	10	9	8
RESERVED		COUNTVALUE					
R-0h		R-0h					

7	6	5	4	3	2	1	0
COUNTVALUE							
R-0h							

Table 9-75. GPMC_PREFETCH_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-24	FIFOPOINTER	R	0h	FIFOPOINTER. 0h (R/W) = 0 byte available to be read or 0 free empty place to be written 40h (R/W) = 64 bytes available to be read or 64 empty places to be written
23-17	RESERVED	R	0h	
16	FIFOTHRESHOLDSTATUS	R	0h	Set when FIFOPointer exceeds FIFOThreshold value. 0h (R/W) = FIFOPointer smaller or equal to FIFOThreshold. Writing to this bit has no effect 1h (R/W) = FIFOPointer greater than FIFOThreshold. Writing to this bit has no effect
15-14	RESERVED	R	0h	
13-0	COUNTVALUE	R	0h	Number of remaining bytes to be read or to be written by the engine according to the TransferCount value. 0h (R/W) = 0 byte remaining to be read or to be written 1h (R/W) = 1 byte remaining to be read or to be write 2000h (R/W) = 8 Kbytes remaining to be read or to be written

9.1.6.25 GPMC_ECC_CONFIG Register (Offset = 1F4h) [reset = 0h]

GPMC_ECC_CONFIG is shown in Figure 9-72 and described in Table 9-76.

Return to Summary Table.

Figure 9-72. GPMC_ECC_CONFIG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="7">RESERVED</td><td>ECCALGORITHM</td></tr>
<tr><td colspan="7">R-0h</td><td>R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="2">ECCBCHTSEL</td><td colspan="4">ECCWRAPMODE</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="2">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>ECC16B</td><td colspan="3">ECCTOPSECTOR</td><td colspan="3">ECCCS</td><td>ECCENABLE</td></tr>
<tr><td>R/W-0h</td><td colspan="3">R/W-0h</td><td colspan="3">R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 9-76. GPMC_ECC_CONFIG Register Field Descriptions

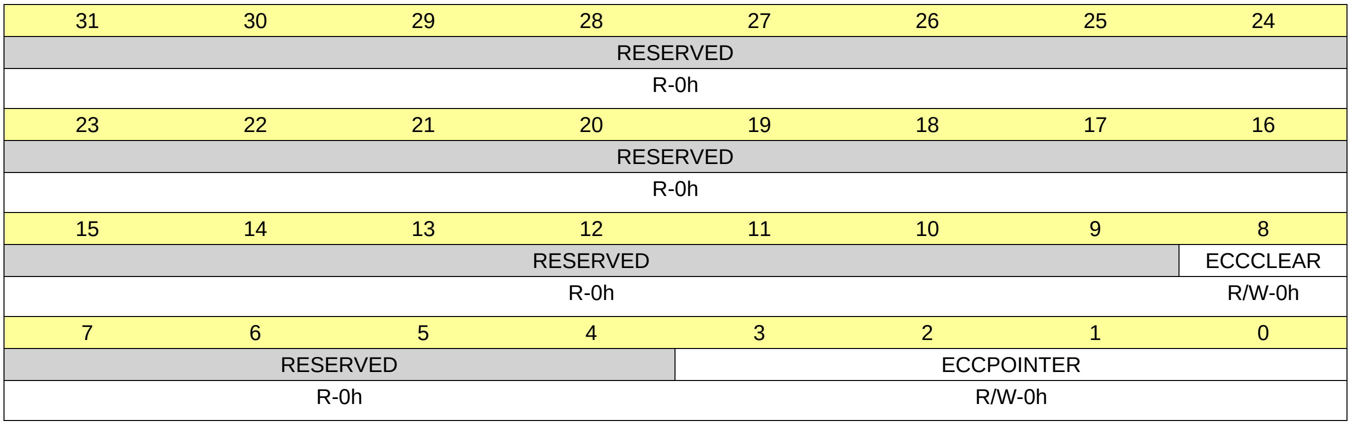
Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	ECCALGORITHM	R/W	0h	ECC algorithm used 0h (R/W) = Hamming code 1h (R/W) = BCH code
15-14	RESERVED	R	0h	
13-12	ECCBCHTSEL	R/W	0h	Error correction capability used for BCH 0h (R/W) = Up to 4 bits error correction (t = 4) 1h (R/W) = Up to 8 bits error correction (t = 8) 2h (R/W) = Up to 16 bits error correction (t = 16) 3h (R/W) = Reserved
11-8	ECCWRAPMODE	R/W	0h	Spare area organization definition for the BCH algorithm.
7	ECC16B	R/W	0h	Selects an ECC calculated on 16 columns 0h (R/W) = ECC calculated on 8 columns 1h (R/W) = ECC calculated on 16 columns
6-4	ECCTOPSECTOR	R/W	0h	Number of sectors to process with the BCH algorithm 0h (R/W) = 1 sector (512kB page) 1h (R/W) = 2 sectors 3h (R/W) = 4 sectors (2kB page) 7h (R/W) = 8 sectors (4kB page)
3-1	ECCCS	R/W	0h	Selects the Chip-select where ECC is computed 0h (R/W) = Chip-select 0 1h (R/W) = Chip-select 1 2h (R/W) = Chip-select 2 3h (R/W) = Chip-select 3 4h (R/W) = Chip-select 4 5h (R/W) = Chip-select 5 6h (R/W) = Reserved 7h (R/W) = Reserved
0	ECCENABLE	R/W	0h	Enables the ECC feature 0h (R/W) = ECC disabled 1h (R/W) = ECC enabled

9.1.6.26 GPMC_ECC_CONTROL Register (Offset = 1F8h) [reset = 0h]

GPMC_ECC_CONTROL is shown in Figure 9-73 and described in Table 9-77.

Return to Summary Table.

Figure 9-73. GPMC_ECC_CONTROL Register



31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							ECCCLEAR
R-0h							R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				ECCPOINTER			
R-0h				R/W-0h			

Table 9-77. GPMC_ECC_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	ECCCLEAR	R/W	0h	Clear all ECC result registers 0h (W) = Ignored 0h (R) = All reads return 0 1h (W) = Clears all ECC result registers
7-4	RESERVED	R	0h	
3-0	ECCPOINTER	R/W	0h	Selects ECC result register (Reads to this field give the dynamic position of the ECC pointer - Writes to this field select the ECC result register where the first ECC computation will be stored). Writing values not listed disables the ECC engine (ECCEnable bit of GPMC_ECC_CONFIG cleared to 0). 0h (R/W) = Writing 0 disables the ECC engine (ECCENABLE bit of GPMC_ECC_CONFIG cleared to 0) 1h (R/W) = ECC result register 1 selected 2h (R/W) = ECC result register 2 selected 3h (R/W) = ECC result register 3 selected 4h (R/W) = ECC result register 4 selected 5h (R/W) = ECC result register 5 selected 6h (R/W) = ECC result register 6 selected 7h (R/W) = ECC result register 7 selected 8h (R/W) = ECC result register 8 selected 9h (R/W) = ECC result register 9 selected

9.1.6.27 GPMC_ECC_SIZE_CONFIG Register (Offset = 1FCh) [reset = 0h]

GPMC_ECC_SIZE_CONFIG is shown in Figure 9-74 and described in Table 9-78.

Return to Summary Table.

Figure 9-74. GPMC_ECC_SIZE_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED		ECCSIZE1					
R-0h		R/W-0h					
23	22	21	20	19	18	17	16
ECCSIZE1		RESERVED		ECCSIZE0			
R/W-0h		R-0h		R/W-0h			
15	14	13	12	11	10	9	8
ECCSIZE0				RESERVED			ECC9RESULT SIZE
R/W-0h				R-0h			R/W-0h
7	6	5	4	3	2	1	0
ECC8RESULT SIZE	ECC7RESULT SIZE	ECC6RESULT SIZE	ECC5RESULT SIZE	ECC4RESULT SIZE	ECC3RESULT SIZE	ECC2RESULT SIZE	ECC1RESULT SIZE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-78. GPMC_ECC_SIZE_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	
29-22	ECCSIZE1	R/W	0h	Defines ECC size 1 0h (R/W) = 2 Bytes 1h (R/W) = 4 Bytes 2h (R/W) = 6 Bytes 3h (R/W) = 8 Bytes FFh (R/W) = 512 Bytes
21-20	RESERVED	R	0h	
19-12	ECCSIZE0	R/W	0h	Defines ECC size 0 0h (R/W) = 2 Bytes 1h (R/W) = 4 Bytes 2h (R/W) = 6 Bytes 3h (R/W) = 8 Bytes FFh (R/W) = 512 Bytes
11-9	RESERVED	R	0h	
8	ECC9RESULTSIZE	R/W	0h	Selects ECC size for ECC 9 result register 0h (R/W) = ECCSIZE0 selected 1h (R/W) = ECCSIZE1 selected
7	ECC8RESULTSIZE	R/W	0h	Selects ECC size for ECC 8 result register 0h (R/W) = ECCSIZE0 selected 1h (R/W) = ECCSIZE1 selected
6	ECC7RESULTSIZE	R/W	0h	Selects ECC size for ECC 7 result register 0h (R/W) = ECCSIZE0 selected 1h (R/W) = ECCSIZE1 selected
5	ECC6RESULTSIZE	R/W	0h	Selects ECC size for ECC 6 result register 0h (R/W) = ECCSIZE0 selected 1h (R/W) = ECCSIZE1 selected
4	ECC5RESULTSIZE	R/W	0h	Selects ECC size for ECC 5 result register 0h (R/W) = ECCSIZE0 selected 1h (R/W) = ECCSIZE1 selected

Table 9-78. GPMC_ECC_SIZE_CONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	ECC4RESULTSIZE	R/W	0h	Selects ECC size for ECC 4 result register 0h (R/W) = ECCSIZE0 selected 1h (R/W) = ECCSIZE1 selected
2	ECC3RESULTSIZE	R/W	0h	Selects ECC size for ECC 3 result register 0h (R/W) = ECCSIZE0 selected 1h (R/W) = ECCSIZE1 selected
1	ECC2RESULTSIZE	R/W	0h	Selects ECC size for ECC 2 result register 0h (R/W) = ECCSIZE0 selected 1h (R/W) = ECCSIZE1 selected
0	ECC1RESULTSIZE	R/W	0h	Selects ECC size for ECC 1 result register 0h (R/W) = ECCSIZE0 selected 1h (R/W) = ECCSIZE1 selected

9.1.6.28 GPMC_ECC1_RESULT Register (Offset = 200h) [reset = 0h]

GPMC_ECC1_RESULT is shown in Figure 9-75 and described in Table 9-79.

Return to Summary Table.

Figure 9-75. GPMC_ECC1_RESULT Register

31	30	29	28	27	26	25	24
RESERVED				P2048O	P1024O	P512O	P256O
R-0h				R-0h	R-0h	R-0h	R-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
P128O	P64O	P32O	P16O	P8O	P4O	P2O	P1O
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED				P2048E	P1024E	P512E	P256E
R-0h				R-0h	R-0h	R-0h	R-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
P128E	P64E	P32E	P16E	P8E	P4E	P2E	P1E
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 9-79. GPMC_ECC1_RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27	P2048O	R	0h	Odd Row Parity bit 2048, only used for ECC computed on 512 Bytes
26	P1024O	R	0h	Odd Row Parity bit 1024
25	P512O	R	0h	Odd Row Parity bit 512
24	P256O	R	0h	Odd Row Parity bit 256
23	P128O	R	0h	Odd Row Parity bit 128
22	P64O	R	0h	Odd Row Parity bit 64
21	P32O	R	0h	Odd Row Parity bit 32
20	P16O	R	0h	Odd Row Parity bit 16
19	P8O	R	0h	Odd Row Parity bit 8
18	P4O	R	0h	Odd Column Parity bit 4
17	P2O	R	0h	Odd Column Parity bit 2
16	P1O	R	0h	Odd Column Parity bit 1
15-12	RESERVED	R	0h	
11	P2048E	R	0h	Even Row Parity bit 2048, only used for ECC computed on 512 Bytes
10	P1024E	R	0h	Even Row Parity bit 1024
9	P512E	R	0h	Even Row Parity bit 512
8	P256E	R	0h	Even Row Parity bit 256
7	P128E	R	0h	Even Row Parity bit 128
6	P64E	R	0h	Even Row Parity bit 64
5	P32E	R	0h	Even Row Parity bit 32
4	P16E	R	0h	Even Row Parity bit 16
3	P8E	R	0h	Even Row Parity bit 8
2	P4E	R	0h	Even Column Parity bit 4
1	P2E	R	0h	Even Column Parity bit 2

Table 9-79. GPMC_ECC1_RESULT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	P1E	R	0h	Even Column Parity bit 1

9.1.6.29 GPMC_ECC2_RESULT Register (Offset = 204h) [reset = 0h]

GPMC_ECC2_RESULT is shown in Figure 9-76 and described in Table 9-80.

Return to Summary Table.

Figure 9-76. GPMC_ECC2_RESULT Register

31	30	29	28	27	26	25	24
RESERVED				P2048O	P1024O	P512O	P256O
R-0h				R-0h	R-0h	R-0h	R-0h

23	22	21	20	19	18	17	16
P128O	P64O	P32O	P16O	P8O	P4O	P2O	P1O
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

15	14	13	12	11	10	9	8
RESERVED				P2048E	P1024E	P512E	P256E
R-0h				R-0h	R-0h	R-0h	R-0h

7	6	5	4	3	2	1	0
P128E	P64E	P32E	P16E	P8E	P4E	P2E	P1E
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 9-80. GPMC_ECC2_RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27	P2048O	R	0h	Odd Row Parity bit 2048, only used for ECC computed on 512 Bytes
26	P1024O	R	0h	Odd Row Parity bit 1024
25	P512O	R	0h	Odd Row Parity bit 512
24	P256O	R	0h	Odd Row Parity bit 256
23	P128O	R	0h	Odd Row Parity bit 128
22	P64O	R	0h	Odd Row Parity bit 64
21	P32O	R	0h	Odd Row Parity bit 32
20	P16O	R	0h	Odd Row Parity bit 16
19	P8O	R	0h	Odd Row Parity bit 8
18	P4O	R	0h	Odd Column Parity bit 4
17	P2O	R	0h	Odd Column Parity bit 2
16	P1O	R	0h	Odd Column Parity bit 1
15-12	RESERVED	R	0h	
11	P2048E	R	0h	Even Row Parity bit 2048, only used for ECC computed on 512 Bytes
10	P1024E	R	0h	Even Row Parity bit 1024
9	P512E	R	0h	Even Row Parity bit 512
8	P256E	R	0h	Even Row Parity bit 256
7	P128E	R	0h	Even Row Parity bit 128
6	P64E	R	0h	Even Row Parity bit 64
5	P32E	R	0h	Even Row Parity bit 32
4	P16E	R	0h	Even Row Parity bit 16
3	P8E	R	0h	Even Row Parity bit 8
2	P4E	R	0h	Even Column Parity bit 4
1	P2E	R	0h	Even Column Parity bit 2

Table 9-80. GPMC_ECC2_RESULT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	P1E	R	0h	Even Column Parity bit 1

9.1.6.30 GPMC_ECC3_RESULT Register (Offset = 208h) [reset = 0h]

GPMC_ECC3_RESULT is shown in Figure 9-77 and described in Table 9-81.

Return to Summary Table.

Figure 9-77. GPMC_ECC3_RESULT Register

31	30	29	28	27	26	25	24
RESERVED				P2048O	P1024O	P512O	P256O
R-0h				R-0h	R-0h	R-0h	R-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
P128O	P64O	P32O	P16O	P8O	P4O	P2O	P1O
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED				P2048E	P1024E	P512E	P256E
R-0h				R-0h	R-0h	R-0h	R-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
P128E	P64E	P32E	P16E	P8E	P4E	P2E	P1E
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 9-81. GPMC_ECC3_RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27	P2048O	R	0h	Odd Row Parity bit 2048, only used for ECC computed on 512 Bytes
26	P1024O	R	0h	Odd Row Parity bit 1024
25	P512O	R	0h	Odd Row Parity bit 512
24	P256O	R	0h	Odd Row Parity bit 256
23	P128O	R	0h	Odd Row Parity bit 128
22	P64O	R	0h	Odd Row Parity bit 64
21	P32O	R	0h	Odd Row Parity bit 32
20	P16O	R	0h	Odd Row Parity bit 16
19	P8O	R	0h	Odd Row Parity bit 8
18	P4O	R	0h	Odd Column Parity bit 4
17	P2O	R	0h	Odd Column Parity bit 2
16	P1O	R	0h	Odd Column Parity bit 1
15-12	RESERVED	R	0h	
11	P2048E	R	0h	Even Row Parity bit 2048, only used for ECC computed on 512 Bytes
10	P1024E	R	0h	Even Row Parity bit 1024
9	P512E	R	0h	Even Row Parity bit 512
8	P256E	R	0h	Even Row Parity bit 256
7	P128E	R	0h	Even Row Parity bit 128
6	P64E	R	0h	Even Row Parity bit 64
5	P32E	R	0h	Even Row Parity bit 32
4	P16E	R	0h	Even Row Parity bit 16
3	P8E	R	0h	Even Row Parity bit 8
2	P4E	R	0h	Even Column Parity bit 4
1	P2E	R	0h	Even Column Parity bit 2

Table 9-81. GPMC_ECC3_RESULT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	P1E	R	0h	Even Column Parity bit 1

9.1.6.31 GPMC_ECC4_RESULT Register (Offset = 20Ch) [reset = 0h]

GPMC_ECC4_RESULT is shown in Figure 9-78 and described in Table 9-82.

Return to Summary Table.

Figure 9-78. GPMC_ECC4_RESULT Register

31	30	29	28	27	26	25	24
RESERVED				P2048O	P1024O	P512O	P256O
R-0h				R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
P128O	P64O	P32O	P16O	P8O	P4O	P2O	P1O
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED				P2048E	P1024E	P512E	P256E
R-0h				R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
P128E	P64E	P32E	P16E	P8E	P4E	P2E	P1E
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 9-82. GPMC_ECC4_RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27	P2048O	R	0h	Odd Row Parity bit 2048, only used for ECC computed on 512 Bytes
26	P1024O	R	0h	Odd Row Parity bit 1024
25	P512O	R	0h	Odd Row Parity bit 512
24	P256O	R	0h	Odd Row Parity bit 256
23	P128O	R	0h	Odd Row Parity bit 128
22	P64O	R	0h	Odd Row Parity bit 64
21	P32O	R	0h	Odd Row Parity bit 32
20	P16O	R	0h	Odd Row Parity bit 16
19	P8O	R	0h	Odd Row Parity bit 8
18	P4O	R	0h	Odd Column Parity bit 4
17	P2O	R	0h	Odd Column Parity bit 2
16	P1O	R	0h	Odd Column Parity bit 1
15-12	RESERVED	R	0h	
11	P2048E	R	0h	Even Row Parity bit 2048, only used for ECC computed on 512 Bytes
10	P1024E	R	0h	Even Row Parity bit 1024
9	P512E	R	0h	Even Row Parity bit 512
8	P256E	R	0h	Even Row Parity bit 256
7	P128E	R	0h	Even Row Parity bit 128
6	P64E	R	0h	Even Row Parity bit 64
5	P32E	R	0h	Even Row Parity bit 32
4	P16E	R	0h	Even Row Parity bit 16
3	P8E	R	0h	Even Row Parity bit 8
2	P4E	R	0h	Even Column Parity bit 4
1	P2E	R	0h	Even Column Parity bit 2

Table 9-82. GPMC_ECC4_RESULT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	P1E	R	0h	Even Column Parity bit 1

9.1.6.32 GPMC_ECC5_RESULT Register (Offset = 210h) [reset = 0h]

GPMC_ECC5_RESULT is shown in Figure 9-79 and described in Table 9-83.

Return to Summary Table.

Figure 9-79. GPMC_ECC5_RESULT Register

31	30	29	28	27	26	25	24
RESERVED				P2048O	P1024O	P512O	P256O
R-0h				R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
P128O	P64O	P32O	P16O	P8O	P4O	P2O	P1O
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED				P2048E	P1024E	P512E	P256E
R-0h				R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
P128E	P64E	P32E	P16E	P8E	P4E	P2E	P1E
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 9-83. GPMC_ECC5_RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27	P2048O	R	0h	Odd Row Parity bit 2048, only used for ECC computed on 512 Bytes
26	P1024O	R	0h	Odd Row Parity bit 1024
25	P512O	R	0h	Odd Row Parity bit 512
24	P256O	R	0h	Odd Row Parity bit 256
23	P128O	R	0h	Odd Row Parity bit 128
22	P64O	R	0h	Odd Row Parity bit 64
21	P32O	R	0h	Odd Row Parity bit 32
20	P16O	R	0h	Odd Row Parity bit 16
19	P8O	R	0h	Odd Row Parity bit 8
18	P4O	R	0h	Odd Column Parity bit 4
17	P2O	R	0h	Odd Column Parity bit 2
16	P1O	R	0h	Odd Column Parity bit 1
15-12	RESERVED	R	0h	
11	P2048E	R	0h	Even Row Parity bit 2048, only used for ECC computed on 512 Bytes
10	P1024E	R	0h	Even Row Parity bit 1024
9	P512E	R	0h	Even Row Parity bit 512
8	P256E	R	0h	Even Row Parity bit 256
7	P128E	R	0h	Even Row Parity bit 128
6	P64E	R	0h	Even Row Parity bit 64
5	P32E	R	0h	Even Row Parity bit 32
4	P16E	R	0h	Even Row Parity bit 16
3	P8E	R	0h	Even Row Parity bit 8
2	P4E	R	0h	Even Column Parity bit 4
1	P2E	R	0h	Even Column Parity bit 2

Table 9-83. GPMC_ECC5_RESULT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	P1E	R	0h	Even Column Parity bit 1

9.1.6.33 GPMC_ECC6_RESULT Register (Offset = 214h) [reset = 0h]

GPMC_ECC6_RESULT is shown in Figure 9-80 and described in Table 9-84.

Return to Summary Table.

Figure 9-80. GPMC_ECC6_RESULT Register

31	30	29	28	27	26	25	24
RESERVED				P2048O	P1024O	P512O	P256O
R-0h				R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
P128O	P64O	P32O	P16O	P8O	P4O	P2O	P1O
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED				P2048E	P1024E	P512E	P256E
R-0h				R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
P128E	P64E	P32E	P16E	P8E	P4E	P2E	P1E
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 9-84. GPMC_ECC6_RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27	P2048O	R	0h	Odd Row Parity bit 2048, only used for ECC computed on 512 Bytes
26	P1024O	R	0h	Odd Row Parity bit 1024
25	P512O	R	0h	Odd Row Parity bit 512
24	P256O	R	0h	Odd Row Parity bit 256
23	P128O	R	0h	Odd Row Parity bit 128
22	P64O	R	0h	Odd Row Parity bit 64
21	P32O	R	0h	Odd Row Parity bit 32
20	P16O	R	0h	Odd Row Parity bit 16
19	P8O	R	0h	Odd Row Parity bit 8
18	P4O	R	0h	Odd Column Parity bit 4
17	P2O	R	0h	Odd Column Parity bit 2
16	P1O	R	0h	Odd Column Parity bit 1
15-12	RESERVED	R	0h	
11	P2048E	R	0h	Even Row Parity bit 2048, only used for ECC computed on 512 Bytes
10	P1024E	R	0h	Even Row Parity bit 1024
9	P512E	R	0h	Even Row Parity bit 512
8	P256E	R	0h	Even Row Parity bit 256
7	P128E	R	0h	Even Row Parity bit 128
6	P64E	R	0h	Even Row Parity bit 64
5	P32E	R	0h	Even Row Parity bit 32
4	P16E	R	0h	Even Row Parity bit 16
3	P8E	R	0h	Even Row Parity bit 8
2	P4E	R	0h	Even Column Parity bit 4
1	P2E	R	0h	Even Column Parity bit 2

Table 9-84. GPMC_ECC6_RESULT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	P1E	R	0h	Even Column Parity bit 1

9.1.6.34 GPMC_ECC7_RESULT Register (Offset = 218h) [reset = 0h]

GPMC_ECC7_RESULT is shown in Figure 9-81 and described in Table 9-85.

Return to Summary Table.

Figure 9-81. GPMC_ECC7_RESULT Register

31	30	29	28	27	26	25	24
RESERVED				P2048O	P1024O	P512O	P256O
R-0h				R-0h	R-0h	R-0h	R-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
P128O	P64O	P32O	P16O	P8O	P4O	P2O	P1O
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED				P2048E	P1024E	P512E	P256E
R-0h				R-0h	R-0h	R-0h	R-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
P128E	P64E	P32E	P16E	P8E	P4E	P2E	P1E
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 9-85. GPMC_ECC7_RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27	P2048O	R	0h	Odd Row Parity bit 2048, only used for ECC computed on 512 Bytes
26	P1024O	R	0h	Odd Row Parity bit 1024
25	P512O	R	0h	Odd Row Parity bit 512
24	P256O	R	0h	Odd Row Parity bit 256
23	P128O	R	0h	Odd Row Parity bit 128
22	P64O	R	0h	Odd Row Parity bit 64
21	P32O	R	0h	Odd Row Parity bit 32
20	P16O	R	0h	Odd Row Parity bit 16
19	P8O	R	0h	Odd Row Parity bit 8
18	P4O	R	0h	Odd Column Parity bit 4
17	P2O	R	0h	Odd Column Parity bit 2
16	P1O	R	0h	Odd Column Parity bit 1
15-12	RESERVED	R	0h	
11	P2048E	R	0h	Even Row Parity bit 2048, only used for ECC computed on 512 Bytes
10	P1024E	R	0h	Even Row Parity bit 1024
9	P512E	R	0h	Even Row Parity bit 512
8	P256E	R	0h	Even Row Parity bit 256
7	P128E	R	0h	Even Row Parity bit 128
6	P64E	R	0h	Even Row Parity bit 64
5	P32E	R	0h	Even Row Parity bit 32
4	P16E	R	0h	Even Row Parity bit 16
3	P8E	R	0h	Even Row Parity bit 8
2	P4E	R	0h	Even Column Parity bit 4
1	P2E	R	0h	Even Column Parity bit 2

Table 9-85. GPMC_ECC7_RESULT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	P1E	R	0h	Even Column Parity bit 1

9.1.6.35 GPMC_ECC8_RESULT Register (Offset = 21Ch) [reset = 0h]

GPMC_ECC8_RESULT is shown in Figure 9-82 and described in Table 9-86.

Return to Summary Table.

Figure 9-82. GPMC_ECC8_RESULT Register

31	30	29	28	27	26	25	24
RESERVED				P2048O	P1024O	P512O	P256O
R-0h				R-0h	R-0h	R-0h	R-0h

23	22	21	20	19	18	17	16
P128O	P64O	P32O	P16O	P8O	P4O	P2O	P1O
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

15	14	13	12	11	10	9	8
RESERVED				P2048E	P1024E	P512E	P256E
R-0h				R-0h	R-0h	R-0h	R-0h

7	6	5	4	3	2	1	0
P128E	P64E	P32E	P16E	P8E	P4E	P2E	P1E
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 9-86. GPMC_ECC8_RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27	P2048O	R	0h	Odd Row Parity bit 2048, only used for ECC computed on 512 Bytes
26	P1024O	R	0h	Odd Row Parity bit 1024
25	P512O	R	0h	Odd Row Parity bit 512
24	P256O	R	0h	Odd Row Parity bit 256
23	P128O	R	0h	Odd Row Parity bit 128
22	P64O	R	0h	Odd Row Parity bit 64
21	P32O	R	0h	Odd Row Parity bit 32
20	P16O	R	0h	Odd Row Parity bit 16
19	P8O	R	0h	Odd Row Parity bit 8
18	P4O	R	0h	Odd Column Parity bit 4
17	P2O	R	0h	Odd Column Parity bit 2
16	P1O	R	0h	Odd Column Parity bit 1
15-12	RESERVED	R	0h	
11	P2048E	R	0h	Even Row Parity bit 2048, only used for ECC computed on 512 Bytes
10	P1024E	R	0h	Even Row Parity bit 1024
9	P512E	R	0h	Even Row Parity bit 512
8	P256E	R	0h	Even Row Parity bit 256
7	P128E	R	0h	Even Row Parity bit 128
6	P64E	R	0h	Even Row Parity bit 64
5	P32E	R	0h	Even Row Parity bit 32
4	P16E	R	0h	Even Row Parity bit 16
3	P8E	R	0h	Even Row Parity bit 8
2	P4E	R	0h	Even Column Parity bit 4
1	P2E	R	0h	Even Column Parity bit 2

Table 9-86. GPMC_ECC8_RESULT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	P1E	R	0h	Even Column Parity bit 1

9.1.6.36 GPMC_ECC9_RESULT Register (Offset = 220h) [reset = 0h]

GPMC_ECC9_RESULT is shown in Figure 9-83 and described in Table 9-87.

Return to Summary Table.

Figure 9-83. GPMC_ECC9_RESULT Register

31	30	29	28	27	26	25	24
RESERVED				P2048O	P1024O	P512O	P256O
R-0h				R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
P128O	P64O	P32O	P16O	P8O	P4O	P2O	P1O
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED				P2048E	P1024E	P512E	P256E
R-0h				R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
P128E	P64E	P32E	P16E	P8E	P4E	P2E	P1E
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 9-87. GPMC_ECC9_RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27	P2048O	R	0h	Odd Row Parity bit 2048, only used for ECC computed on 512 Bytes
26	P1024O	R	0h	Odd Row Parity bit 1024
25	P512O	R	0h	Odd Row Parity bit 512
24	P256O	R	0h	Odd Row Parity bit 256
23	P128O	R	0h	Odd Row Parity bit 128
22	P64O	R	0h	Odd Row Parity bit 64
21	P32O	R	0h	Odd Row Parity bit 32
20	P16O	R	0h	Odd Row Parity bit 16
19	P8O	R	0h	Odd Row Parity bit 8
18	P4O	R	0h	Odd Column Parity bit 4
17	P2O	R	0h	Odd Column Parity bit 2
16	P1O	R	0h	Odd Column Parity bit 1
15-12	RESERVED	R	0h	
11	P2048E	R	0h	Even Row Parity bit 2048, only used for ECC computed on 512 Bytes
10	P1024E	R	0h	Even Row Parity bit 1024
9	P512E	R	0h	Even Row Parity bit 512
8	P256E	R	0h	Even Row Parity bit 256
7	P128E	R	0h	Even Row Parity bit 128
6	P64E	R	0h	Even Row Parity bit 64
5	P32E	R	0h	Even Row Parity bit 32
4	P16E	R	0h	Even Row Parity bit 16
3	P8E	R	0h	Even Row Parity bit 8
2	P4E	R	0h	Even Column Parity bit 4
1	P2E	R	0h	Even Column Parity bit 2

Table 9-87. GPMC_ECC9_RESULT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	P1E	R	0h	Even Column Parity bit 1

9.1.6.37 GPMC_BCH_RESULT0_0 Register (Offset = 240h) [reset = 0h]

GPMC_BCH_RESULT0_0 is shown in Figure 9-84 and described in Table 9-88.

Return to Summary Table.

Figure 9-84. GPMC_BCH_RESULT0_0 Register

31-0
BCH_RESULT0_0
R/W-0h

Table 9-88. GPMC_BCH_RESULT0_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT0_0	R/W	0h	BCH ECC result, bits 0 to 31

9.1.6.38 GPMC_BCH_RESULT1_0 Register (Offset = 244h) [reset = 0h]

GPMC_BCH_RESULT1_0 is shown in Figure 9-85 and described in Table 9-89.

Return to Summary Table.

Figure 9-85. GPMC_BCH_RESULT1_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT1_0																															
R/W-0h																															

Table 9-89. GPMC_BCH_RESULT1_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT1_0	R/W	0h	BCH ECC result, bits 32 to 63

9.1.6.39 GPMC_BCH_RESULT2_0 Register (Offset = 248h) [reset = 0h]

GPMC_BCH_RESULT2_0 is shown in Figure 9-86 and described in Table 9-90.

Return to Summary Table.

Figure 9-86. GPMC_BCH_RESULT2_0 Register

31-0
BCH_RESULT2_0
R/W-0h

Table 9-90. GPMC_BCH_RESULT2_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT2_0	R/W	0h	BCH ECC result, bits 64 to 95

9.1.6.40 GPMC_BCH_RESULT3_0 Register (Offset = 24Ch) [reset = 0h]

GPMC_BCH_RESULT3_0 is shown in Figure 9-87 and described in Table 9-91.

Return to Summary Table.

Figure 9-87. GPMC_BCH_RESULT3_0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BCH_RESULT3_0</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 9-91. GPMC_BCH_RESULT3_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT3_0	R/W	0h	BCH ECC result, bits 96 to 127

9.1.6.41 GPMC_BCH_RESULT0_1 Register (Offset = 250h) [reset = 0h]

GPMC_BCH_RESULT0_1 is shown in Figure 9-88 and described in Table 9-92.

Return to Summary Table.

Figure 9-88. GPMC_BCH_RESULT0_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT0_1																															
R/W-0h																															

Table 9-92. GPMC_BCH_RESULT0_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT0_1	R/W	0h	BCH ECC result, bits 0 to 31

9.1.6.42 GPMC_BCH_RESULT1_1 Register (Offset = 254h) [reset = 0h]

GPMC_BCH_RESULT1_1 is shown in Figure 9-89 and described in Table 9-93.

Return to Summary Table.

Figure 9-89. GPMC_BCH_RESULT1_1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BCH_RESULT1_1</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 9-93. GPMC_BCH_RESULT1_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT1_1	R/W	0h	BCH ECC result, bits 32 to 63

9.1.6.43 GPMC_BCH_RESULT2_1 Register (Offset = 258h) [reset = 0h]

GPMC_BCH_RESULT2_1 is shown in Figure 9-90 and described in Table 9-94.

Return to Summary Table.

Figure 9-90. GPMC_BCH_RESULT2_1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BCH_RESULT2_1</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 9-94. GPMC_BCH_RESULT2_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT2_1	R/W	0h	BCH ECC result, bits 64 to 95

9.1.6.44 GPMC_BCH_RESULT3_1 Register (Offset = 25Ch) [reset = 0h]

GPMC_BCH_RESULT3_1 is shown in Figure 9-91 and described in Table 9-95.

Return to Summary Table.

Figure 9-91. GPMC_BCH_RESULT3_1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BCH_RESULT3_1</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 9-95. GPMC_BCH_RESULT3_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT3_1	R/W	0h	BCH ECC result, bits 96 to 127

9.1.6.45 GPMC_BCH_RESULT0_2 Register (Offset = 260h) [reset = 0h]

GPMC_BCH_RESULT0_2 is shown in Figure 9-92 and described in Table 9-96.

Return to Summary Table.

Figure 9-92. GPMC_BCH_RESULT0_2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT0_2																															
R/W-0h																															

Table 9-96. GPMC_BCH_RESULT0_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT0_2	R/W	0h	BCH ECC result, bits 0 to 31

9.1.6.46 GPMC_BCH_RESULT1_2 Register (Offset = 264h) [reset = 0h]

GPMC_BCH_RESULT1_2 is shown in Figure 9-93 and described in Table 9-97.

Return to Summary Table.

Figure 9-93. GPMC_BCH_RESULT1_2 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BCH_RESULT1_2</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 9-97. GPMC_BCH_RESULT1_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT1_2	R/W	0h	BCH ECC result, bits 32 to 63

9.1.6.47 GPMC_BCH_RESULT2_2 Register (Offset = 268h) [reset = 0h]

GPMC_BCH_RESULT2_2 is shown in Figure 9-94 and described in Table 9-98.

Return to Summary Table.

Figure 9-94. GPMC_BCH_RESULT2_2 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BCH_RESULT2_2</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 9-98. GPMC_BCH_RESULT2_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT2_2	R/W	0h	BCH ECC result, bits 64 to 95

9.1.6.48 GPMC_BCH_RESULT3_2 Register (Offset = 26Ch) [reset = 0h]

GPMC_BCH_RESULT3_2 is shown in Figure 9-95 and described in Table 9-99.

Return to Summary Table.

Figure 9-95. GPMC_BCH_RESULT3_2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT3_2																															
R/W-0h																															

Table 9-99. GPMC_BCH_RESULT3_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT3_2	R/W	0h	BCH ECC result, bits 96 to 127

9.1.6.49 GPMC_BCH_RESULT0_3 Register (Offset = 270h) [reset = 0h]

GPMC_BCH_RESULT0_3 is shown in Figure 9-96 and described in Table 9-100.

Return to Summary Table.

Figure 9-96. GPMC_BCH_RESULT0_3 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BCH_RESULT0_3</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 9-100. GPMC_BCH_RESULT0_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT0_3	R/W	0h	BCH ECC result, bits 0 to 31

9.1.6.50 GPMC_BCH_RESULT1_3 Register (Offset = 274h) [reset = 0h]

GPMC_BCH_RESULT1_3 is shown in Figure 9-97 and described in Table 9-101.

Return to Summary Table.

Figure 9-97. GPMC_BCH_RESULT1_3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT1_3																															
R/W-0h																															

Table 9-101. GPMC_BCH_RESULT1_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT1_3	R/W	0h	BCH ECC result, bits 32 to 63

9.1.6.51 GPMC_BCH_RESULT2_3 Register (Offset = 278h) [reset = 0h]

GPMC_BCH_RESULT2_3 is shown in Figure 9-98 and described in Table 9-102.

Return to Summary Table.

Figure 9-98. GPMC_BCH_RESULT2_3 Register

31-0
BCH_RESULT2_3
R/W-0h

Table 9-102. GPMC_BCH_RESULT2_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT2_3	R/W	0h	BCH ECC result, bits 64 to 95

9.1.6.52 GPMC_BCH_RESULT3_3 Register (Offset = 27Ch) [reset = 0h]

GPMC_BCH_RESULT3_3 is shown in Figure 9-99 and described in Table 9-103.

Return to Summary Table.

Figure 9-99. GPMC_BCH_RESULT3_3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT3_3																															
R/W-0h																															

Table 9-103. GPMC_BCH_RESULT3_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT3_3	R/W	0h	BCH ECC result, bits 96 to 127

9.1.6.53 GPMC_BCH_RESULT0_4 Register (Offset = 280h) [reset = 0h]

GPMC_BCH_RESULT0_4 is shown in Figure 9-100 and described in Table 9-104.

Return to Summary Table.

Figure 9-100. GPMC_BCH_RESULT0_4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT0_4																															
R/W-0h																															

Table 9-104. GPMC_BCH_RESULT0_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT0_4	R/W	0h	BCH ECC result, bits 0 to 31

9.1.6.54 GPMC_BCH_RESULT1_4 Register (Offset = 284h) [reset = 0h]

GPMC_BCH_RESULT1_4 is shown in Figure 9-101 and described in Table 9-105.

Return to Summary Table.

Figure 9-101. GPMC_BCH_RESULT1_4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT1_4																															
R/W-0h																															

Table 9-105. GPMC_BCH_RESULT1_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT1_4	R/W	0h	BCH ECC result, bits 32 to 63

9.1.6.55 GPMC_BCH_RESULT2_4 Register (Offset = 288h) [reset = 0h]

GPMC_BCH_RESULT2_4 is shown in Figure 9-102 and described in Table 9-106.

Return to Summary Table.

Figure 9-102. GPMC_BCH_RESULT2_4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT2_4																															
R/W-0h																															

Table 9-106. GPMC_BCH_RESULT2_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT2_4	R/W	0h	BCH ECC result, bits 64 to 95

9.1.6.56 GPMC_BCH_RESULT3_4 Register (Offset = 28Ch) [reset = 0h]

GPMC_BCH_RESULT3_4 is shown in Figure 9-103 and described in Table 9-107.

Return to Summary Table.

Figure 9-103. GPMC_BCH_RESULT3_4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT3_4																															
R/W-0h																															

Table 9-107. GPMC_BCH_RESULT3_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT3_4	R/W	0h	BCH ECC result, bits 96 to 127

9.1.6.57 GPMC_BCH_RESULT0_5 Register (Offset = 290h) [reset = 0h]

GPMC_BCH_RESULT0_5 is shown in Figure 9-104 and described in Table 9-108.

Return to Summary Table.

Figure 9-104. GPMC_BCH_RESULT0_5 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BCH_RESULT0_5</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 9-108. GPMC_BCH_RESULT0_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT0_5	R/W	0h	BCH ECC result, bits 0 to 31

9.1.6.58 GPMC_BCH_RESULT1_5 Register (Offset = 294h) [reset = 0h]

GPMC_BCH_RESULT1_5 is shown in Figure 9-105 and described in Table 9-109.

Return to Summary Table.

Figure 9-105. GPMC_BCH_RESULT1_5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT1_5																															
R/W-0h																															

Table 9-109. GPMC_BCH_RESULT1_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT1_5	R/W	0h	BCH ECC result, bits 32 to 63

9.1.6.59 GPMC_BCH_RESULT2_5 Register (Offset = 298h) [reset = 0h]

GPMC_BCH_RESULT2_5 is shown in Figure 9-106 and described in Table 9-110.

Return to Summary Table.

Figure 9-106. GPMC_BCH_RESULT2_5 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BCH_RESULT2_5</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 9-110. GPMC_BCH_RESULT2_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT2_5	R/W	0h	BCH ECC result, bits 64 to 95

9.1.6.60 GPMC_BCH_RESULT3_5 Register (Offset = 29Ch) [reset = 0h]

GPMC_BCH_RESULT3_5 is shown in Figure 9-107 and described in Table 9-111.

Return to Summary Table.

Figure 9-107. GPMC_BCH_RESULT3_5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT3_5																															
R/W-0h																															

Table 9-111. GPMC_BCH_RESULT3_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT3_5	R/W	0h	BCH ECC result, bits 96 to 127

9.1.6.61 GPMC_BCH_RESULT0_6 Register (Offset = 2A0h) [reset = 0h]

GPMC_BCH_RESULT0_6 is shown in Figure 9-108 and described in Table 9-112.

Return to Summary Table.

Figure 9-108. GPMC_BCH_RESULT0_6 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BCH_RESULT0_6</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 9-112. GPMC_BCH_RESULT0_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT0_6	R/W	0h	BCH ECC result, bits 0 to 31

9.1.6.62 GPMC_BCH_RESULT1_6 Register (Offset = 2A4h) [reset = 0h]

GPMC_BCH_RESULT1_6 is shown in Figure 9-109 and described in Table 9-113.

Return to Summary Table.

Figure 9-109. GPMC_BCH_RESULT1_6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT1_6																															
R/W-0h																															

Table 9-113. GPMC_BCH_RESULT1_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT1_6	R/W	0h	BCH ECC result, bits 32 to 63

9.1.6.63 GPMC_BCH_RESULT2_6 Register (Offset = 2A8h) [reset = 0h]

GPMC_BCH_RESULT2_6 is shown in Figure 9-110 and described in Table 9-114.

Return to Summary Table.

Figure 9-110. GPMC_BCH_RESULT2_6 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BCH_RESULT2_6</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 9-114. GPMC_BCH_RESULT2_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT2_6	R/W	0h	BCH ECC result, bits 64 to 95

9.1.6.64 GPMC_BCH_RESULT3_6 Register (Offset = 2ACh) [reset = 0h]

GPMC_BCH_RESULT3_6 is shown in Figure 9-111 and described in Table 9-115.

Return to Summary Table.

Figure 9-111. GPMC_BCH_RESULT3_6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT3_6																															
R/W-0h																															

Table 9-115. GPMC_BCH_RESULT3_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT3_6	R/W	0h	BCH ECC result, bits 96 to 127

9.1.6.65 GPMC_BCH_SWDATA Register (Offset = 2D0h) [reset = 0h]

GPMC_BCH_SWDATA is shown in Figure 9-112 and described in Table 9-116.

Return to Summary Table.

This register is used to directly pass data to the BCH ECC calculator without accessing the actual NAND flash interface.

Figure 9-112. GPMC_BCH_SWDATA Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">BCH_DATA</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">R/W-0h</td></tr>
</table>

Table 9-116. GPMC_BCH_SWDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	BCH_DATA	R/W	0h	Data to be included in the BCH calculation. Only bits 0 to 7 are taken into account, if the calculator is configured to use 8 bits data (GPMC_ECC_CONFIG[7] ECC16B = 0).

9.1.6.66 GPMC_BCH_RESULT4_0 Register (Offset = 300h) [reset = 0h]

GPMC_BCH_RESULT4_0 is shown in Figure 9-113 and described in Table 9-117.

Return to Summary Table.

Figure 9-113. GPMC_BCH_RESULT4_0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BCH_RESULT4_0</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 9-117. GPMC_BCH_RESULT4_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT4_0	R/W	0h	BCH ECC result, bits 128 to 159

9.1.6.67 GPMC_BCH_RESULT5_0 Register (Offset = 304h) [reset = 0h]

GPMC_BCH_RESULT5_0 is shown in Figure 9-114 and described in Table 9-118.

Return to Summary Table.

Figure 9-114. GPMC_BCH_RESULT5_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT5_0																															
R/W-0h																															

Table 9-118. GPMC_BCH_RESULT5_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT5_0	R/W	0h	BCH ECC result, bits 160 to 191

9.1.6.68 GPMC_BCH_RESULT6_0 Register (Offset = 308h) [reset = 0h]

GPMC_BCH_RESULT6_0 is shown in Figure 9-115 and described in Table 9-119.

Return to Summary Table.

Figure 9-115. GPMC_BCH_RESULT6_0 Register

31 - 0
BCH_RESULT6_0
R/W-0h

Table 9-119. GPMC_BCH_RESULT6_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT6_0	R/W	0h	BCH ECC result, bits 192 to 207

9.1.6.69 GPMC_BCH_RESULT4_1 Register (Offset = 310h) [reset = 0h]

GPMC_BCH_RESULT4_1 is shown in Figure 9-116 and described in Table 9-120.

Return to Summary Table.

Figure 9-116. GPMC_BCH_RESULT4_1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BCH_RESULT4_1</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 9-120. GPMC_BCH_RESULT4_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT4_1	R/W	0h	BCH ECC result, bits 128 to 159

9.1.6.70 GPMC_BCH_RESULT5_1 Register (Offset = 314h) [reset = 0h]

GPMC_BCH_RESULT5_1 is shown in Figure 9-117 and described in Table 9-121.

Return to Summary Table.

Figure 9-117. GPMC_BCH_RESULT5_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT5_1																															
R/W-0h																															

Table 9-121. GPMC_BCH_RESULT5_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT5_1	R/W	0h	BCH ECC result, bits 160 to 191

9.1.6.71 GPMC_BCH_RESULT6_1 Register (Offset = 318h) [reset = 0h]

GPMC_BCH_RESULT6_1 is shown in Figure 9-118 and described in Table 9-122.

Return to Summary Table.

Figure 9-118. GPMC_BCH_RESULT6_1 Register

31-0
BCH_RESULT6_1
R/W-0h

Table 9-122. GPMC_BCH_RESULT6_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT6_1	R/W	0h	BCH ECC result, bits 192 to 207

9.1.6.72 GPMC_BCH_RESULT4_2 Register (Offset = 320h) [reset = 0h]

GPMC_BCH_RESULT4_2 is shown in Figure 9-119 and described in Table 9-123.

Return to Summary Table.

Figure 9-119. GPMC_BCH_RESULT4_2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT4_2																															
R/W-0h																															

Table 9-123. GPMC_BCH_RESULT4_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT4_2	R/W	0h	BCH ECC result, bits 128 to 159

9.1.6.73 GPMC_BCH_RESULT5_2 Register (Offset = 324h) [reset = 0h]

GPMC_BCH_RESULT5_2 is shown in Figure 9-120 and described in Table 9-124.

Return to Summary Table.

Figure 9-120. GPMC_BCH_RESULT5_2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT5_2																															
R/W-0h																															

Table 9-124. GPMC_BCH_RESULT5_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT5_2	R/W	0h	BCH ECC result, bits 160 to 191

9.1.6.74 GPMC_BCH_RESULT6_2 Register (Offset = 328h) [reset = 0h]

GPMC_BCH_RESULT6_2 is shown in Figure 9-121 and described in Table 9-125.

Return to Summary Table.

Figure 9-121. GPMC_BCH_RESULT6_2 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BCH_RESULT6_2</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 9-125. GPMC_BCH_RESULT6_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT6_2	R/W	0h	BCH ECC result, bits 192 to 207

9.1.6.75 GPMC_BCH_RESULT4_3 Register (Offset = 330h) [reset = 0h]

GPMC_BCH_RESULT4_3 is shown in Figure 9-122 and described in Table 9-126.

Return to Summary Table.

Figure 9-122. GPMC_BCH_RESULT4_3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT4_3																															
R/W-0h																															

Table 9-126. GPMC_BCH_RESULT4_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT4_3	R/W	0h	BCH ECC result, bits 128 to 159

9.1.6.76 GPMC_BCH_RESULT5_3 Register (Offset = 334h) [reset = 0h]

GPMC_BCH_RESULT5_3 is shown in Figure 9-123 and described in Table 9-127.

Return to Summary Table.

Figure 9-123. GPMC_BCH_RESULT5_3 Register

31-0
BCH_RESULT5_3
R/W-0h

Table 9-127. GPMC_BCH_RESULT5_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT5_3	R/W	0h	BCH ECC result, bits 160 to 191

9.1.6.77 GPMC_BCH_RESULT6_3 Register (Offset = 338h) [reset = 0h]

GPMC_BCH_RESULT6_3 is shown in Figure 9-124 and described in Table 9-128.

Return to Summary Table.

Figure 9-124. GPMC_BCH_RESULT6_3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT6_3																															
R/W-0h																															

Table 9-128. GPMC_BCH_RESULT6_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT6_3	R/W	0h	BCH ECC result, bits 192 to 207

9.1.6.78 GPMC_BCH_RESULT4_4 Register (Offset = 340h) [reset = 0h]

GPMC_BCH_RESULT4_4 is shown in Figure 9-125 and described in Table 9-129.

Return to Summary Table.

Figure 9-125. GPMC_BCH_RESULT4_4 Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
BCH_RESULT4_4
R/W-0h

Table 9-129. GPMC_BCH_RESULT4_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT4_4	R/W	0h	BCH ECC result, bits 128 to 159

9.1.6.79 GPMC_BCH_RESULT5_4 Register (Offset = 344h) [reset = 0h]

GPMC_BCH_RESULT5_4 is shown in Figure 9-126 and described in Table 9-130.

Return to Summary Table.

Figure 9-126. GPMC_BCH_RESULT5_4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT5_4																															
R/W-0h																															

Table 9-130. GPMC_BCH_RESULT5_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT5_4	R/W	0h	BCH ECC result, bits 160 to 191

9.1.6.80 GPMC_BCH_RESULT6_4 Register (Offset = 348h) [reset = 0h]

GPMC_BCH_RESULT6_4 is shown in Figure 9-127 and described in Table 9-131.

Return to Summary Table.

Figure 9-127. GPMC_BCH_RESULT6_4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT6_4																															
R/W-0h																															

Table 9-131. GPMC_BCH_RESULT6_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT6_4	R/W	0h	BCH ECC result, bits 192 to 207

9.1.6.81 GPMC_BCH_RESULT4_5 Register (Offset = 350h) [reset = 0h]

GPMC_BCH_RESULT4_5 is shown in Figure 9-128 and described in Table 9-132.

Return to Summary Table.

Figure 9-128. GPMC_BCH_RESULT4_5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT4_5																															
R/W-0h																															

Table 9-132. GPMC_BCH_RESULT4_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT4_5	R/W	0h	BCH ECC result, bits 128 to 159

9.1.6.82 GPMC_BCH_RESULT5_5 Register (Offset = 354h) [reset = 0h]

GPMC_BCH_RESULT5_5 is shown in Figure 9-129 and described in Table 9-133.

Return to Summary Table.

Figure 9-129. GPMC_BCH_RESULT5_5 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BCH_RESULT5_5</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 9-133. GPMC_BCH_RESULT5_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT5_5	R/W	0h	BCH ECC result, bits 160 to 191

9.1.6.83 GPMC_BCH_RESULT6_5 Register (Offset = 358h) [reset = 0h]

GPMC_BCH_RESULT6_5 is shown in Figure 9-130 and described in Table 9-134.

Return to Summary Table.

Figure 9-130. GPMC_BCH_RESULT6_5 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BCH_RESULT6_5</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 9-134. GPMC_BCH_RESULT6_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT6_5	R/W	0h	BCH ECC result, bits 192 to 207

9.1.6.84 GPMC_BCH_RESULT4_6 Register (Offset = 360h) [reset = 0h]

GPMC_BCH_RESULT4_6 is shown in Figure 9-131 and described in Table 9-135.

Return to Summary Table.

Figure 9-131. GPMC_BCH_RESULT4_6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT4_6																															
R/W-0h																															

Table 9-135. GPMC_BCH_RESULT4_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT4_6	R/W	0h	BCH ECC result, bits 128 to 159

9.1.6.85 GPMC_BCH_RESULT5_6 Register (Offset = 364h) [reset = 0h]

GPMC_BCH_RESULT5_6 is shown in Figure 9-132 and described in Table 9-136.

Return to Summary Table.

Figure 9-132. GPMC_BCH_RESULT5_6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT5_6																															
R/W-0h																															

Table 9-136. GPMC_BCH_RESULT5_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT5_6	R/W	0h	BCH ECC result, bits 160 to 191

9.1.6.86 GPMC_BCH_RESULT6_6 Register (Offset = 368h) [reset = 0h]

GPMC_BCH_RESULT6_6 is shown in Figure 9-133 and described in Table 9-137.

Return to Summary Table.

Figure 9-133. GPMC_BCH_RESULT6_6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT6_6																															
R/W-0h																															

Table 9-137. GPMC_BCH_RESULT6_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT6_6	R/W	0h	BCH ECC result, bits 192 to 207

9.1.6.87 GPMC_BCH_RESULT0_7 Register (Offset = 4B0h) [reset = 0h]

GPMC_BCH_RESULT0_7 is shown in Figure 9-134 and described in Table 9-138.

Return to Summary Table.

Figure 9-134. GPMC_BCH_RESULT0_7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT0_7																															
R/W-0h																															

Table 9-138. GPMC_BCH_RESULT0_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT0_7	R/W	0h	BCH ECC result, bits 0 to 31

9.1.6.88 GPMC_BCH_RESULT1_7 Register (Offset = 4B4h) [reset = 0h]

GPMC_BCH_RESULT1_7 is shown in Figure 9-135 and described in Table 9-139.

Return to Summary Table.

Figure 9-135. GPMC_BCH_RESULT1_7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT1_7																															
R/W-0h																															

Table 9-139. GPMC_BCH_RESULT1_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT1_7	R/W	0h	BCH ECC result, bits 32 to 63

9.1.6.89 GPMC_BCH_RESULT2_7 Register (Offset = 4B8h) [reset = 0h]

GPMC_BCH_RESULT2_7 is shown in Figure 9-136 and described in Table 9-140.

Return to Summary Table.

Figure 9-136. GPMC_BCH_RESULT2_7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT2_7																															
R/W-0h																															

Table 9-140. GPMC_BCH_RESULT2_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT2_7	R/W	0h	BCH ECC result, bits 64 to 95

9.1.6.90 GPMC_BCH_RESULT3_7 Register (Offset = 4BCh) [reset = 0h]

GPMC_BCH_RESULT3_7 is shown in Figure 9-137 and described in Table 9-141.

Return to Summary Table.

Figure 9-137. GPMC_BCH_RESULT3_7 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BCH_RESULT3_7</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 9-141. GPMC_BCH_RESULT3_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT3_7	R/W	0h	BCH ECC result, bits 96 to 127

9.1.6.91 GPMC_BCH_RESULT4_7 Register (Offset = 570h) [reset = 0h]

GPMC_BCH_RESULT4_7 is shown in Figure 9-138 and described in Table 9-142.

Return to Summary Table.

Figure 9-138. GPMC_BCH_RESULT4_7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT4_7																															
R/W-0h																															

Table 9-142. GPMC_BCH_RESULT4_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT4_7	R/W	0h	BCH ECC result, bits 128 to 159

9.1.6.92 GPMC_BCH_RESULT5_7 Register (Offset = 574h) [reset = 0h]

GPMC_BCH_RESULT5_7 is shown in Figure 9-139 and described in Table 9-143.

Return to Summary Table.

Figure 9-139. GPMC_BCH_RESULT5_7 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BCH_RESULT5_7</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 9-143. GPMC_BCH_RESULT5_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT5_7	R/W	0h	BCH ECC result, bits 160 to 191

9.1.6.93 GPMC_BCH_RESULT6_7 Register (Offset = 578h) [reset = 0h]

GPMC_BCH_RESULT6_7 is shown in Figure 9-140 and described in Table 9-144.

Return to Summary Table.

Figure 9-140. GPMC_BCH_RESULT6_7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT5_7																															
R/W-0h																															

Table 9-144. GPMC_BCH_RESULT6_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCH_RESULT5_7	R/W	0h	BCH ECC result, bits 192 to 207

9.2 OCMC-RAM

9.2.1 Introduction

OCMC-RAM Features

The on-chip memory controller consists of two separate modules that are OCP to memory wrappers. The first wrapper is for a ROM; the second is for a RAM. Each wrapper has its own dedicated interface to the L3 interconnect.

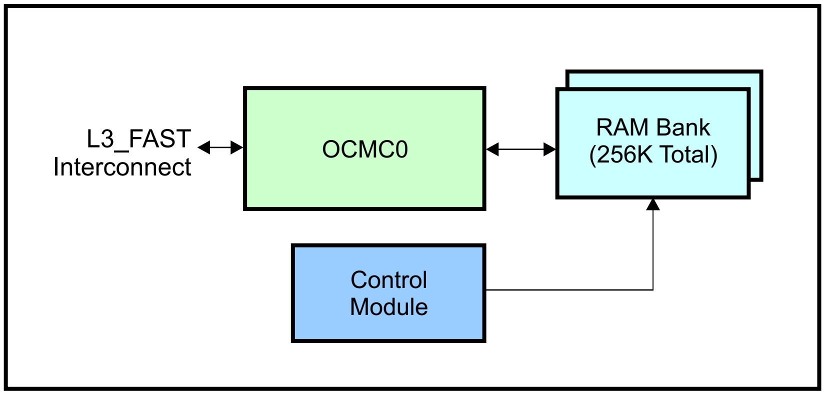
- 32- or 64-bit width
- Initial latency max 2 cycles (due to OCP to memory core wrapper).
- Multiple memory bank control based on address MSBs
- Full OCP IP 2.0 Burst support. No wait state.

Unsupported OCMC-RAM Features

For this device, the OCMC-RAM implementation does not support parity.

9.2.2 Integration

This device includes a single instantiation of the on-chip memory controller interfacing to a single 64K bank of RAM.



OCMC RAM Integration

OCMC RAM Connectivity Attributes

The general connectivity attributes for the OCMC RAM modules are summarized in OCMC RAM Connectivity Attributes.

OCMC RAM Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L3_GCLK
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	None
DMA Requests	None
Physical Address	L3 Fast slave port

OCMC RAM Clock and Reset Management

The OCMC module uses a single clock for the module and its OCP interface.

OCMC RAM Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
prcm_ocmc_clock Interface / Functional clock	200 MHz	CORE_CLKOUTM4	pd_per_l3_gclk From PRCM

OCMC RAM Pin List

The OCMC RAM module does not include any external interface pins.

9.3 EMIF

This section describes the external memory interface (EMIF) for the device.

9.3.1 Introduction

9.3.1.1 Features

The general features of the EMIF module are as follows:

- 16-bit and 32-bit data path to external SDRAM memory
- One 128-bit OCPIP 2.2 interface
- Support for the following memory types:
  - LPDDR2
  - DDR3

Supported External Memory Features:

- Memory device capacity
  - Up to 2GB addressability
- Flexible bank/row/column/chip-select address multiplexing schemes
- CAS latencies:
  - DDR3 => 5, 6, 7, 8, 9, 10, and 11
  - LPDDR2 => 3, 4, 5, 6, 7, and 8
- The following number of internal banks:
  - DDR3 => 1, 2, 4, and 8
  - LPDDR2 => 1, 2, 4, and 8
- Supports 256-, 512-, 1024-, and 2048-word page sizes
- Supports burst length of 8 (sequential burst)
  - DDR3 burst interrupt of BL8 not supported
  - LPDDR2 burst read is interruptible by another read
  - LPDDR2 burst write is interruptible by another write
- Write/read leveling/calibration and data eye training in conjunction with DID for DDR3
- Self-refresh and power-down modes for low power:
  - Flexible OCP to DDR address mapping to support Partial Array Self Refresh in LPDDR2 and DDR3.
  - Temperature-controlled self-refresh for LPDDR2
  - On-chip temperature sensor for DDR3
- Periodic ZQ calibration for LPDDR2 and DDR3
- ODT on DDR3
- Prioritized refresh scheduling
- Programmable SDRAM refresh rate and backlog counter
- Programmable SDRAM timing parameters
- Big and little endian modes

Unsupported External Memory Features:

- EMIF controller does not support LPDDR-NVM devices

9.3.2 Integration

9.3.2.1 EMIF Connectivity Attributes

The general connectivity attributes for the EMIF are shown in Table 9-145.

Table 9-145. EMIF Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_EMIF_L3_GCLK (OCP) PD_PER_EMIF_GCLK (Func)
Reset Signals	CORE_PWRON_RET_RST_N EMIF_DDR_PHY_PWRON_RST_N DLL_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	1 interrupt to MPU Subsystem (DDRERR0)
DMA Requests	None
Physical Address	L3 Fast Slave Port

9.3.2.2 EMIF Clock Management

The EMIF4 OCP interface (ocp_clk) is clocked by the L3 Fast clock sourced from the Core PLL. The DDR Command and Data macros are clocked by the DDR PLL. The PRCM divides this clock by two to create the EMIF functional clock (m_clk).

The OCP and functional clocks may be asynchronous because synchronization is managed in the EMIF4 internal FIFO (EMIF4 is set in asynchronous mode).

Table 9-146. EMIF Clock Signals

Clock Signal	Maximum Frequency	Reference Source	Comments
sys2e_clk (Interface clock)	200 MHz	CORE_CLKOUTM4	pd_per_emif_l3_gclk From PRCM
m_clk (EMIF functional clock)	100 MHz	DDR PLL CLKOUT / 2	pd_per_emif_gclk From PRCM
cmd0_dfi_clk cmd1_dfi_clk cmd2_dfi_clk data0_dfi_clk data1_dfi_clk data2_dfi_clk data3_dfi_clk (Macro clocks)	400 MHz	DDR PLL CLKOUT	clkout_po From DDR PLL
dll_clk	400 MHz	DDR PLL CLKOUTM4	ddr_dll_gclk From PRCM
dll_aging_clk	25 MHz (typ)	CLK_M_OSC	pd_per_dll_aging_gclk From PRCM

9.3.2.3 EMIF Pin List

Table 9-147 shows the EMIF/DDR external interface signals for each of the supported memory types. Note that LPDDR2 uses a multiplexed command/address interface that is different from the other memory types.

Table 9-147. EMIF Pin List

Pin	Type	DDR3	LPDDR2	Description
DDR_CK DDR_NCK	O	CK CKn	CK CKn	Differential clock pair
DDR_CKE0	O	CKE0	CKE0	Clock enable 0
DDR_CKE1	O	CKE1	CKE1	Clock enable 1
DDR_CSn0	O	CSn0	CSn0	Chip select 0
DDR_CSn1	O	CSn1	CSn1	Chip select 1
DDR_RASn	O	RASn	CA0	Row address strobe/Command/address[0]
DDR_CASn	O	CASn	CA1	Column address strobe/Command/address[1]
DDR_WEn	O	WEn	CA2	Write enable/Command/address[2]
DDR_BA0	O	BA0	CA7	Bank address[0]/Command/address[7]
DDR_BA1	O	BA1	CA8	Bank address[1]/Command/address[2]
DDR_BA2	O	BA2	CA9	Bank address[2]/Command/address[2]
DDR_A0	O	A0	not used	Row/column address[0]
DDR_A1	O	A1	CA5	Row/column address[1]/Command/address[5]
DDR_A2	O	A2	CA6	Row/column address[2]/Command/address[6]
DDR_A3	O	A3	not used	Row/column address[3]
DDR_A4	O	A4	not used	Row/column address[4]
DDR_A5	O	A5	not used	Row/column address[5]
DDR_A6	O	A6	not used	Row/column address[6]
DDR_A7	O	A7	not used	Row/column address[7]
DDR_A8	O	A8	not used	Row/column address[8]
DDR_A9	O	A9	not used	Row/column address[9]
DDR_A10	O	A10	CA4	Row/column address[10]/Command/address[4]
DDR_A11	O	A11	not used	Row/column address[11]
DDR_A12	O	A12	not used	Row/column address[12]
DDR_A13	O	A13	CA3	Row/column address[13]/Command/address[3]
DDR_A14	O	A14	not used	Row/column address[14]
DDR_A15	O	A15	not used	Row/column address[15]
DDR_DQS[3:0]	I/O	DQS[3:0]	DQS[3:0]	Data strobes
DDR_DQSn[3:0]	I/O	DQSn[3:0]	DQSn[3:0]	Complimentary data strobes
DDR_DQM[3:0]	O	DQM[3:0]	DM[3:0]	Data masks
DDR_D[31:0]	I/O	DQ[31:0]	DQ[31:0]	Data bus
DDR_ODT[1:0]	O	ODT[1:0]	not used	On-die termination
DDR_RESETn	O	not used	not used	DDR device reset
DDR_VREF(1)	I	VREF	VREF	I/O Voltage reference

(1) VREF is an analog input used to set the reference switching threshold of the DDR inputs and must be connected to the proper voltage source. It is not directly connected to the memory devices.

9.3.3 EMIF Functional Description

9.3.3.1 Block Diagram

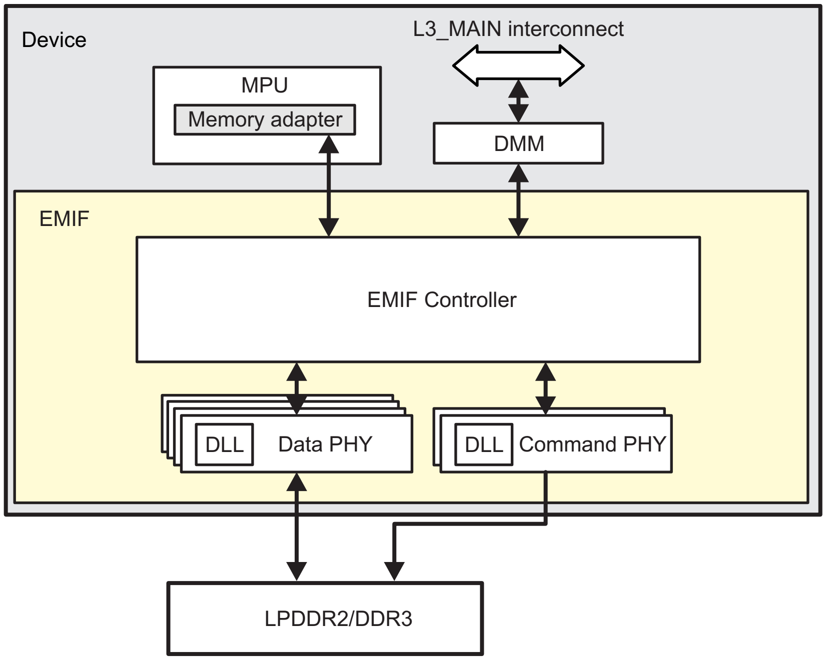
The EMIF module provides an interface to DDR3 and LPDDR2 SDRAM memories.

Figure 9-141 shows the interconnection between the EMIF module and the other modules.

Digital locked loops (DLLs) are used to delay the input DQS signals during reads so that these strobe signals can be used to latch incoming data on the DQ pins, as required by the DDR standard.

Physical layers (PHYs) convert single-data rate (SDR) signals to DDR signals.

Figure 9-141. EMIF Block Diagram



9.3.3.1.1 FIFO Description

The EMIF module contains the following FIFOs:

- Command FIFO
- Write data FIFO
- Return command FIFO
- Two read data FIFOs

Figure 9-142 shows the overall architecture of the EMIF FIFOs.

Figure 9-142. FIFO Block Diagram

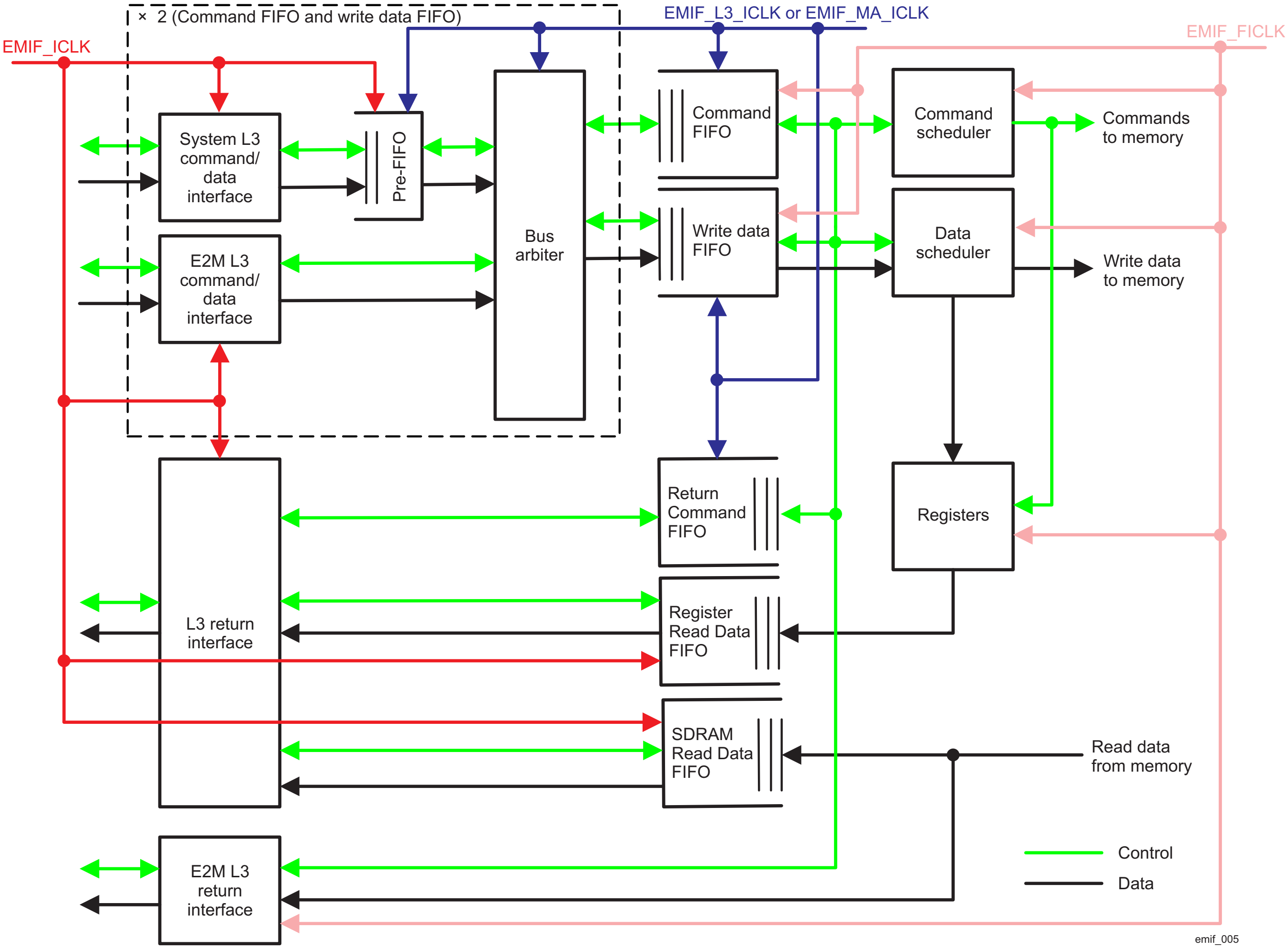


Table 9-148 lists the allocation of the entries.

Table 9-148. FIFO Allocation

Parameter	System Local Interface Entries	MPU Local Interface Entries
Pre Command FIFO	6	4
Command FIFO	Programmable (1)	Programmable (1)
Pre Write FIFO	6	8
Write Data FIFO (256-bit)	Up to (19 × 256 bits) + 6	Up to 19 + 8
Return Command FIFO	22	24
SDRAM Read Data FIFO	22	24
Register Read Data FIFO	2	0

(1) The total number of entries in the command FIFO is 10.

The EMIF has one shared command FIFO, one shared write FIFO and a separate return data FIFO for each of the three interface ports. Because the command FIFO allocation must be controlled in one clock domain, the SYS and LL OCP command and write interfaces are synchronized to the MPU clock domain in the pre-command and pre-write FIFOs. The allocation of each FIFO is shown in Table 9-148.

9.3.3.1.2 MPU Port Restrictions

The EMIF MPU port is defined only to process memory requests. All register accesses are processed through the system port of the EMIF. The EMIF MPU port does not support 2D or register requests required or provided by the system interface. When Class of Service is used, the MPU port has a fixed Connection ID equal to 0x0.

MFLAG is not supported, therefore the EMIF4D_READ_WRITE_EXECUTION_THR[31] MFLAG_OVERRIDE bit must always be set to 0x1. When this bit is set to 0x1 Class of Service is used.

To maintain coherency, the following rules must be followed:

- Any command arriving on MPU or system interface that matches an address in the command FIFO is executed after the command in the command FIFO
- The matching address is any address within a 2,000-address boundary
- On a 2D transfer, the starting address is the compared address. The computed addresses of the 2D transfer are not considered in address overlapping.
- Any command arriving within a 10-cycle window of another, from the different interfaces that do not match any address in the command FIFO, but may match command addresses arriving on a different interface, can be executed in any order.

9.3.3.1.3 Arbitration of Commands in the Command FIFO

The EMIF looks at all the commands stored in the command FIFO to schedule commands to the external memory. All commands with the same MConnID on a particular local interface complete in order. The EMIF does not ensure ordering between commands with different MConnIDs or between commands from two local interfaces. For more information about L3_MAIN interconnect terms (MConnID, MCMD, MADDR, etc.) see Section 4.1.2, *L3 Interconnect*.

However, the EMIF does maintain data coherency. Therefore, the EMIF blocks a command, regardless of priority or the local interface, if that command is to the same block address (2048 bytes) as an older command that is not complete. Thus, the EMIF may have one pending read or write for each MConnID. Among all pending reads, the EMIF selects all reads that have their corresponding SDRAM banks already open. Similarly, among all pending writes, the EMIF selects all writes that have their corresponding SDRAM banks already open. Accesses to memory mapped registers are treated as accesses that have open banks.

As a result of this reordering, the EMIF may now have several pending reads and writes that have their corresponding banks open. The EMIF then selects the highest priority read from pending reads, and the highest priority write from pending writes. If two or more commands have the highest priority, the EMIF selects the oldest command. As a result, the EMIF may now have the next read and a write command. If the return command FIFO and the read data FIFO have space and the external bus conflict is resolved, the EMIF performs the final read command before the final write command. If the return command FIFO has space but the read data FIFO is full, the EMIF performs the final write command before the final read command. Resolution of external bus conflict means all the SDRAM command-to-command counters are satisfied and the read-to-write or write-to-read turnaround time is met.

The EMIF does not support tag interleaving. In other words, for an local interface, the EMIF completes executing an local command before it switches to another command. The EMIF can, however, interleave execution between commands from two local interfaces.

The data coherency inside the EMIF is ensured only in a single level of local infrastructure. For example, if a write from a secondary local bus segment is blocked by a bridge element, the read from a tertiary bus can still beat the write to the EMIF. In such a case, to confirm that a write from master A has landed before a read from master B is performed, master A must wait for the write status from the EMIF before indicating to master B that the data is ready to be read. If master A does not use the local wait status, it must do the following:

1. Perform the required write.
2. Perform a dummy write to the EMIF4D_MOD_ID_REV register.
3. Perform a dummy read to the EMIF4D_MOD_ID_REV register.
4. Indicate to master B that the data is ready to be read after completion of read in Step 3. The completion of read in Step 3 ensures that the previous writes were done.

Apart from reads and writes, the EMIF must also open and close SDRAM banks and maintain the refresh counts for an SDRAM. The priority of SDRAM commands with respect to refresh levels are:

1. SDRAM refresh request when refresh-must level is reached (highest priority)
2. ZQ calibration
3. Leveling
4. local request for a read or write
5. local request for a write
6. SDRAM activate commands
7. SDRAM deactivate commands
8. SDRAM power-down request
9. SDRAM deep power-down request
10. SDRAM refresh request when refresh-may or release level is reached
11. SDRAM self-refresh request (lowest priority)

To avoid continuous blocking effect which can be caused by a continuous stream of high-priority commands which thus block the lower priority commands, the EMIF momentarily raises the priority of the oldest command over all other commands when the time for the oldest command configured through the EMIF4D_COS_CONFIG[7:0] PR_OLD_COUNT bit field expires.

It should be taken into account that while performing the scheduling algorithm described, the EMIF may also encounter a condition in which continuous stream of SDRAM commands to a row in an open bank can block commands to another row in the same bank.

In addition to this scheduling, the highest priority condition is a reset command. If this condition occurs, the EMIF abandons what it is currently doing and begins its start-up sequence. In this case, commands and data stored in the FIFOs are lost. The EMIF also starts its start-up sequence whenever the EMIF4D_SDRAM_CONFIG register is written and the EMIF4D_SDRAM_REFRESH_CTRL[31] INITREF_DIS bit is set to 0. In this case, commands and data stored in the FIFOs are not lost. The EMIF ensures that in-flight read or write transactions to the SDRAM are complete before starting the initialization sequence.

All the accesses to an SDRAM are pipelined to maximize use of the external bus. All of these are done while fulfilling the access timing requirements of an SDRAM.

9.3.3.2 Clock Management

The EMIF can gate EMIF_FICLK. There is an internal mechanism that can stop EMIF_FICLK automatically. EMIF_FICLK is stopped only after the SDRAM is put into self-refresh mode and the power-idle protocol on the local bus completes. The EMIF_FICLK frequency can be changed only after putting the external SDRAM in self-refresh mode.

The EMIF waits for the DLL lock before performing any memory access.

EMIF_FICLK frequency is equal to half of the EMIF_PHY_FCLK frequency.

9.3.3.3 Reset

The EMIF does not support a software reset.

The EMIF supports a global warm reset mode, during which the EMIF keeps the SDRAM content. Upon a request from the PRCM module indicating a need to enter global warm reset mode, the EMIF does the following:

1. During leveling operation, EMIF will immediately exit this mode and automatically perform a write to the MR1 register of DDR3 memory to disable the leveling at the memory side too.
2. EMIF completes the ongoing access, and then puts the SDRAM in self-refresh mode. If the EMIF4D_SDRAM_REFRESH_CTRL[31] INITREF_DIS field is set to 1, the EMIF does not put SDRAM in self-refresh mode. If EMIF is in LPDDR2 mode and if the reg_refresh_en field in the LPDDR2 Mode Reg Config register is set to 0, the EMIF does not put LPDDR2 in self-refresh.
3. EMIF clears all its FIFO contents.
4. EMIF does not wait for all interrupts to be serviced.

To exit the global warm reset:

1. If the EMIF was in Self Refresh state, it will exit Self Refresh state.
2. If leveling was enabled at the time of a global warm reset, a PHY reset must occur to bring the PHY back into a known state, as it may have been left in a leveling state upon warm reset assertion. To guarantee that the SDRAM memory clocks are off when issuing PHY reset, software can use the EMIF4D_POWER_MANAGEMENT_CTRL register to enter self refresh before asserting the PHY reset.

9.3.3.4 System Power Management

9.3.3.4.1 Self-Refresh Mode

The EMIF supports SDRAM self-refresh mode for low power. The EMIF automatically puts the SDRAM into self-refresh mode after the EMIF is idle for EMIF4D_POWER_MANAGEMENT_CTRL[7:4] SR_TIMING number of DDR clock cycles and the EMIF4D_POWER_MANAGEMENT_CTRL[10:8] LP_MODE bit field is set to 0x2. The EMIF will complete all pending refreshes before it puts the SDRAM into self-refresh. Therefore, after the expiration of SR_TIMING, the EMIF will start issuing refreshes to complete the refresh backlog down to the refresh release level (given that the LPDDR2 requirement of tREFBW is met) and then issues a SELF-REFRESH command to the SDRAM.

In self-refresh mode, the EMIF automatically stops the SDRAM clock. The EMIF drives CKE pin low to maintain self-refresh mode.

When the SDRAM is in self-refresh mode, the EMIF services register accesses normally.

If the reg_lp_mode field is a value other than 2, or an SDRAM access is requested while it is in self-refresh, and reg_t_cke + 1 cycles have elapsed since the SELF-REFRESH command was issued, the EMIF will bring the SDRAM out of self-refresh. The value of reg_t_cke is taken from SDRAM Timing 2 register. For DDR3, EMIF will also exit self-refresh to perform incremental leveling

For LPDDR2, the EMIF:

- Drives phy_sdramclkstop low to enable clocks.
- Drives pad_cke_o high.
- Waits for reg_t_xsnr + 1 cycles. The value of reg_t_xsnr is taken from SDRAM Timing 2 register.
- Starts an auto-refresh cycle in the next cycle. It also services all refreshes down to the refresh_release level.
- Enters its idle state and can issue any commands.

For DDR3, the EMIF:

- Drives phy_sdramclkstop low to enable clocks.
- Drives pad_cke_o high.
- Waits for reg_t_xsnr + 1 cycles. The value of reg_t_xsnr is taken from SDRAM Timing 2 register.
- If the reg_ddr_disable_dll bit in the SDRAM Config register is 1, the EMIF issues a LOAD MODE REGISTER command to the extended mode register 1 (pad_ba_o[2:0] = 0x1) with the pad_a_o bits

set as follows:

Bits	Value	Description
pad_a_o[15:13]	0x0	Reserved
pad_a_o[12]	0x0	Output buffer enabled
pad_a_o[11]	0x0	TDQS disable
pad_a_o[10]	0x0	Reserved
pad_a_o[9]	reg_ddr_term[2]	DDR3 termination resistor value from SDRAM Config register
pad_a_o[8]	0x0	Reserved
pad_a_o[7]	0x0	Write leveling disabled
pad_a_o[6]	reg_ddr_term[1]	DDR3 termination resistor value from SDRAM Config register
pad_a_o[5]	reg_sdram_drive[1]	SDRAM drive strength from SDRAM Config register
pad_a_o[4:3]	0x0	Additive latency = 0
pad_a_o[2]	reg_ddr_term[0]	DDR3 termination resistor value from SDRAM Config register
pad_a_o[1]	reg_sdram_drive[0]	SDRAM drive strength from SDRAM Config register
pad_a_o[0]	0x0	Disable DLL

- Starts an auto-refresh cycle in the next cycle. It also services all refreshes down to the refresh_release level.
- Enters its idle state and can issue any other commands except a write or a read. A write or a read will only be issued after reg_t_xsrd + 1 clock cycles have elapsed since pad_cke_o is driven high. The value of reg_t_xsrd is taken from SDRAM Timing 2 register.

To use partial array self-refresh, the EMIF4D_SDRAM_REFRESH_CTRL[26:24] PASR bits must be appropriately programmed. The EMIF performs bank interleaving when EMIFRD_SDRAM_CONFIG[28:27] IBANK_POS = 0x0. Because the SDRAM is partially refreshed during partial array self-refresh, for software ease, it is recommended that the IBANK_POS bit field to be set to 0x1, 0x2, or 0x3 depending on the scheme used. If IBANK_POS is set to 0x0, software must move critical data into the banks that are going to be refreshed during partial array self-refresh.

9.3.3.4.2 Power-Down Mode

The EMIF supports SDRAM power-down mode for low power. The EMIF automatically puts the SDRAM into power-down mode after it is idle for EMIF4D_POWER_MANAGEMENT_CTRL[15:12] PD_TIM number of DDR clock cycles and the EMIF4D_POWER_MANAGEMENT_CTRL[10:8] LP_MODE bit field is set to 0x4.

If refresh-must level is not reached before power-down entering, EMIF will not precharge all SDRAM banks before it issues the power-down command. As a result of this EMIF puts the SDRAM in active power-down mode. If refresh-must level is reached before power-down entering, EMIF will precharge all SDRAM banks and before it issues the power-down command, EMIF issues refreshes until refresh-release level is reached. As a result of this EMIF puts the SDRAM in precharge power-down mode.

In power-down mode, the EMIF does not stop the clocks to the SDRAM. The EMIF maintains the CKE pin low to maintain the power-down mode.

When the SDRAM is in power-down mode, the EMIF services register accesses normally.

The EMIF brings SDRAM out of power-down mode if the SDRAM is in power-down mode and one of the following occurs:

- EMIF4D_POWER_MANAGEMENT_CTRL[10:8] LP_MODE bit field is set to a value other than 0x4.
- An SDRAM access is requested.
- The refresh-must level is reached while the SDRAM/NVM is in power-down mode.

For DDR3, the EMIF also exits power-down mode to perform incremental leveling. If the refresh_must level brings the SDRAM out of power-down mode, EMIF will re-enter power-down mode when the refreshes are complete if there is no SDRAM request.

To exit power-down, the EMIF:

1. Drives CKE high after t_cke + 1 cycles have elapsed since the power-down command was issued. The value of t_cke is taken from EMIF4D_SDRAM_TIMING_2[2:0] T_CKE bit field.
2. Waits for EMIF4D_SDRAM_TIMING_2[30:28] T_XP + 1 cycles
3. Enters its idle state and can issue any commands

9.3.3.4.3 Deep Power-Down Mode

To save the most power, the EMIF supports deep power-down mode for LPDDR2.

The SDRAM can be forced into deep power-down through software by setting to 1 the DPD_EN field in the EMIF4D_POWER_MANAGEMENT_CTRL register. In this case, the EMIF will continue normal operation until all SDRAM memory access requests have been serviced. At this point the EMIF will issue a DEEP POWER-DOWN command. The EMIF then maintains pad_cke_o low to maintain the deep power-down state.

Setting the EMIF4D_POWER_MANAGEMENT_CTRL[11] DPD_EN field to 1 overrides the setting of EMIF4D_POWER_MANAGEMENT_CTRL[10:8] LP_MODE field. Therefore, if the SDRAM is in self-refresh or power-down mode, and EMIF4D_POWER_MANAGEMENT_CTRL[11] DPD_EN field is set to 1, the EMIF will exit those modes and enter deep power-down mode.

When the SDRAM is in deep power-down, the EMIF services register accesses as normal. If the EMIF4D_POWER_MANAGEMENT_CTRL[11] DPD_EN field is set to 0, or an SDRAM access is requested, the EMIF will bring the SDRAM out of deep power-down by:

- Performing SDRAM initialization, as specified in the LPDDR2 initialization section (Section 9.3.3.8).
- Entering its idle state where it can issue any command.
- Using software to perform initialization, as specified in the LPDDR2 Initialization section (Section 9.3.3.8).

Because the EMIF performs initialization upon exiting deep power-down mode, the REFRESH_RATE field in the EMIF4D_SDRAM_REFRESH_CTRL register must be set appropriately to meet the 200us wait requirement for LPDDR2 during initialization.

9.3.3.4.4 Save and Restore Mode

The LPDDR2/DDR3 memory controller supports a save and restore mechanism to completely switch off power to the LPDDR2/DDR3 memory controller. The following sequence of operations is followed to put the LPDDR2/DDR3 memory controller in off mode:

1. An external master reads the following memory mapped registers and saves their value external to the LPDDR2/DDR3 memory controller.
    - SDRAM Config register (SDRCR)
    - SDRAM Config 2 register
    - SDRAM Refresh Control register (SDRRCR)
    - SDRAM Refresh Control Shadow register (SDRRCSR)
    - SDRAM Timing 1 register (SDRTIM1)
    - SDRAM Timing 1 Shadow register (SDRTIM1SR)
    - SDRAM Timing 2 register (SDRTIM2)
    - SDRAM Timing 2 Shadow register (SDRTIM2SR)
    - SDRAM Timing 3 register (SDRTIM3)
    - SDRAM Timing 3 Shadow register (SDRTIM3SR)
    - Power Management Control register (PMCR)
    - Power Management Control Shadow register (PMCSR)
    - Interface Configuration register (OCP_CONFIG)
    - System OCP Interrupt Enable Set Register (SOIESR)
    - DDR PHY Control 1 register (DDRPHYCR)
    - DDR PHY Control 1 Shadow register (DDRPHYCSR)

2. Memory controller completes all pending transactions and drains all its FIFOs.
3. Memory controller puts the SDRAM in self-refresh.
4. Memory controller copies all shadow memory mapped registers to its main registers. It is assumed that the shadow register always has the same value as its corresponding main register.
5. Memory controller waits for all interrupts to be serviced.
6. Memory controller acknowledges assertion of internal power down request.

9.3.3.5 Interrupt Support

The EMIF only supports Idle, Write, Read, and WriteNonPost command types (as indicated by MCmd). Also, the EMIF only supports incrementing, wrapping, and two-dimensional block addressing modes (as indicated by MBurstSeq). If an access request for an unsupported command type or addressing mode is received, the EMIF will set the ERR_SYS or ERR_LL bit (depending on which interface sent the error command) in the Interrupt Raw Status register. The EMIF will also set these bits if an access request to an unsupported MAddrSpace is received.

The EMIF will also set the reg_ta_sys or reg_ta_ll bit in Interrupt Raw Status register if a change in SDRAM temperature is detected. For more details, see Section 9.3.3.15, *Temperature Monitoring*.

The EMIF will only output the interrupts on the interrupt lines if they are enabled by writing a 1 to the corresponding bits in the Interrupt Enable Set register. The interrupts can be disabled by writing a 1 to the corresponding bits in the Interrupt Enable Clear register.

When enabled, the corresponding bits in the Interrupt Status register will also be set if the above error condition occurs. The interrupts can be cleared once serviced by writing a 1 to the corresponding bits either in the Interrupt Raw Status or Interrupt Status register. The software must also write to the End of Interrupt register to indicate that the interrupt was serviced.

9.3.3.6 SDRAM Refresh Scheduling

The EMIF uses two counters to schedule the Refresh (REF) commands: a 13-bit decrementing refresh interval counter and a 4-bit refresh backlog counter. The interval counter is used to define the rate at which connected SDRAM devices are refreshed. It is loaded with the value of the EMIF4D_SDRAM_REFRESH_CTRL[15:0] REFRESH_RATE bit field at reset (only the 13 LSBs are taken). The interval counter decrements by 1 each cycle until it reaches 0x0, at which point it reloads from the EMIF4D_SDRAM_REFRESH_CTRL[15:0] REFRESH_RATE bit field and restarts decrementing. The counter also reloads and restarts decrementing whenever the EMIF4D_SDRAM_REFRESH_CTRL[15:0] REFRESH_RATE bit field is updated.

The refresh backlog counter records the number of the outstanding REF commands which the EMIF controller currently has. The backlog counter increments by 1 each time the interval counter reloads (unless it has reached its maximum value of 8). The backlog counter decrements by 1 each time the EMIF issues a REF command (unless it is already 0). For the range of values that the backlog counter can take, there are three levels of urgency with which the EMIF must perform refresh cycle in which it issues REF commands:

1. Refresh-may level is reached when the backlog count is greater than 0x0, which indicates that there is a refresh backlog and if the EMIF is not busy and there are no open SDRAM banks, the EMIF must perform refresh cycle.
2. Refresh-release level is reached when the backlog count is greater than 0x4, which indicates that the refresh backlog is getting bigger and if the EMIF is not busy it must perform refresh cycle even if there is an open SDRAM bank.
3. Refresh-must level is reached when the backlog count is greater than 0x7, which indicates that the refresh backlog is becoming excessive and the EMIF must perform refresh cycle before any new memory access request being serviced. The EMIF starts servicing new memory accesses after the refresh-release level is cleared.

The refresh counters do not operate when the SDRAM is in self-refresh mode. The refresh counters also start tracking the missed refreshes only after initialization is complete for all DDR types except LPDDR2. For LPDDR2, the refresh counters starts tracking missed refreshes after the RESET command is issued. In addition for LPDDR2, the EMIF will ensure that no more than 8 AUTO REFRESH commands are issued in any rolling tREFBW (=4*8*tRFC) window.

The time between two REF commands is set through the EMIF4D_SDRAM_TIMING_3[12:4] T_RFC bit field.

9.3.3.7 SDRAM Initialization

9.3.3.7.1 DDR3 SDRAM Initialization

NOTE: The EMIF does not perform any transactions until the DDR3 initialization sequence is complete.

On coming out of reset, the EMIF performs a DDR3 SDRAM initialization sequence as follows if EMIF4D_SDRAM_CONFIG[31:29] SDRAM_TYPE is equal to 0x3 and EMIF4D_SDRAM_REFRESH_CTRL[31] INITREF_DIS bit is set to 0x0:

1. Drives the CKE pin low
2. After 16 SDRAM refresh rate intervals, issues a NOP command with CKE pin held high. The SDRAM refresh rate is as defined in the EMIF4D_SDRAM_REFRESH_CTRL[15:0] REFRESH_RATE bit field.
3. After one SDRAM refresh rate interval, issues MRS command to the DDR3 MR2 register (bits BA[2:0] = 0x2) with bits A[15:0] set as in Table 9-149

Table 9-149. Load Value For The MR2 Register During DDR3 SDRAM Initialization

Bits	Value	Description
A[15:11]	0x0	Reserved
A[10:9]	EMIF4D_SDRAM_CONFIG[22:21] DYN_ODT	Dynamic ODT value
A[8]	0x0	Reserved
A[7]	EMIF4D_SDRAM_REFRESH_CTRL [29] SRT	Self-refresh temperature range
A[6]	EMIF4D_SDRAM_REFRESH_CTRL [28] ASR	Auto self-refresh enable
A[5]	0x0	Reserved
A[4:3]	EMIF4D_SDRAM_CONFIG[17:16] CWL	CAS write latency
A[2:0]	EMIF4D_SDRAM_REFRESH_CTRL [26:24] PASR	Partial array self-refresh

4. Issues MRS command to the DDR3 MR3 register (bits BA[2:0] = 0x3) with A[15:0] = 0x0
5. Issues MRS command to the DDR3 MR1 register (BA[2:0] = 0x1) with A[15:0] set as in Table 9-150

Table 9-150. Load Value For The MR1 Register During DDR3 SDRAM Initialization

Bits	Value	Description
A[15:13]	0x0	Reserved
A[12]	0x0	Output buffer enabled
A[11]	0x0	TDQS disable
A[10]	0x0	Reserved
A[9], A[6], A[2]	EMIF4D_SDRAM_CONFIG[26:24] DDR_TERM	DDR3 termination resistor value
A[8]	0x0	Reserved
A[7]	0x0	Write leveling disabled

Table 9-150. Load Value For The MR1 Register During DDR3 SDRAM Initialization (continued)

Bits	Value	Description
A[5], A[1]	EMIF4D_SDRAM_CONFIG[19:18] SDRAM_DRIVE	SDRAM drive strength
A[4:3]	0x0	Additive latency = 0
A[0]	EMIF4D_SDRAM_CONFIG[20] DDR_DISABLE_DLL = 0x0	Enable SDRAM DLL

6. Issues MRS command to the DDR3 MR0 register (BA[2:0] = 0x0) with A[15:0] set as in Table 9-151

Table 9-151. Load Value For The MR0 Register During DDR3 SDRAM Initialization

Bits	Value	Description
A[15:13]	0x0	Reserved
A[12]	0x1	The DDR3 SDRAM DLL is "ON" after precharge power-down entering
A[11:9]	EMIF4D_SDRAM_TIMING_1[20:17] T_WR	Write recovery for autoprecharge
A[8]	0x1	DLL reset
A[7]	0x0	Normal mode
A[6:4], A[2]	EMIF4D_SDRAM_CONFIG[13:10] CL	Value for CAS latency
A[3]	0x0	Nibble sequential read burst type
A[1:0]	0x0	Burst length of 8

7. Issues a ZQCL command to start long ZQ calibration
8. Waits for t_{DLLK} and t_{ZQinit} to complete
9. Issues REF command
10. The EMIF enters its IDLE state.

The EMIF also performs the initialization sequence whenever the EMIF4D_SDRAM_CONFIG register is written. But in this case, the EMIF starts from Step 3.

When the EMIF comes out of reset, the delay time in Step 2 resulting from the 16 refresh rate intervals + 8 cycles is approximately 16 × REFRESH_RATE / input frequency.

9.3.3.8 LPDDR2 Initialization

NOTE: The EMIF does not perform any transactions until the LPDDR2 initialization sequence is complete.

On coming out of reset, the EMIF performs an LPDDR2 initialization sequence as follows if the reg_sdram_type field in the SDRAM Config register is equal to 4:

1. Drives CKE pin high and starts to continuously issues NOP commands.
2. After 16 SDRAM refresh rate intervals, issues a PRECHARGE-ALL command. The SDRAM refresh rate is as defined in the reg_refresh_rate field description (see description of SDRAM Refresh Control register).
3. Issues a RESET command.
4. The EMIF enters its IDLE state.
5. The software then performs the LPDDR2 initialization using the EMIF's LPDDR2 Mode Reg Config and LPDDR2 Mode Reg Data registers. The software must enable refreshes by writing a 1 to the reg_refresh_en field in the LPDDR2 Mode Reg Config register during the last MRW command.

The EMIF also performs the initialization sequence whenever the SDRAM Config register is written to and the LPDDR2 initialization was not previously performed because reg_initref_dis was set to 0. Once the EMIF performs initialization, re-writing the SDRAM Config register will not cause re-initialization.

The reg_refresh_rate value at reset is the config_refresh_def_val port value. When the EMIF comes out of reset, the delay time in step 2 resulting from the 16 refresh rate intervals and 8 cycles is approximately 16 * reg_refresh_rate / input frequency. It is up to the user to tie off the config_refresh_def_val port with a correct value to meet the typical LPDDR2 device-specified delay time of 200us between power-up and the application of RESET command.

9.3.3.9 DDR3 Read-Write Leveling

The EMIF supports read-write leveling in conjunction with the DDR PHY for DDR3. The EMIF supports two types of write/read leveling:

- Full leveling

  The EMIF does not perform full leveling after initialization upon reset deassertion. Full leveling must be triggered by software after the EMIF MMR’s are properly configured. The EMIF also supports triggering of full leveling through software through the use of the reg_rdwrlvlfull_en field in the Read-Write Leveling Control register. Full leveling will be interrupted by auto refresh, if a refresh must state is reached. Upon completion of the auto refresh, full leveling will be re-entered at the point it exited and continue till completion.

- Incremental leveling

  The EMIF supports incremental leveling to better track voltage and temperature changes during normal operation. The incremental leveling can be enabled by writing a non-zero value to the reg_wrlvlinc_int, reg_rdlvlgateinc_int, and reg_rdlvlinc_int fields in the Read-Write Leveling Control register. The EMIF periodically triggers incremental write leveling every time reg_wrlvlinc_int expires. In other words, the reg_wrlvlinc_int defines the interval between successive incremental write leveling. Similarly, the EMIF periodically triggers incremental read DQS gate training every time reg_rdlvlgateinc_int expires, and triggers incremental read data eye training every time reg_rdlvlinc_int expires. To minimize impact on bandwidth, the software can program these intervals such that these three intervals do not expire at same time. The value of interval programmed depends on the slope of voltage and temperature changes.

  The EMIF supports increasing the rate of incremental leveling automatically for a defined period of time. This can be achieved by a) programming the Read-Write Leveling Ramp Window register and the Read-Write Leveling Ramp Control register and b) using the inclvl_rate_chng port. Whenever a pulse is received on the inclvl_rate_chng port, the EMIF uses the intervals programmed in the Read-Write Leveling Ramp Control register until the reg_rdwrlvlinc_rmp_win in the Read-Write Leveling Ramp Window register expires. After reg_rdwrlvlinc_rmp_win expires, the EMIF switches back to use the intervals programmed in the Read-Write Leveling Control register. To ensure none of the incremental leveling events are missed, the reg_rdwrlvlinc_rmp_win must be programmed greater than the intervals in the Read-Write Leveling Ramp Control register.

If the EMIF is in the self-refresh or power-down modes when any of the incremental leveling intervals expire, the EMIF will exit the self-refresh or power-down mode, perform the required leveling, and then re-enter the self-refresh or power-down mode.

Each leveling type has three parts:

- Write leveling
- Read data eye training
- Read DQS gate training

9.3.3.10 EMIF Access Cycles

By default, the EMIF keeps SDRAM chip selects signal high. To direct a command to only one of the SDRAMs, EMIF asserts the chip select to the SDRAM for the duration of the command. If the reg_ebank field in the SDRAM Config register is set to 0, chip select 1 will always be driven high except during initialization and for refresh, power-down, self-refresh, and deep power-down commands.

The EMIF always performs burst accesses to the SDRAM. Multiple SDRAM bursts may need to service a single OCP burst request. Table 9-152 through Table 9-156 show a few examples how EMIF performs SDRAM accesses for a linear incrementing transaction type. T0, T1, and others are clock cycles. R0 is read starting at column 0, R8 is read starting at column 8, and R16 is read starting at column 16. D0-1 is the data from column 0 and 1, D2-3 is the data from column 2 and 3, and so on.

Table 9-152. 64-Byte Linear Read Starting at Address 0x0 (All DDR)

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
R0				R8							
				D0-1	D2-3	D4-5	D6-7	D8-9	D10-11	D12-13	D14-15

Table 9-153. 64-Byte Linear Read Starting at Address 0x8 (LPDDR2-S2)

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14
R2			R8				R16							
				D2-3	D4-5	D6-7	D8-9	D10-11	D12-13	D14-15	D16-17	Unused	Unused	Unused

Table 9-154. 64-Byte Linear Read Starting at Address 0x8 (LPDDR2-S4)

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15
R2				R8				R16							
				D2-3	D4-5	D6-7	Unused	D8-9	D10-11	D12-13	D14-15	D16-17	Unused	Unused	Unused

Table 9-155. 64-Byte Linear Read Starting at Address 0x10 (All DDR)

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13
R4		R8				R16							
				D4-5	D6-7	D8-9	D10-11	D12-13	D14-15	D16-17	D18-19	Unused	Unused

Table 9-156. 64-Byte Linear Read Starting at Address 0x18 (All DDR)

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13
R6		R8				R16							
				D6-7	Unused	D8-9	D10-11	D12-13	D14-15	D16-17	D18-19	D20-21	Unused

The EMIF uses the unused data phases in the preceding figures by issuing successive read commands if there are reads to open banks pending in the command FIFO.

The write data conversion from single data rate to double data rate must be done outside the EMIF. The SDRAM_DQS signal generation (with right timing) from the phy_dqs_en signal must also be done outside the EMIF.

9.3.3.11 Turnaround Time

Table 9-157 lists the turnaround time that EMIF introduces on the data bus for various back-to-back accesses. The EMIF takes advantage of the CAS latencies and packs the commands as close as possible on the control bus to introduce the following turnaround time on the data bus.

Table 9-157. Turnaround Time

Current Access	Next Access	Turnaround Time (Number of DDR Clock Cycles)
SDRAM write	SDRAM read	EMIF4D_SDRAM_TIMING_1[2:0] T_WTR + 1 + CL
SDRAM read	SDRAM write	EMIF4D_SDRAM_TIMING_1[31:29] T_RTW + 1

9.3.3.12 SDRAM Address Mapping

The LPDDR2/DDR3 memory controller views external LPDDR2/DDR3 SDRAM as one continuous block of memory. This statement is true regardless of the number of memory devices located on the chip select space. The LPDDR2/DDR3 memory controller receives LPDDR2/DDR3 memory access requests along with a 32-bit logical address from the rest of the system. In turn, LPDDR2/DDR3 memory controller uses the logical address to generate a row/page, column, and bank address for the LPDDR2/DDR3 SDRAM. The number of column, row and bank address bits used is determined by the IBANK, RSIZE and PAGESIZE fields (see Table 9-158). The LPDDR2/DDR3 memory controller uses up to 16 bits for the row/page address.

Table 9-158. IBANK, RSIZE and PAGESIZE Fields Information

Bit Field	Bit Value	Bit Description
RSIZE		Defines the number of address lines to be connected to LPDDR2/DDR3 memory device
	0	9 row bits
	1h	10 row bits
	2h	11 row bits
	3h	12 row bits
	4h	13 row bits
	5h	14 row bits
	6h	15 row bits
	7h	16 row bits
PAGESIZE		Defines the page size of each page of the external LPDDR2/DDR3 memory device
	0	256 words (requires 8 column address bits)
	1h	512 words (requires 9 column address bits)
	2h	1024 words (requires 10 column address bits)
	3h	2048 words (requires 11 column address bits)
IBANK		Defines the number of internal banks on the external LPDDR2/DDR3 memory device
	0	1 bank
	1h	2 banks
	2h	4 banks
	3h	8 banks

Table 9-158. IBANK, RSIZE and PAGESIZE Fields Information (continued)

Bit Field	Bit Value	Bit Description
EBANK		Defines the number of LPDDR2/DDR3 memory controller chip selects
	0	CS0 only
	1h	Reserved

When addressing SDRAM, if the REG_IBANK_POS field in the SDRAM Config register is set to 0, and the REG_EBANK_POS field in the SDRAM Config 2 register is also set to 0, the LPDDR2/DDR3 memory controller uses the three fields, IBANK, EBANK and PAGESIZE in the SDRAM Config register to determine the mapping from source address to SDRAM row, column, bank, and chip select. If the REG_IBANK_POS field in the SDRAM Config register is set to 1, 2, or 3, or the REG_EBANK_POS field in the SDRAM Config 2 register is set to 1, the LPDDR2/DDR3 memory controller uses the 4 fields - IBANK, EBANK, PAGESIZE, and ROWSIZE in the SDRAM Config register to determine the mapping from source address to SDRAM row, column, bank, and chip select. In all cases the LPDDR2/DDR3 memory controller considers its SDRAM address space to be a single logical block regardless of the number of physical devices or whether the devices are mapped across 1 or 2 LPDDR2/DDR3 memory controller chip selects.

9.3.3.12.1 Address Mapping when REG_IBANK_POS=0 and REG_EBANK_POS=0

For REG_IBANK_POS=0 and REG_EBANK_POS=0, the effect of address mapping scheme is that as the source address increments across LPDDR2/DDR3 memory device page boundaries, the LPDDR2/DDR3 controller moves onto the same page in the next bank in the current device DDR_CSn[0]. This movement along the banks of the current proceeds to the same page in the next device (if EBANK=1, DDR_CSn[1]) and proceeds through the same page in all its banks before moving over to the next page in the first device(DDR_CSn[0]). The LPDDR2/DDR3 controller exploits this traversal across internal banks and chip selects while remaining on the same page to maximize the number of open LPDDR2/DDR3 memory device banks within the overall LPDDR2/DDR3 memory device space.

Thus, the LPDDR2/DDR3 controller can keep a maximum of 16 banks (8 internal banks across 2 chip selects) open at a time, and can interleave among all of them.

Table 9-159. OCP Address to LPDDR2/DDR3 Address Mapping for REG_IBANK_POS=0 and REG_EBANK_POS=0

Logical Address			
Row Address	Chip Select	Bank Address	Column Address
LPDDR2 = 15 bits DDR3 = 16 bits	# of bits defined by EBANK of SDRCR	# of bits defined by IBANK of SDRCR	# of bits defined by PAGESIZE of SDRCR
	EBANK=0 => 0 bits	IBANK=0 => 0 bits	PAGESIZE=0 => 8 bits
	EBANK=1 => 1 bit	IBANK=1 => 1 bit	PAGESIZE=1 => 9 bits
		IBANK=2 => 2 bits	PAGESIZE=2 => 10 bits
		IBANK=3 => 3 bits	PAGESIZE=3 => 11 bits

9.3.3.12.2 Address Mapping when REG_IBANK_POS = 1 and REG_EBANK_POS = 0

For REG_IBANK_POS = 1 and REG_EBANK_POS = 0, the interleaving of banks within a device (per chip select) is limited to 4 banks. However, it can still interleave banks between the two chip selects. Thus, the LPDDR2/DDR3 controller can keep a maximum of 16 banks (8 internal banks across 2 chip selects) open at a time, but can only interleave among eight of them.

Table 9-160. OCP Address to LPDDR2/DDR3 Address Mapping for REG_IBANK_POS=1 and REG_EBANK_POS=0

Logical Address				
Bank Address[2]	**Row Address**	**Chip Select**	**Bank Address[1:0]**	**Column Address**
# of bits defined by IBANK of SDRCR	# of bits defined by RSIZE of SDRCR	# of bits defined by EBANK of SDRCR	# of bits defined by IBANK of SDRCR	# of bits defined by PAGESIZE of SDRCR
IBANK=0 => 0 bits	RSIZE=0 => 9 bits	EBANK=0 => 0 bits	IBANK=0 => 0 bits	PAGESIZE=0 => 8 bits
IBANK=1 => 0 bits	RSIZE=1 => 10 bits	EBANK=1 => 1 bit	IBANK=1 => 1 bit	PAGESIZE=1 => 9 bits
IBANK=2 => 0 bits	RSIZE=2 => 11 bits		IBANK=2 => 2 bits	PAGESIZE=2 => 10 bits
IBANK=3 => 1 bit	RSIZE=3 => 12 bits		IBANK=3 => 3 bits	PAGESIZE=3 => 11 bits
	RSIZE=4 => 13 bits			
	RSIZE=5 => 14 bits			
	RSIZE=6 => 15 bits			
	RSIZE=7 => 16 bits			

9.3.3.12.3 Address Mapping when REG_IBANK_POS=2 and REG_EBANK_POS = 0

For REG_IBANK_POS=2 and REG_EBANK_POS = 0, the interleaving of banks within a device (per chip select) is limited to 2 banks. However, it can still interleave banks between the two chip selects. Thus, the LPDDR2/DDR3 controller can keep a maximum of 16 banks (eight internal banks across 2 chip selects) open at a time, but can only interleave among four of them.

Table 9-161. OCP Address to LPDDR2/DDR3 Address Mapping for REG_IBANK_POS=2 and REG_EBANK_POS=0

Logical Address				
Bank Address[2:1]	**Row Address**	**Chip Select**	**Bank Address[0]**	**Column Address**
# of bits defined by IBANK of SDRCR	# of bits defined by RSIZE of SDRCR	# of bits defined by EBANK of SDRCR	# of bits defined by IBANK of SDRCR	# of bits defined by PAGESIZE of SDRCR
IBANK=0 => 0 bits	RSIZE=0 => 9 bits	EBANK=0 => 0 bits	IBANK=0 => 0 bits	PAGESIZE=0 => 8 bits
IBANK=1 => 0 bits	RSIZE=1 => 10 bits	EBANK=1 => 1 bit	IBANK=1 => 1 bit	PAGESIZE=1 => 9 bits
IBANK=2 => 1 bit	RSIZE=2 => 11 bits		IBANK=2 => 1 bit	PAGESIZE=2 => 10 bits
IBANK=3 => 2 bits	RSIZE=3 => 12 bits		IBANK=3 => 1 bit	PAGESIZE=3 => 11 bits
	RSIZE=4 => 13 bits			
	RSIZE=5 => 14 bits			
	RSIZE=6 => 15 bits			
	RSIZE=7 => 16 bits			

9.3.3.12.4 Address Mapping when REG_IBANK_POS= 3 and REG_EBANK_POS = 0

For REG_IBANK_POS= 3 and REG_EBANK_POS = 0, the LPDDR2/DDR3 controller cannot interleave banks within a device (per chip select). However, it can still interleave banks between the two chip selects. Thus, the LPDDR2/DDR3 controller can keep a maximum of 16 banks (8 internal banks across 2 chip selects) open at a time, but can only interleave among two of them.

Table 9-162. OCP Address to LPDDR2/DDR3 Address Mapping for REG_IBANK_POS=3 and REG_EBANK_POS=0

Logical Address			
Bank Address	**Row Address**	**Chip Select**	**Column Address**
# of bits defined by IBANK of SDRCR	# of bits defined by RSIZE of SDRCR	# of bits defined by EBANK of SDRCR	# of bits defined by PAGESIZE of SDRCR
IBANK=0 => 0 bits	RSIZE=0 => 9 bits	EBANK=0 => 0 bits	PAGESIZE=0 => 8 bits
IBANK=1 => 1 bit	RSIZE=1 => 10 bits	EBANK=1 => 1 bit	PAGESIZE=1 => 9 bits

Table 9-162. OCP Address to LPDDR2/DDR3 Address Mapping for REG_IBANK_POS=3 and REG_EBANK_POS=0 (continued)

Logical Address			
Bank Address	**Row Address**	**Chip Select**	**Column Address**
IBANK=2 => 2 bits	RSIZE=2 => 11 bits		PAGESIZE=2 => 10 bits
IBANK=3 => 3 bits	RSIZE=3 => 12 bits		PAGESIZE=3 => 11 bits
	RSIZE=4 => 13 bits		
	RSIZE=5 => 14 bits		
	RSIZE=6 => 15 bits		
	RSIZE=7 => 16 bits		

9.3.3.12.5 Address Mapping when REG_IBANK_POS = 0 and REG_EBANK_POS = 1

For REG_IBANK_POS = 0 and REG_EBANK_POS = 1, the LPDDR2/DDR3 memory controller interleaves among all the banks within a device (per chip select). However, the LPDDR2/DDR3 memory controller cannot interleave banks between the two chip selects. Thus, the LPDDR2/DDR3 memory controller can keep a maximum of 16 banks (8 internal banks across 2 chip selects) open at a time, but can only interleave among 8 of them.

Table 9-163. OCP Address to LPDDR2/DDR3 Address Mapping for REG_IBANK_POS=0 and REG_EBANK_POS=1

Logical Address			
Chip Select	**Row Address**	**Bank Address**	**Column Address**
# of bits defined by EBANK of SDRCR	# of bits defined by RSIZE of SDRCR	# of bits defined by IBANK of SDRCR	# of bits defined by PAGESIZE of SDRCR
EBANK=0 => 0 bits	RSIZE=0 => 9 bits	IBANK=0 => 0 bits	PAGESIZE=0 => 8 bits
EBANK=1 => 1 bit	RSIZE=1 => 10 bits	IBANK=1 => 1 bit	PAGESIZE=1 => 9 bits
	RSIZE=2 => 11 bits	IBANK=2 => 2 bits	PAGESIZE=2 => 10 bits
	RSIZE=3 => 12 bits	IBANK=3 => 3 bits	PAGESIZE=3 => 11 bits
	RSIZE=4 => 13 bits		
	RSIZE=5 => 14 bits		
	RSIZE=6 => 15 bits		
	RSIZE=7 => 16 bits		

9.3.3.12.6 Address Mapping when REG_IBANK_POS = 1 and REG_EBANK_POS = 1

For REG_IBANK_POS = 1 and REG_EBANK_POS = 1, the interleaving of banks within a device (per chip select) is limited to 4 banks. Also, the LPDDR2/DDR3 memory controller cannot interleave banks between the two chip selects. Thus, the LPDDR2/DDR3 memory controller can keep a maximum of 16 banks (8 internal banks across 2 chip selects) open at a time, but can only interleave among four of them.

Table 9-164. OCP Address to LPDDR2/DDR3 Address Mapping for REG_IBANK_POS=1 and REG_EBANK_POS = 1

Logical Address				
Chip Select	**Bank Address[2]**	**Row Address**	**Bank Address[1:0]**	**Column Address**
# of bits defined by EBANK of SDRCR	# of bits defined by IBANK of SDRCR	# of bits defined by RSIZE of SDRCR	# of bits defined by IBANK of SDRCR	# of bits defined by PAGESIZE of SDRCR
EBANK=0 => 0 bits	IBANK=0 => 0 bits	RSIZE=0 => 9 bits	IBANK=0 => 0 bits	PAGESIZE=0 => 8 bits
EBANK=1 => 1 bit	IBANK=1 => 0 bits	RSIZE=1 => 10 bits	IBANK=1 => 1 bit	PAGESIZE=1 => 9 bits
	IBANK=2 => 0 bits	RSIZE=2 => 11 bits	IBANK=2 => 2 bits	PAGESIZE=2 => 10 bits
	IBANK=3 => 1 bit	RSIZE=3 => 12 bits	IBANK=3 => 2 bits	PAGESIZE=3 => 11 bits
		RSIZE=4 => 13 bits		

Table 9-164. OCP Address to LPDDR2/DDR3 Address Mapping for REG_IBANK_POS=1 and REG_EBANK_POS = 1 (continued)

Logical Address				
Chip Select	**Bank Address[2]**	**Row Address**	**Bank Address[1:0]**	**Column Address**
		RSIZE=5 => 14 bits		
		RSIZE=6 => 15 bits		
		RSIZE=7 => 16 bits		

9.3.3.12.7 *Address Mapping when REG_IBANK_POS = 2 and REG_EBANK_POS = 1*

For REG_IBANK_POS = 2 and REG_EBANK_POS = 1, the interleaving of banks within a device (per chip select) is limited to 2 banks. Also, the LPDDR2/DDR3 memory controller cannot interleave banks between the two chip selects. Thus, the LPDDR2/DDR3 memory controller can keep a maximum of 16 banks (8 internal banks across 2 chip selects) open at a time, but can only interleave among two of them.

Table 9-165. OCP Address to LPDDR2/DDR3 Address Mapping for REG_IBANK_POS=2 and REG_EBANK_POS = 1

Logical Address				
Chip Select	**Bank Address[2:1]**	**Row Address**	**Bank Address[0]**	**Column Address**
# of bits defined by EBANK of SDRCR	# of bits defined by IBANK of SDRCR	# of bits defined by RSIZE of SDRCR	# of bits defined by IBANK of SDRCR	# of bits defined by PAGESIZE of SDRCR
EBANK=0 => 0 bits	IBANK=0 => 0 bits	RSIZE=0 => 9 bits	IBANK=0 => 0 bits	PAGESIZE=0 => 8 bits
EBANK=1 => 1 bit	IBANK=1 => 0 bits	RSIZE=1 => 10 bits	IBANK=1 => 1 bit	PAGESIZE=1 => 9 bits
	IBANK=2 => 1 bit	RSIZE=2 => 11 bits	IBANK=2 => 1 bit	PAGESIZE=2 => 10 bits
	IBANK=3 => 2 bits	RSIZE=3 => 12 bits	IBANK=3 => 1 bit	PAGESIZE=3 => 11 bits
		RSIZE=4 => 13 bits		
		RSIZE=5 => 14 bits		
		RSIZE=6 => 15 bits		
		RSIZE=7 => 16 bits		

9.3.3.12.8 *Address Mapping when REG_IBANK_POS = 3 and REG_EBANK_POS = 1*

For REG_IBANK_POS = 3 and REG_EBANK_POS = 1, the LPDDR2/DDR3 memory controller cannot interleave banks within a device (per chip select) or between the two chip selects. Thus, the LPDDR2/DDR3 memory controller can keep a maximum of 16 banks (8 internal banks across two chip selects) open at a time, but cannot interleave among of them.

Table 9-166. OCP Address to LPDDR2/DDR3 Address Mapping for REG_IBANK_POS=3 and REG_EBANK_POS=1

Logical Address			
Chip Select	**Bank Address**	**Row Address**	**Column Address**
# of bits defined by EBANK of SDRCR	# of bits defined by IBANK of SDRCR	# of bits defined by RSIZE of SDRCR	# of bits defined by PAGESIZE of SDRCR
EBANK=0 => 0 bits	IBANK=0 => 0 bits	RSIZE=0 => 9 bits	PAGESIZE=0 => 8 bits
EBANK=1 => 1 bit	IBANK=1 => 1 bit	RSIZE=1 => 10 bits	PAGESIZE=1 => 9 bits
	IBANK=2 => 2 bits	RSIZE=2 => 11 bits	PAGESIZE=2 => 10 bits
	IBANK=3 => 3 bits	RSIZE=3 => 12 bits	PAGESIZE=3 => 11 bits
		RSIZE=4 => 13 bits	
		RSIZE=5 => 14 bits	
		RSIZE=6 => 15 bits	
		RSIZE=7 => 16 bits	

Since the LPDDR2/DDR3 memory controller interleaves among less number of banks when IBANK_POS!= 0 or EBANK_POS= 1, these cases are lower in performance than the IBANK_POS= 0 case. Thus these cases are only recommended to be used along with partial array self-refresh where performance can be traded off for power savings.

9.3.3.13 PHY DLL Calibration

When running in normal locked mode, the PHY DLL gets a reference clock (EMIFi_DLL_FCLK) from the PRCM, which is used by the DLL master to lock to the right frequency and provide the control code for a full period phase shift to the slave. The slave uses this code as a control for its internal delay line to produce the required delay for the signal considered.

When working in locked mode, the delay lines only get an updated control value from the master DLLs when an explicit dll_calib command is issued by the EMIF controller. Failure to send such commands on a timely basis will result in inaccurate delay-line information if there is a significant voltage and temperature drift in the system. It is recommended to issue at least one command every 100 µs. EMIF automatically sends ctrl_update commands for:

- Refresh Exit
- Self Refresh Exit
- phy_ready asserted during initialization

The PHY also internally generates a control value update upon completion of a leveling operation. Control is also added when leveling is not used and there are gradual voltage changes during frequency change. The EMIF4D_DLL_CALIB_CTRL register can be programmed to generate a phy_dll_calib for a periodic interval based on EMIF_FICLK cycles, so allow continued memory access as voltage is changing. A safe window of no activity will be guaranteed for this periodic generation of phy_dll_calib. In addition, a one shot generator for phy_pll_calib has also been added that will generate a single phy_dll_calib by setting the EMIF4D_MISC_REG[0] DLL_CALIB_OS bit to 1.

9.3.3.14 DDR3 Output Impedance Calibration

The EMIF controller supports automatic output impedance (ZQ) calibration for LPDDR2 and DDR3 memories. The ZQ calibration can be enabled by setting to 0x1 the EMIF4D_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[30] ZQ_CS0EN bit. The EMIF supports three types of ZQ calibration commands:

- ZQINIT: ZQ calibration command during initialization
- ZQCL: ZQ calibration long command
- ZQCS: ZQ calibration short command

For DDR3, the EMIF will automatically issue a ZQINIT command during initialization. For LPDDR2, it is the software’s responsibility for issuing a ZQINIT command (MRW to Mode Register 10). The EMIF will wait and block any other command for (reg_zq_zqinit_mult+1)*(req_zq_zqcl_mult+1)*(reg_zq_zqcs+1) number of clock DDR clock cycles every time a ZQINIT command is issued.

The EMIF periodically issues a ZQCS command every time reg_zq_refinterval expires. In other words, the reg_zq_refinterval defines the interval between ZQCS commands. The EMIF will wait and block any other command for (reg_zq_zqcs+1) number of DDR clock cycles every time a ZQCS command is issued.

If the reg_zq_sfexiten field is set to a 1, the EMIF will issue a ZQCL command every time it exits self-refresh, active power-down, and pre-charge power-down mode. The EMIF will wait and block any other command for (req_zq_zqcl_mult+1)*(reg_zq_zqcs+1) number of DDR clock cycles every time a ZQCL command is issued.

If a separate calibration resistor is used per device, the ZQ calibration can be performed simultaneously over both chip selects. To enable ZQ calibration to be performed simultaneously over both chip selects, the reg_zq_dualcalen field must be set to a 1. If reg_zq_dualcalen is set to a 0, the EMIF will serially perform ZQ calibration per chip select.

9.3.3.15 Temperature Monitoring

The EMIF supports automatic temperature monitoring for LPDDR2 to facilitate the software to update the refresh rate according to the LPDDR2 device temperature changes. The temperature monitoring can be enabled per chip select by setting reg_ta_cs0en and reg_ta_cs1en fields in the SDRAM Temperature Alert Config register.

The EMIF periodically polls the temperature of LPDDR2 (by issuing an MRR command to Mode Register 4) every time reg_ta_refinterval expires. In other words, the reg_ta_refinterval defines the interval between temperature alert polls. If the EMIF sees a 1 on bit 7 of the read data value from MRR, indicating that the temperature has changed, it sends an interrupt both on the system and low-latency interrupt lines. After receiving the interrupt, the software will update the reg_refresh_rate field in the SDRAM Refresh Control register to the required value as per the temperature change.

If reg_ta_sfexiten field is set to a 1, the EMIF will poll for temperature change every time it exits self-refresh, active power-down, and pre-charge power-down modes.

Since the EMIF is performing a MRR, it needs information on how the LPDDR2 are connected. The reg_ta_devwdt and reg_ta_devcnt fields in the Temperature Alert Config register provides the necessary information to the EMIF for MRR data compare. For example, if reg_ta_devwdt is set to is set to 0 indicating 8-bit devices used, and if reg_ta_devcnt is set to 2 indicating four devices used to form a 32-bit bus, the mask used for checking would be 4’b1111, that is, the EMIF would expect data on each lane of the bytes on a 32-bit DDR bus.

9.3.3.16 Class of Service

The commands in the Command FIFO can be mapped to two classes of service namely 1 and 2. The mapping of commands to a particular class of service can be done based on the priority or the connection ID. The mapping based on priority can be done by setting the appropriate values in the EMIF4D_PRIORITY_TO_CLASS_OF_SERVICE_MAPPING register. The mapping based on connection ID can be done by setting the appropriate values of connection ID and the masks in the EMIF4D_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING and EMIF4D_CONNECTION_ID_TO_CLASS_OF_SERVICE_2_MAPPING registers. There are 3 connection ID and mask values that can be set for each class of service. In conjunction with the masks, each class of service can have a maximum of 144 connection IDs mapped to it. For example, a connection ID value of 0xFF along with a mask value of 0x3 will map all connection IDs from 0xF8 to 0xFF to that particular class of service.

Each class of service has an associated latency counter (EMIF4D_COS_CONFIG[23:16] COS_COUNT_1 and EMIF4D_COS_CONFIG[15:8] COS_COUNT_2). When the latency counter for a command expires, that is, reaches the value programmed for the class of service that the command belongs to, that command is executed next. If there is more than one command that has expired latency counter, the command with the highest priority is executed first. One exception to this rule is, if the EMIF4D_COS_CONFIG[7:0] PR_OLD_COUNT value expires for the oldest command in the queue. That command is executed first irrespective of priority or class of service. This is done to prevent the continuous blocking effect.

The EMIF4D_COS_CONFIG[7:0] PR_OLD_COUNT value is used to identify when the oldest command in the command FIFO has timed out. At this point during the arbitration process, this oldest command is issued regardless of the priority of the other commands in the FIFO. This feature is disabled when writing 0x0 to the EMIF4D_COS_CONFIG[7:0] PR_OLD_COUNT bit field. After issuing the oldest command, the other remaining commands in the FIFO are reordered by age. The next oldest command in the FIFO is given highest priority again and issued after the EMIF4D_COS_CONFIG[7:0] PR_OLD_COUNT value expires. If a new value in the PR_OLD_COUNT bit field is written during counting, that is, before PR_OLD_COUNT expires, the counter keeps working but if this value is smaller the oldest command is issued sooner and if this value is larger the oldest command is issued later.

The connection ID mapping allows the same connection ID to be put in both class of service 1 and 2. Also, a transaction might belong to one class of service if viewed by connection ID and might belong to another class of service if viewed by priority. In these cases, the command will belong to both class of service. The EMIF will try executing the command as soon as possible, when the smaller of the two counters (EMIF4D_COS_CONFIG[23:16] COS_COUNT_1 and EMIF4D_COS_CONFIG[15:8] COS_COUNT_2) expires.

9.3.3.17 Performance Counters

The EMIF4D_PERFORMANCE_CTR_1 and EMIF4D_PERFORMANCE_CTR_2 registers are used to monitor or calculate the EMIF Controller bandwidth and efficiency. These counters are able to count events such as total SDRAM accesses, SDRAM activates, reads, writes, and other events. Each counter counts independent of the other. In addition to the ability of events counting, the counters can also filter the events from a particular master or address space. The events counting and filter enabling are configured using the EMIF4D_PERFORMANCE_CTR_CONFIG register. The filter value used is configured through the EMIF4D_PERFORMANCE_CTR_MASTER_REGION_SELECT register. Each counter can be configured independently.

Table 9-167 lists all the events that can be counted and whether a filter can be applied to a particular event. A filter is applied to an event if the following bits are set to 0x1 for that event:

- For Performance Counter 1: EMIF4D_PERFORMANCE_CTR_CONFIG[15] CNTR1_MCONNID_EN and EMIF4D_PERFORMANCE_CTR_CONFIG[14] CNTR1_REGION_EN;
- For Performance Counter 2: EMIF4D_PERFORMANCE_CTR_CONFIG[31] CNTR2_MCONNID_EN and EMIF4D_PERFORMANCE_CTR_CONFIG[30] CNTR2_REGION_EN.

Table 9-167. Performance Counter Filter Configuration

CNTRn_CFG[1]	CNTRn_REGION_EN	CNTRn_MCONNID_EN	Description
0x0	0x0	0x0 or 0x1	Count total SDRAM accesses
0x1	0x0	0x0 or 0x1	Count total SDRAM activates
0x2	0x0 or 0x1	0x0 or 0x1	Count total reads
0x3	0x0 or 0x1	0x0 or 0x1	Count total writes
0x4	0x0	0x0	Count number of EMIF_FICLK clock cycles during which the local Command FIFO is full
0x5	0x0	0x0	Count number of EMIF_FICLK clock cycles during which the local Write Data FIFO is full
0x6	0x0	0x0	Count number of EMIF_FICLK clock cycles during which the local Read Data FIFO is full
0x7	0x0	0x0	Count number of EMIF_FICLK clock cycles during which the local Return Command FIFO is full
0x8	0x0 or 0x1	0x0 or 0x1	Count number of priority elevations
0x9	0x0	0x0	Count number of EMIF_FICLK clock cycles that a command was pending
0xA	0x0	0x0	Count number of EMIF_FICLK cycles used by the EMIF controller for reads and writes.
0xB - 0xF	0x0	0x0	Reserved for future use.

[1] n = 1 or 2

NOTE: When the MReqDebug qualifier is set to 0x1 for a particular local command, the performance counters are not incremented for that particular command if the CNTRn_CONFIG values are equal to 0x0, 0x1, 0x2, 0x3, or 0xA.

9.3.3.17.1 Performance Counters General Examples

- **General Example for Counting All Write Accesses**
  If the EMIF4D_PERFORMANCE_CTR_1 register is used to count all write accesses from master with connection ID equal to 0xA (this is the system MMU) the following steps should be performed:
  - To enable the writes counting, the EMIF4D_PERFORMANCE_CTR_CONFIG[3:0] CNTR1_CFG bit field must be set to 0x3.
  - The EMIF4D_PERFORMANCE_CTR_MASTER_REGION_SELECT[15:8] MCONNID1 bit field must be set to 0xA.
  - To enable filtering, the EMIF4D_PERFORMANCE_CTR_CONFIG[15] CNTR1_MCONNID_EN bit must be set to 0x1.

With this configuration EMIF4D_PERFORMANCE_CTR_1 counts every write made to the EMIF from master 0xA to any address space. This does not include accesses from other masters and commands other than writes.

- **General Example for Counting Total Access**
  If the EMIF4D_PERFORMANCE_CTR_2 register is used to count total accesses to the SDRAM regardless of the address space or master the following steps should be performed:
  - To enable counting of all accesses to the SDRAM, the EMIF4D_PERFORMANCE_CTR_CONFIG[19:16] CNTR2_CFG bit field must be set to 0x0.
  - To disable filtering, both the EMIF4D_PERFORMANCE_CTR_CONFIG[31] CNTR2_MCONNID_EN and EMIF4D_PERFORMANCE_CTR_CONFIG[30] CNTR2_REGION_EN bits must be set to 0x0.

    With this configuration EMIF4D_PERFORMANCE_CTR_2 counts every access made to the SDRAM. This includes all accesses from all masters and to any address space.
- **General Example for Counting All Read Accesses**
  If the EMIF4D_PERFORMANCE_CTR_1 register is used to count all read accesses from master with connection ID equal to 0xA (this is the system MMU) to address space 0x0 the following steps should be performed:
  - To enable the reads counting, the EMIF4D_PERFORMANCE_CTR_CONFIG[3:0] CNTR1_CFG bit field must be set to 0x2.
  - The EMIF4D_PERFORMANCE_CTR_MASTER_REGION_SELECT[15:8] MCONNID1bit field must be set to 0xA
  - The EMIF4D_PERFORMANCE_CTR_MASTER_REGION_SELECT[1:0] REGION_SEL1 bit field must be set to 0x0.
  - To enable filtering, both the EMIF4D_PERFORMANCE_CTR_CONFIG[15] CNTR1_MCONNID_EN and the EMIF4D_PERFORMANCE_CTR_CONFIG[14] CNTR1_REGION_EN bits must be set to 0x1.

    With this configuration, EMIF4D_PERFORMANCE_CTR_1 counts every read made to the EMIF from master 0xA to address space 0x0. This does not include accesses from other masters or to other address spaces and does not include commands other than reads.

9.3.4 EMIF Registers

Table 9-168 lists the memory-mapped registers for the EMIF. All register offset addresses not listed in Table 9-168 should be considered as reserved locations and the register contents should not be modified.

Table 9-168. EMIF Registers

Offset	Acronym	Register Name	Section
0h	EMIF4D_MOD_ID_REV		Section 9.3.4.1
4h	EMIF4D_STS		Section 9.3.4.2
8h	EMIF4D_SDRAM_CONFIG		Section 9.3.4.3
Ch	EMIF4D_SDRAM_CONFIG_2		Section 9.3.4.4
10h	EMIF4D_SDRAM_REFRESH_CTRL		Section 9.3.4.5
14h	EMIF4D_SDRAM_REFRESH_CTRL_SHADOW		Section 9.3.4.6
18h	EMIF4D_SDRAM_TIMING_1		Section 9.3.4.7
1Ch	EMIF4D_SDRAM_TIMING_1_SHADOW		Section 9.3.4.8
20h	EMIF4D_SDRAM_TIMING_2		Section 9.3.4.9
24h	EMIF4D_SDRAM_TIMING_2_SHADOW		Section 9.3.4.10
28h	EMIF4D_SDRAM_TIMING_3		Section 9.3.4.11
2Ch	EMIF4D_SDRAM_TIMING_3_SHADOW		Section 9.3.4.12
30h	EMIF4D_LPDDR2_NVM_TIMING		Section 9.3.4.13
34h	EMIF4D_LPDDR2_NVM_TIMING_SHADOW		Section 9.3.4.14
38h	EMIF4D_POWER_MANAGEMENT_CTRL		Section 9.3.4.15

Table 9-168. EMIF Registers (continued)

Offset	Acronym	Register Name	Section
3Ch	EMIF4D_POWER_MANAGEMENT_CTRL_SHADOW		Section 9.3.4.16
40h	EMIF4D_LPDDR2_MODE_REG_DATA		Section 9.3.4.17
50h	EMIF4D_LPDDR2_MODE_REG_CONFIG		Section 9.3.4.18
54h	EMIF4D_OCP_CONFIG		Section 9.3.4.19
58h	EMIF4D_OCP_CONFIG_VALUE_1		Section 9.3.4.20
5Ch	EMIF4D_OCP_CONFIG_VALUE_2		Section 9.3.4.21
60h	EMIF4D_IODFT_TEST_LOGIC_GLOBAL_CTRL		Section 9.3.4.22
64h	EMIF4D_IODFT_TEST_LOGIC_CTRL_MISR_RESULT		Section 9.3.4.23
68h	EMIF4D_IODFT_TEST_LOGIC_ADDR_MISR_RESULT		Section 9.3.4.24
6Ch	EMIF4D_IODFT_TEST_LOGIC_DATA_MISR_RESULT_1		Section 9.3.4.25
70h	EMIF4D_IODFT_TEST_LOGIC_DATA_MISR_RESULT_2		Section 9.3.4.26
74h	EMIF4D_IODFT_TEST_LOGIC_DATA_MISR_RESULT_3		Section 9.3.4.27
80h	EMIF4D_PERFORMANCE_CTR_1		Section 9.3.4.28
84h	EMIF4D_PERFORMANCE_CTR_2		Section 9.3.4.29
88h	EMIF4D_PERFORMANCE_CTR_CONFIG		Section 9.3.4.30
8Ch	EMIF4D_PERFORMANCE_CTR_MASTER_REGION_SELECT		Section 9.3.4.31
90h	EMIF4D_PERFORMANCE_CTR_TIME		Section 9.3.4.32
94h	EMIF4D_MISC_REG		Section 9.3.4.33
98h	EMIF4D_DLL_CALIB_CTRL		Section 9.3.4.34
9Ch	EMIF4D_DLL_CALIB_CTRL_SHADOW		Section 9.3.4.35
A0h	EMIF4D_END_OF_INTR		Section 9.3.4.36
A4h	EMIF4D_SYSTEM_OCP_INTR_RAW_STS		Section 9.3.4.37
A8h	EMIF4D_LOW_LAT_OCP_INTR_RAW_STS		Section 9.3.4.38
ACh	EMIF4D_SYSTEM_OCP_INTR_STS		Section 9.3.4.39
B0h	EMIF4D_LOW_LAT_OCP_INTR_STS		Section 9.3.4.40
B4h	EMIF4D_SYSTEM_OCP_INTR_EN_SET		Section 9.3.4.41
B8h	EMIF4D_LOW_LAT_OCP_INTR_EN_SET		Section 9.3.4.42
BCh	EMIF4D_SYSTEM_OCP_INTR_EN_CLR		Section 9.3.4.43
C0h	EMIF4D_LOW_LAT_OCP_INTR_EN_CLR		Section 9.3.4.44
C8h	EMIF4D_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG		Section 9.3.4.45
CCh	EMIF4D_TEMPERATURE_ALERT_CONFIG		Section 9.3.4.46
D0h	EMIF4D_OCP_ERROR_LOG		Section 9.3.4.47
D4h	EMIF4D_READ_WRITE_LEVELING_RAMP_WINDOW		Section 9.3.4.48
D8h	EMIF4D_READ_WRITE_LEVELING_RAMP_CTRL		Section 9.3.4.49
DCh	EMIF4D_READ_WRITE_LEVELING_CTRL		Section 9.3.4.50
E4h	EMIF4D_DDR_PHY_CTRL_1		Section 9.3.4.51
E8h	EMIF4D_DDR_PHY_CTRL_1_SHADOW		Section 9.3.4.52

Table 9-168. EMIF Registers (continued)

Offset	Acronym	Register Name	Section
100h	EMIF4D_PRIORITY_TO_CLASS_OF_SERVICE_MAPPING		Section 9.3.4.53
104h	EMIF4D_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING		Section 9.3.4.54
108h	EMIF4D_CONNECTION_ID_TO_CLASS_OF_SERVICE_2_MAPPING		Section 9.3.4.55
110h	EMIF4D_ECC_CTRL_REG		Section 9.3.4.56
114h	EMIF4D_ECC_ADDR_RANGE_1		Section 9.3.4.57
118h	EMIF4D_ECC_ADDR_RANGE_2		Section 9.3.4.58
120h	EMIF4D_READ_WRITE_EXECUTION_THR		Section 9.3.4.59
124h	EMIF4D_COS_CONFIG		Section 9.3.4.60
130h	EMIF4D_1B_ECC_ERR_CNT		Section 9.3.4.61
134h	EMIF4D_1B_ECC_ERR_THRSH		Section 9.3.4.62
138h	EMIF4D_1B_ECC_ERR_DIST_1		Section 9.3.4.63
13Ch	EMIF4D_1B_ECC_ERR_ADDR_LOG		Section 9.3.4.64
140h	EMIF4D_2B_ECC_ERR_ADDR_LOG		Section 9.3.4.65
144h	EMIF4D_PHY_STS_1		Section 9.3.4.66
148h	EMIF4D_PHY_STS_2		Section 9.3.4.67
14Ch	EMIF4D_PHY_STS_3		Section 9.3.4.68
150h	EMIF4D_PHY_STS_4		Section 9.3.4.69
154h	EMIF4D_PHY_STS_5		Section 9.3.4.70
158h	EMIF4D_PHY_STS_6		Section 9.3.4.71
15Ch	EMIF4D_PHY_STS_7		Section 9.3.4.72
160h	EMIF4D_PHY_STS_8		Section 9.3.4.73
164h	EMIF4D_PHY_STS_9		Section 9.3.4.74
168h	EMIF4D_PHY_STS_10		Section 9.3.4.75
16Ch	EMIF4D_PHY_STS_11		Section 9.3.4.76
170h	EMIF4D_PHY_STS_12		Section 9.3.4.77
174h	EMIF4D_PHY_STS_13		Section 9.3.4.78
178h	EMIF4D_PHY_STS_14		Section 9.3.4.79
17Ch	EMIF4D_PHY_STS_15		Section 9.3.4.80
180h	EMIF4D_PHY_STS_16		Section 9.3.4.81
184h	EMIF4D_PHY_STS_17		Section 9.3.4.82
188h	EMIF4D_PHY_STS_18		Section 9.3.4.83
18Ch	EMIF4D_PHY_STS_19		Section 9.3.4.84
190h	EMIF4D_PHY_STS_20		Section 9.3.4.85
194h	EMIF4D_PHY_STS_21		Section 9.3.4.86
198h	EMIF4D_PHY_STS_22		Section 9.3.4.87
19Ch	EMIF4D_PHY_STS_23		Section 9.3.4.88
1A0h	EMIF4D_PHY_STS_24		Section 9.3.4.89
1A4h	EMIF4D_PHY_STS_25		Section 9.3.4.90
1A8h	EMIF4D_PHY_STS_26		Section 9.3.4.91
1ACh	EMIF4D_PHY_STS_27		Section 9.3.4.92
1B0h	EMIF4D_PHY_STS_28		Section 9.3.4.93
200h	EMIF4D_EXT_PHY_CTRL_1		Section 9.3.4.94
204h	EMIF4D_EXT_PHY_CTRL_1_SHADOW		Section 9.3.4.95
208h	EMIF4D_EXT_PHY_CTRL_2		Section 9.3.4.96
20Ch	EMIF4D_EXT_PHY_CTRL_2_SHADOW		Section 9.3.4.97

Table 9-168. EMIF Registers (continued)

Offset	Acronym	Register Name	Section
210h	EMIF4D_EXT_PHY_CTRL_3		Section 9.3.4.98
214h	EMIF4D_EXT_PHY_CTRL_3_SHADOW		Section 9.3.4.99
218h	EMIF4D_EXT_PHY_CTRL_4		Section 9.3.4.100
21Ch	EMIF4D_EXT_PHY_CTRL_4_SHADOW		Section 9.3.4.101
220h	EMIF4D_EXT_PHY_CTRL_5		Section 9.3.4.102
224h	EMIF4D_EXT_PHY_CTRL_5_SHADOW		Section 9.3.4.103
228h	EMIF4D_EXT_PHY_CTRL_6		Section 9.3.4.104
22Ch	EMIF4D_EXT_PHY_CTRL_6_SHADOW		Section 9.3.4.105
230h	EMIF4D_EXT_PHY_CTRL_7		Section 9.3.4.106
234h	EMIF4D_EXT_PHY_CTRL_7_SHADOW		Section 9.3.4.107
238h	EMIF4D_EXT_PHY_CTRL_8		Section 9.3.4.108
23Ch	EMIF4D_EXT_PHY_CTRL_8_SHADOW		Section 9.3.4.109
240h	EMIF4D_EXT_PHY_CTRL_9		Section 9.3.4.110
244h	EMIF4D_EXT_PHY_CTRL_9_SHADOW		Section 9.3.4.111
248h	EMIF4D_EXT_PHY_CTRL_10		Section 9.3.4.112
24Ch	EMIF4D_EXT_PHY_CTRL_10_SHADOW		Section 9.3.4.113
250h	EMIF4D_EXT_PHY_CTRL_11		Section 9.3.4.114
254h	EMIF4D_EXT_PHY_CTRL_11_SHADOW		Section 9.3.4.115
258h	EMIF4D_EXT_PHY_CTRL_12		Section 9.3.4.116
25Ch	EMIF4D_EXT_PHY_CTRL_12_SHADOW		Section 9.3.4.117
260h	EMIF4D_EXT_PHY_CTRL_13		Section 9.3.4.118
264h	EMIF4D_EXT_PHY_CTRL_13_SHADOW		Section 9.3.4.119
268h	EMIF4D_EXT_PHY_CTRL_14		Section 9.3.4.120
26Ch	EMIF4D_EXT_PHY_CTRL_14_SHADOW		Section 9.3.4.121
270h	EMIF4D_EXT_PHY_CTRL_15		Section 9.3.4.122
274h	EMIF4D_EXT_PHY_CTRL_15_SHADOW		Section 9.3.4.123
278h	EMIF4D_EXT_PHY_CTRL_16		Section 9.3.4.124
27Ch	EMIF4D_EXT_PHY_CTRL_16_SHADOW		Section 9.3.4.125
280h	EMIF4D_EXT_PHY_CTRL_17		Section 9.3.4.126
284h	EMIF4D_EXT_PHY_CTRL_17_SHADOW		Section 9.3.4.127
288h	EMIF4D_EXT_PHY_CTRL_18		Section 9.3.4.128
28Ch	EMIF4D_EXT_PHY_CTRL_18_SHADOW		Section 9.3.4.129
290h	EMIF4D_EXT_PHY_CTRL_19		Section 9.3.4.130
294h	EMIF4D_EXT_PHY_CTRL_19_SHADOW		Section 9.3.4.131
298h	EMIF4D_EXT_PHY_CTRL_20		Section 9.3.4.132
29Ch	EMIF4D_EXT_PHY_CTRL_20_SHADOW		Section 9.3.4.133
2A0h	EMIF4D_EXT_PHY_CTRL_21		Section 9.3.4.134
2A4h	EMIF4D_EXT_PHY_CTRL_21_SHADOW		Section 9.3.4.135
2A8h	EMIF4D_EXT_PHY_CTRL_22		Section 9.3.4.136
2ACh	EMIF4D_EXT_PHY_CTRL_22_SHADOW		Section 9.3.4.137
2B0h	EMIF4D_EXT_PHY_CTRL_23		Section 9.3.4.138
2B4h	EMIF4D_EXT_PHY_CTRL_23_SHADOW		Section 9.3.4.139
2B8h	EMIF4D_EXT_PHY_CTRL_24		Section 9.3.4.140
2BCh	EMIF4D_EXT_PHY_CTRL_24_SHADOW		Section 9.3.4.141
2C0h	EMIF4D_EXT_PHY_CTRL_25		Section 9.3.4.142
2C4h	EMIF4D_EXT_PHY_CTRL_25_SHADOW		Section 9.3.4.143
2C8h	EMIF4D_EXT_PHY_CTRL_26		Section 9.3.4.144

Table 9-168. EMIF Registers (continued)

Offset	Acronym	Register Name	Section
2CCh	EMIF4D_EXT_PHY_CTRL_26_SHADOW		Section 9.3.4.145
2D0h	EMIF4D_EXT_PHY_CTRL_27		Section 9.3.4.146
2D4h	EMIF4D_EXT_PHY_CTRL_27_SHADOW		Section 9.3.4.147
2D8h	EMIF4D_EXT_PHY_CTRL_28		Section 9.3.4.148
2DCh	EMIF4D_EXT_PHY_CTRL_28_SHADOW		Section 9.3.4.149
2E0h	EMIF4D_EXT_PHY_CTRL_29		Section 9.3.4.150
2E4h	EMIF4D_EXT_PHY_CTRL_29_SHADOW		Section 9.3.4.151
2E8h	EMIF4D_EXT_PHY_CTRL_30		Section 9.3.4.152
2ECh	EMIF4D_EXT_PHY_CTRL_30_SHADOW		Section 9.3.4.153
2F0h	EMIF4D_EXT_PHY_CTRL_31		Section 9.3.4.154
2F4h	EMIF4D_EXT_PHY_CTRL_31_SHADOW		Section 9.3.4.155
2F8h	EMIF4D_EXT_PHY_CTRL_32		Section 9.3.4.156
2FCh	EMIF4D_EXT_PHY_CTRL_32_SHADOW		Section 9.3.4.157
300h	EMIF4D_EXT_PHY_CTRL_33		Section 9.3.4.158
304h	EMIF4D_EXT_PHY_CTRL_33_SHADOW		Section 9.3.4.159
308h	EMIF4D_EXT_PHY_CTRL_34		Section 9.3.4.160
30Ch	EMIF4D_EXT_PHY_CTRL_34_SHADOW		Section 9.3.4.161
310h	EMIF4D_EXT_PHY_CTRL_35		Section 9.3.4.162
314h	EMIF4D_EXT_PHY_CTRL_35_SHADOW		Section 9.3.4.163
318h	EMIF4D_EXT_PHY_CTRL_36		Section 9.3.4.164
31Ch	EMIF4D_EXT_PHY_CTRL_36_SHADOW		Section 9.3.4.165

9.3.4.1 EMIF4D_MOD_ID_REV Register (Offset = 0h) [reset = 50440500h]

EMIF4D_MOD_ID_REV is shown in Figure 9-143 and described in Table 9-169.

Return to Summary Table.

Figure 9-143. EMIF4D_MOD_ID_REV Register

31	30	29	28	27	26	25	24
SCHEME		BU		MODULE_ID			
R-1h		R-1h		R-44h			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
MODULE_ID							
R-44h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RTL_VERSION					MAJOR_REVISION		
R-0h					R-5h		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED		MINOR_REVISION					
R-0-0h		R-0h					

Table 9-169. EMIF4D_MOD_ID_REV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	Used to distinguish between old and current schemes.
29-28	BU	R	1h	Business Unit.
27-16	MODULE_ID	R	44h	EMIF module ID.
15-11	RTL_VERSION	R	0h	RTL Version.
10-8	MAJOR_REVISION	R	5h	Major Revision.
7-6	RESERVED	R-0	0h	
5-0	MINOR_REVISION	R	0h	Minor Revision.

9.3.4.2 EMIF4D_STS Register (Offset = 4h) [reset = 0h]

EMIF4D_STS is shown in Figure 9-144 and described in Table 9-170.

Return to Summary Table.

Figure 9-144. EMIF4D_STS Register

31	30	29	28	27	26	25	24
BE	DUAL_CLK_MODE	FAST_INIT	RESERVED				
R-0h	R-0h	R-0h	R-0-0h				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	RDLVLGATETO	RDLVLTO	WRLVLTO	RESERVED	PHY_DLL_READY	RESERVED	
R-0-0h	R-0h	R-0h	R-0h	R-0-0h	R-0h	R-0-0h	

Table 9-170. EMIF4D_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	BE	R	0h	Big Endian. Reflects the value on the config_big_endian port that defines whether the EMIF is in big or little endian mode.
30	DUAL_CLK_MODE	R	0h	Dual Clock mode. Reflects the value on the config_dual_clk_mode port that defines whether the ocp_clk and m_clk are asynchronous.
29	FAST_INIT	R	0h	Fast Init. Reflects the value on the config_fast_init port that defines whether the EMIF fast initialization mode has been enabled.
28-7	RESERVED	R-0	0h	
6	RDLVLGATETO	R	0h	Read DQS Gate Training Timeout. Value of 1 indicates read DQS gate training has timed out because read DQS gate training done was not received from the PHY.
5	RDLVLTO	R	0h	Read Data Eye Training Timeout. Value of 1 indicates read data eye training has timed out because read data eye training done was not received from the PHY.
4	WRLVLTO	R	0h	Write Leveling Timeout. Value of 1 indicates write leveling has timed out because write leveling done was not received from the PHY.
3	RESERVED	R-0	0h	
2	PHY_DLL_READY	R	0h	DDR PHY Ready. Reflects the value on the phy_ready port (active high) that defines whether the DDR PHY is ready for normal operation.
1-0	RESERVED	R-0	0h	

9.3.4.3 EMIF4D_SDRAM_CONFIG Register (Offset = 8h) [reset = 0h]

EMIF4D_SDRAM_CONFIG is shown in Figure 9-145 and described in Table 9-171.

Return to Summary Table.

Figure 9-145. EMIF4D_SDRAM_CONFIG Register

31	30	29	28	27	26	25	24
SDRAM_TYPE			IBANK_POS		DDR_TERM		
R/W-0h			R/W-0h		R/W-0h		

23	22	21	20	19	18	17	16
LPDDR2_DDQS	DYN_ODT		DDR_DISABLE_DLL	SDRAM_DRIVE		CWL	
R/W-0h	R/W-0h		R/W-0h	R/W-0h		R/W-0h	

15	14	13	12	11	10	9	8
NARROW_MODE		CL				ROWSIZE	
R/W-0h		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
ROWSIZE	IBANK			EBANK	PAGESIZE		
R/W-0h	R/W-0h			R/W-0h	R/W-0h		

Table 9-171. EMIF4D_SDRAM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	SDRAM_TYPE	R/W	0h	SDRAM Type selection. Set to 3 for DDR3. Set to 4 for LPDDR2. All other values are reserved.
28-27	IBANK_POS	R/W	0h	Internal bank position. Set to 0 to assign internal bank address bits from lower OCP address bits as shown in tables. Set to 1, 2, or 3 to assign internal bank address bits from higher OCP address bits as shown in tables.
26-24	DDR_TERM	R/W	0h	DDR3 termination resistor value. Set to 0 to disable termination. Set to 1 for RZQ/4. Set to 2 for RZQ/2. Set to 3 for RZQ/6. Set to 4 for RZQ/12. Set to 5 for RZQ/8. All other values are reserved.
23	LPDDR2_DDQS	R/W	0h	LPDDR2 differential DQS enable. Set to 0 for single ended DQS. Set to 1 for differential DQS.
22-21	DYN_ODT	R/W	0h	DDR3 Dynamic ODT. Set to 0 to turn off dynamic ODT. Set to 1 for RZQ/4 and set to 2 for RZQ/2. All other values are reserved.
20	DDR_DISABLE_DLL	R/W	0h	Disable DLL select. Set to 1 to disable DLL inside SDRAM.
19-18	SDRAM_DRIVE	R/W	0h	SDRAM drive strength. For DDR3, set to 0 for RZQ/6 and set to 1 for RZQ/7. All other values are reserved.
17-16	CWL	R/W	0h	DDR3 CAS Write latency. Value of 0, 1, 2, and 3 (CAS write latency of 5, 6, 7, and 8) are supported. Use the lowest value supported for best performance. All other values are reserved.
15-14	NARROW_MODE	R/W	0h	SDRAM data bus width. Set to 0 for 32 bit and set to 1 for 16 bit. All other values are reserved.

Table 9-171. EMIF4D_SDRAM_CONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-10	CL	R/W	0h	CAS Latency. The value of this field defines the CAS latency to be used when accessing connected SDRAM devices. Value of 2, 4, 6, 8, 10, 12, and 14 (CAS latency of 5, 6, 7, 8, 9, 10, and 11) are supported for DDR3. Value of 3, 4, 5, 6, 7, and 8 (CAS latency of 3, 4, 5, 6, 7, and 8) are supported for LPDDR2 SDRAM. All other values are reserved.
9-7	ROWSIZE	R/W	0h	Row Size. Defines the number of row address bits of connected SDRAM devices. Set to 0 for 9 row bits, set to 1 for 10 row bits, set to 2 for 11 row bits, set to 3 for 12 row bits, set to 4 for 13 row bits, set to 5 for 14 row bits, set to 6 for 15 row bits, and set to 7 for 16 row bits. This field is only used when reg_ibank_pos field in SDRAM Config register is set to 1, 2, or 3, or reg_ebank_pos field in SDRAM Config 2 register is set to 1.
6-4	IBANK	R/W	0h	Internal Bank setup. Defines number of banks inside connected SDRAM devices. Set to 0 for 1 bank, set to 1 for 2 banks, set to 2 for 4 banks, and set to 3 for 8 banks. All other values are reserved.
3	EBANK	R/W	0h	External chip select setup. Defines whether SDRAM accesses will use 1 or 2 chip select lines. Set to 0 to use pad_cs_o_n[0] only. Set to 1 to use pad_cs_o_n [1:0]. This bit will automaticlly be set to 0 if reg_cs1nvmen field in the LPDDR2 NVM is set to 1.
2-0	PAGESIZE	R/W	0h	Page Size. Defines the internal page size of connected SDRAM devices. Set to 0 for 256-word page (8 column bits), set to 1 for 512-word page (9 column bits), set to 2 for 1024-word page (10 column bits), and set to 3 for 2048-word page (11 column bits). All other values are reserved.

9.3.4.4 EMIF4D_SDRAM_CONFIG_2 Register (Offset = Ch) [reset = 0h]

EMIF4D_SDRAM_CONFIG_2 is shown in Figure 9-146 and described in Table 9-172.

Return to Summary Table.

Figure 9-146. EMIF4D_SDRAM_CONFIG_2 Register

31	30	29	28	27	26	25	24
RESERVED	CS1NVMEN	RESERVED		EBANK_POS	RESERVED		
R-0-0h	R/W-0h	R-0-0h		R/W-0h	R-0-0h		
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		RDBNUM		RESERVED	RDBSIZE		
R-0-0h		R/W-0h		R-0-0h	R/W-0h		

Table 9-172. EMIF4D_SDRAM_CONFIG_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0	0h	
30	CS1NVMEN	R/W	0h	CS1 LPDDR2 NVM enable. Set to 1 if LPDDR2 NVM is connected to CS1. This bit will automatically be set to 0 if reg_sdram_type field in the SDRAM Config register is not set to LPDDR2.
29-28	RESERVED	R-0	0h	
27	EBANK_POS	R/W	0h	External bank position. Set to 0 to assign external bank address bits from lower OCP address bits as shown in tables. Set to 1 to assign external bank address bits from higher OCP address bits as shown in tables.
26-6	RESERVED	R-0	0h	
5-4	RDBNUM	R/W	0h	Row Buffer setup. Defines number of row buffers inside connected LPDDR2 NVM devices. Set to 0 for 1 row buffer, set to 1 for 2 row buffers, set to 2 for 4 row buffers, and set to 3 for 8 row buffers. All other values are reserved.
3	RESERVED	R-0	0h	
2-0	RDBSIZE	R/W	0h	Row Data Buffer Size. Defines the row data buffer size of connected LPDDR2 NVM devices. Set to 0 for 32 bytes, set to 1 for 64 bytes, set to 2 for 128 bytes, set to 3 for 256 bytes, set to 4 for 512 bytes, set to 5 for 1024 bytes, set to 6 for 2048 bytes, and set to 7 for 4096 bytes.

9.3.4.5 EMIF4D_SDRAM_REFRESH_CTRL Register (Offset = 10h) [reset = 0h]

EMIF4D_SDRAM_REFRESH_CTRL is shown in Figure 9-147 and described in Table 9-173.

Return to Summary Table.

Figure 9-147. EMIF4D_SDRAM_REFRESH_CTRL Register

31	30	29	28	27	26	25	24
INITREF_DIS	RESERVED	SRT	ASR	RESERVED	PASR		
R/W-0h	R-0-0h	R/W-0h	R/W-0h	R-0-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
REFRESH_RATE							
R/W-0h							
7	6	5	4	3	2	1	0
REFRESH_RATE							
R/W-0h							

Table 9-173. EMIF4D_SDRAM_REFRESH_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INITREF_DIS	R/W	0h	Initialization and Refresh disable. When set to 1, EMIF will disable SDRAM initialization and refreshes, but will carry out SDRAM write/read transactions.
30	RESERVED	R-0	0h	
29	SRT	R/W	0h	DDR3 Self Refresh temperature range. Set to 0 for normal operating temperature range and set to 1 for extended operating temperature range when the reg_asr field is set to 0. This bit must be set to 0 if the reg_asr field is set to 1. A write to this field will cause the EMIF to start the SDRAM initialization sequence.
28	ASR	R/W	0h	DDR3 Auto Self Refresh enable. Set to 1 for auto Self Refresh enable. Set to 0 for manual Self Refresh reference indicated by the reg_srt field. A write to this field will cause the EMIF to start the SDRAM initialization sequence.
27	RESERVED	R-0	0h	
26-24	PASR	R/W	0h	Partial Array Self Refresh. These bits get loaded into the Extended Mode Register of DDR3 during initialization. Set to 0 for full array, set to 1 or 5 for 1/2 array, set to 2 or 6 for 1/4 array, set to 3 or 7 for 1/8 array, and set to 4 for 3/4 array to be refreshed. All other values are reserved. A write to this field will cause the EMIF to start the SDRAM initialization sequence.
23-16	RESERVED	R-0	0h	
15-0	REFRESH_RATE	R/W	0h	Refresh Rate. Value in this field is used to define the rate at which connected SDRAM devices will be refreshed. SDRAM refresh rate = EMIF rate / reg_refresh_rate where EMIF rate is equal to m_clk rate. If reg_refresh_rate < (8*reg_t_rfc)+reg_t_rp+reg_t_rcd+20 then it will be loaded with (8*reg_t_rfc)+reg_t_rp+reg_t_rcd+20. This is done to avoid lock-up situations when illegal values are programmed.

9.3.4.6 EMIF4D_SDRAM_REFRESH_CTRL_SHADOW Register (Offset = 14h) [reset = 0h]

EMIF4D_SDRAM_REFRESH_CTRL_SHADOW is shown in Figure 9-148 and described in Table 9-174.

Return to Summary Table.

Figure 9-148. EMIF4D_SDRAM_REFRESH_CTRL_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REFRESH_RATE_SHDW															
R/W-0h															

Table 9-174. EMIF4D_SDRAM_REFRESH_CTRL_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	
15-0	REFRESH_RATE_SHDW	R/W	0h	Shadow field for reg_refresh_rate. This field is loaded into reg_refresh_rate field in SDRAM Refresh Control register when SIdleAck is asserted. This register is not auto corrected when the value is invalid.

9.3.4.7 EMIF4D_SDRAM_TIMING_1 Register (Offset = 18h) [reset = 0h]

EMIF4D_SDRAM_TIMING_1 is shown in Figure 9-149 and described in Table 9-175.

Return to Summary Table.

Figure 9-149. EMIF4D_SDRAM_TIMING_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T_RTW			T_RP				T_RCD				T_WR				T_RAS
R/W-0h			R/W-0h				R/W-0h				R/W-0h				R/W-0h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T_RAS				T_RC						T_RRD			T_WTR		
R/W-0h				R/W-0h						R/W-0h			R/W-0h		

Table 9-175. EMIF4D_SDRAM_TIMING_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	T_RTW	R/W	0h	Minimum number of DDR clock cycles between Read to Write data phases, minus one.
28-25	T_RP	R/W	0h	Minimum number of DDR clock cycles from Precharge to Activate or Refresh, minus one.
24-21	T_RCD	R/W	0h	Minimum number of DDR clock cycles from Activate to Read or Write, minus one.
20-17	T_WR	R/W	0h	Minimum number of DDR clock cycles from last Write transfer to Pre-charge, minus one.
16-12	T_RAS	R/W	0h	Minimum number of DDR clock cycles from Activate to Pre-charge, minus one. reg_t_ras >= reg_t_rcd.
11-6	T_RC	R/W	0h	Minimum number of DDR clock cycles from Activate to Activate, minus one.
5-3	T_RRD	R/W	0h	Minimum number of DDR clock cycles from Activate to Activate for a different bank, minus one. For an 8-bank DDR3 or 8-bank LPDDR2, this field must be equal to ((tFAW/(4*tCK))-1). For 4-bank LPDDR2, the field must be equal to (tRRD/tCK)-1.
2-0	T_WTR	R/W	0h	Minimum number of DDR clock cycles from last Write to Read, minus one.

9.3.4.8 EMIF4D_SDRAM_TIMING_1_SHADOW Register (Offset = 1Ch) [reset = 0h]

EMIF4D_SDRAM_TIMING_1_SHADOW is shown in Figure 9-150 and described in Table 9-176.

Return to Summary Table.

Figure 9-150. EMIF4D_SDRAM_TIMING_1_SHADOW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="3">T_RTW_SHDW</td><td colspan="4">T_RP_SHDW</td><td>T_RCD_SHDW</td></tr>
<tr><td colspan="3">R/W-0h</td><td colspan="4">R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="3">T_RCD_SHDW</td><td colspan="4">T_WR_SHDW</td><td>T_RAS_SHDW</td></tr>
<tr><td colspan="3">R/W-0h</td><td colspan="4">R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="4">T_RAS_SHDW</td><td colspan="4">T_RC_SHDW</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">T_RC_SHDW</td><td colspan="3">T_RRD_SHDW</td><td colspan="3">T_WTR_SHDW</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="3">R/W-0h</td><td colspan="3">R/W-0h</td></tr>
</table>

Table 9-176. EMIF4D_SDRAM_TIMING_1_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	T_RTW_SHDW	R/W	0h	Shadow field for reg_t_rtw. This field is loaded into reg_t_rtw field in SDRAM Timing 1 register when SIdleAck is asserted.
28-25	T_RP_SHDW	R/W	0h	Shadow field for reg_t_rp. This field is loaded into reg_t_rp field in SDRAM Timing 1 register when SIdleAck is asserted.
24-21	T_RCD_SHDW	R/W	0h	Shadow field for reg_t_rcd. This field is loaded into reg_t_rcd field in SDRAM Timing 1 register when SIdleAck is asserted.
20-17	T_WR_SHDW	R/W	0h	Shadow field for reg_t_wr. This field is loaded into reg_t_wr field in SDRAM Timing 1 register when SIdleAck is asserted.
16-12	T_RAS_SHDW	R/W	0h	Shadow field for reg_t_ras. This field is loaded into reg_t_ras field in SDRAM Timing 1 register when SIdleAck is asserted.
11-6	T_RC_SHDW	R/W	0h	Shadow field for reg_t_rc. This field is loaded into reg_t_rc field in SDRAM Timing 1 register when SIdleAck is asserted.
5-3	T_RRD_SHDW	R/W	0h	Shadow field for reg_t_rrd. This field is loaded into reg_t_rrd field in SDRAM Timing 1 register when SIdleAck is asserted.
2-0	T_WTR_SHDW	R/W	0h	Shadow field for reg_t_wtr. This field is loaded into reg_t_wtr field in SDRAM Timing 1 register when SIdleAck is asserted.

9.3.4.9 EMIF4D_SDRAM_TIMING_2 Register (Offset = 20h) [reset = 0h]

EMIF4D_SDRAM_TIMING_2 is shown in Figure 9-151 and described in Table 9-177.

Return to Summary Table.

Figure 9-151. EMIF4D_SDRAM_TIMING_2 Register

31	30	29	28	27	26	25	24
RESERVED	T_XP			RESERVED			T_XSNR
R-0-0h	R/W-0h			R/W-0h			R/W-0h
23	22	21	20	19	18	17	16
T_XSNR							
R/W-0h							
15	14	13	12	11	10	9	8
T_XSRD							
R/W-0h							
7	6	5	4	3	2	1	0
T_XSRD		T_RTP			T_CKE		
R/W-0h		R/W-0h			R/W-0h		

Table 9-177. EMIF4D_SDRAM_TIMING_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0	0h	
30-28	T_XP	R/W	0h	Minimum number of DDR clock cycles from Powerdown exit to any command other than a Read command, minus one.
27-25	RESERVED	R/W	0h	
24-16	T_XSNR	R/W	0h	Minimum number of DDR clock cycles from Self-Refresh exit to any command other than a Read command, minus one.
15-6	T_XSRD	R/W	0h	Minimum number of DDR clock cycles from Self-Refresh exit to a Read command, minus one.
5-3	T_RTP	R/W	0h	Minimum number of DDR clock cycles from the last Read command to a Pre-charge command for DDR3, minus one.
2-0	T_CKE	R/W	0h	Minimum number of DDR clock cycles between pad_cke_o changes, minus one.

9.3.4.10 EMIF4D_SDRAM_TIMING_2_SHADOW Register (Offset = 24h) [reset = 0h]

EMIF4D_SDRAM_TIMING_2_SHADOW is shown in Figure 9-152 and described in Table 9-178.

Return to Summary Table.

Figure 9-152. EMIF4D_SDRAM_TIMING_2_SHADOW Register

31	30	29	28	27	26	25	24
RESERVED	T_XP_SHDW			RESERVED			T_XSNR_SHDW
R-0-0h	R/W-0h			R/W-0h			R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
T_XSNR_SHDW							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
T_XSRD_SHDW							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
T_XSRD_SHDW		T_RTP_SHDW			T_CKE_SHDW		
R/W-0h		R/W-0h			R/W-0h		

Table 9-178. EMIF4D_SDRAM_TIMING_2_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0	0h	
30-28	T_XP_SHDW	R/W	0h	Shadow field for reg_t_xp. This field is loaded into reg_t_xp field in SDRAM Timing 2 register when SIdleAck is asserted.
27-25	RESERVED	R/W	0h	
24-16	T_XSNR_SHDW	R/W	0h	Shadow field for reg_t_xsnr. This field is loaded into reg_t_xsnr field in SDRAM Timing 2 register when SIdleAck is asserted.
15-6	T_XSRD_SHDW	R/W	0h	Shadow field for reg_t_xsrd. This field is loaded into reg_t_xsrd field in SDRAM Timing 2 register when SIdleAck is asserted.
5-3	T_RTP_SHDW	R/W	0h	Shadow field for reg_t_rtp. This field is loaded into reg_t_rtp field in SDRAM Timing 2 register when SIdleAck is asserted.
2-0	T_CKE_SHDW	R/W	0h	Shadow field for reg_t_cke. This field is loaded into reg_t_cke field in SDRAM Timing 2 register when SIdleAck is asserted.

9.3.4.11 EMIF4D_SDRAM_TIMING_3 Register (Offset = 28h) [reset = 0h]

EMIF4D_SDRAM_TIMING_3 is shown in Figure 9-153 and described in Table 9-179.

Return to Summary Table.

Figure 9-153. EMIF4D_SDRAM_TIMING_3 Register

31	30	29	28	27	26	25	24
T_PDLL_UL				T_CSTA			
R/W-0h				R/W-0h			

23	22	21	20	19	18	17	16
T_CKESR			ZQ_ZQCS				
R/W-0h			R/W-0h				

15	14	13	12	11	10	9	8
ZQ_ZQCS	T_TDQSCKMAX		T_RFC				
R/W-0h	R/W-0h		R/W-0h				

7	6	5	4	3	2	1	0
T_RFC				T_RAS_MAX			
R/W-0h				R/W-0h			

Table 9-179. EMIF4D_SDRAM_TIMING_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	T_PDLL_UL	R/W	0h	Minimum number of DDR clock cycles for PHY DLL to unlock. A value of N will be equal to N x 128 clocks.
27-24	T_CSTA	R/W	0h	Minimum number of DDR clock cycles between write-to-write or read-to-read data phases to different chip selects, minus one.
23-21	T_CKESR	R/W	0h	Minimum number of DDR clock cycles for which LPDDR2 must remain in Self Refresh, minus one.
20-15	ZQ_ZQCS	R/W	0h	Number of DDR clock cycles for a ZQCS command, minus one.
14-13	T_TDQSCKMAX	R/W	0h	Number of DDR clock cycles that satisfies tDQSCKmax for LPDDR2, minus one.
12-4	T_RFC	R/W	0h	Minimum number of DDR clock cycles from Refresh or Load Mode to Refresh or Activate, minus one.
3-0	T_RAS_MAX	R/W	0h	Maximum number of reg_refresh_rate intervals from Activate to Precharge command. This field must be set to 0xF for all DDRs supported by this device.

9.3.4.12 EMIF4D_SDRAM_TIMING_3_SHADOW Register (Offset = 2Ch) [reset = 0h]

EMIF4D_SDRAM_TIMING_3_SHADOW is shown in Figure 9-154 and described in Table 9-180.

Return to Summary Table.

Figure 9-154. EMIF4D_SDRAM_TIMING_3_SHADOW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="4">T_PDLL_UL_SHDW</td><td colspan="4">T_CSTA_SHDW</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="3">T_CKESR_SHDW</td><td colspan="5">ZQ_ZQCS_SHDW</td></tr>
<tr><td colspan="3">R/W-0h</td><td colspan="5">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>ZQ_ZQCS_SHDW</td><td colspan="2">T_TDQSCKMAX_SHDW</td><td colspan="5">T_RFC_SHDW</td></tr>
<tr><td>R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="5">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">T_RFC_SHDW</td><td colspan="4">T_RAS_MAX_SHDW</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 9-180. EMIF4D_SDRAM_TIMING_3_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	T_PDLL_UL_SHDW	R/W	0h	Shadow field for reg_t_pdll_ul. This field is loaded into reg_t_pdll_ul field in SDRAM Timing 3 register when SIdleAck is asserted.
27-24	T_CSTA_SHDW	R/W	0h	Shadow field for reg_t_csta. This field is loaded into reg_t_csta field in SDRAM Timing 3 register when SIdleAck is asserted.
23-21	T_CKESR_SHDW	R/W	0h	Shadow field for reg_t_ckesr. This field is loaded into reg_t_ckesr field in SDRAM Timing 3 register when SIdleAck is asserted.
20-15	ZQ_ZQCS_SHDW	R/W	0h	Shadow field for reg_zq_zqcs. This field is loaded into reg_t_tdqsckmax field in SDRAM Timing 3 register when SIdleAck is asserted.
14-13	T_TDQSCKMAX_SHDW	R/W	0h	Shadow field for reg_t_tdqsckmax. This field is loaded into reg_t_tdqsckmax field in SDRAM Timing 3 register when SIdleAck is asserted.
12-4	T_RFC_SHDW	R/W	0h	Shadow field for reg_t_rfc. This field is loaded into reg_t_rfc field in SDRAM Timing 3 register when SIdleAck is asserted.
3-0	T_RAS_MAX_SHDW	R/W	0h	Shadow field for reg_t_ras_max. This field is loaded into reg_t_ras_max field in SDRAM Timing 3 register when SIdleAck is asserted.

9.3.4.13 EMIF4D_LPDDR2_NVM_TIMING Register (Offset = 30h) [reset = 0h]

EMIF4D_LPDDR2_NVM_TIMING is shown in Figure 9-155 and described in Table 9-181.

Return to Summary Table.

Figure 9-155. EMIF4D_LPDDR2_NVM_TIMING Register

31	30	29	28	27	26	25	24
RESERVED	NVM_T_XP			RESERVED	NVM_T_WTR		
R-0-0h	R/W-0h			R-0-0h	R/W-0h		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
NVM_T_RP				NVM_T_WRA			
R/W-0h				R/W-0h			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
NVM_T_RRD							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NVM_T_RCDMIN							
R/W-0h							

Table 9-181. EMIF4D_LPDDR2_NVM_TIMING Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0	0h	
30-28	NVM_T_XP	R/W	0h	Minimum number of DDR clock cycles from Powerdown exit to any command, minus one.
27	RESERVED	R-0	0h	
26-24	NVM_T_WTR	R/W	0h	Minimum number of DDR clock cycles from last Write to Read, minus one.
23-20	NVM_T_RP	R/W	0h	Minimum number of DDR clock cycles from Preactive to Activate, minus one.
19-16	NVM_T_WRA	R/W	0h	Minimum number of DDR clock cycles from last Write transfer to Activate, minus one.
15-8	NVM_T_RRD	R/W	0h	Minimum number of DDR clock cycles from Activate to Activate for a different bank, minus one.
7-0	NVM_T_RCDMIN	R/W	0h	Minimum number of DDR clock cycles from Activate to Read or Write, minus one.

9.3.4.14 EMIF4D_LPDDR2_NVM_TIMING_SHADOW Register (Offset = 34h) [reset = 0h]

EMIF4D_LPDDR2_NVM_TIMING_SHADOW is shown in Figure 9-156 and described in Table 9-182.

Return to Summary Table.

Figure 9-156. EMIF4D_LPDDR2_NVM_TIMING_SHADOW Register

31	30	29	28	27	26	25	24
RESERVED	NVM_T_XP_SHDW			RESERVED	NVM_T_WTR_SHDW		
R-0-0h	R/W-0h			R-0-0h	R/W-0h		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
NVM_T_RP_SHDW				NVM_T_WRA_SHDW			
R/W-0h				R/W-0h			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
NVM_T_RRD_SHDW							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NVM_T_RCDMIN_SHDW							
R/W-0h							

Table 9-182. EMIF4D_LPDDR2_NVM_TIMING_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0	0h	
30-28	NVM_T_XP_SHDW	R/W	0h	Shadow field for reg_nvm_t_xp. This field is loaded into reg_nvm_t_xp field in LPDDR2 NVM Timing register when SIdleAck is asserted.
27	RESERVED	R-0	0h	
26-24	NVM_T_WTR_SHDW	R/W	0h	Shadow field for reg_nvm_t_wtr. This field is loaded into reg_nvm_t_wtr field in LPDDR2 NVM Timing register when SIdleAck is asserted.
23-20	NVM_T_RP_SHDW	R/W	0h	Shadow field for reg_nvm_t_rp. This field is loaded into reg_nvm_t_rp field in LPDDR2 NVM Timing register when SIdleAck is asserted.
19-16	NVM_T_WRA_SHDW	R/W	0h	Shadow field for reg_nvm_t_wra. This field is loaded into reg_nvm_t_wra field in LPDDR2 NVM Timing register when SIdleAck is asserted.
15-8	NVM_T_RRD_SHDW	R/W	0h	Shadow field for reg_nvm_t_rrd. This field is loaded into reg_nvm_t_rrd field in LPDDR2 NVM Timing register when SIdleAck is asserted.
7-0	NVM_T_RCDMIN_SHDW	R/W	0h	Shadow field for reg_nvm_t_rcdmin. This field is loaded into reg_nvm_t_rcdmin field in LPDDR2 NVM Timing register when SIdleAck is asserted.

9.3.4.15 EMIF4D_POWER_MANAGEMENT_CTRL Register (Offset = 38h) [reset = 0h]

EMIF4D_POWER_MANAGEMENT_CTRL is shown in Figure 9-157 and described in Table 9-183.

Return to Summary Table.

Figure 9-157. EMIF4D_POWER_MANAGEMENT_CTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="4">PD_TIM</td><td>DPD_EN</td><td colspan="3">LP_MODE</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R/W-0h</td><td colspan="3">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">SR_TIM</td><td colspan="4">CS_TIM</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 9-183. EMIF4D_POWER_MANAGEMENT_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	
15-12	PD_TIM	R/W	0h	Power Mangement timer for Power-Down. The EMIF will put the external SDRAM in Power-Down mode after the EMIF is idle for these number of DDR clock cycles and if reg_lp_mode field is set to 4. Set to 0 to immediately enter Power-Down mode. Set to 1 for 16 clocks, set to 2 for 32 clocks, set to 3 for 64 clocks, set to 4 for 128 clocks, set to 5 for 256 clocks, set to 6 for 512 clocks, set to 7 for 1024 clocks, set to 8 for 2048 clocks, set to 9 for 4096 clocks, set to 10 for 8192 clocks, set to 11 for 16384 clocks, set to 12 for 32768 clocks, set to 13 for 65536 clocks, set to 14 for 131072 clocks, and set to 15 for 262144 clocks.
11	DPD_EN	R/W	0h	Deep Power Down enable. Set to 0 for normal operation. Set to 1 to enter deep power down mode. This mode will override the reg_lp_mode field setting.
10-8	LP_MODE	R/W	0h	Automatic Power Management enable. Set to 1 for Clock Stop, set to 2 for Self Refresh, and set to 4 for Power-Down. All other values will disable automatic power management.
7-4	SR_TIM	R/W	0h	Power Mangement timer for Self Refresh. The EMIF will put the external SDRAM in Self Refresh mode after the EMIF is idle for these number of DDR clock cycles and if reg_lp_mode field is set to 2. Set to 0 to immediately enter Self Refresh mode. Set to 1 for 16 clocks, set to 2 for 32 clocks, set to 3 for 64 clocks, set to 4 for 128 clocks, set to 5 for 256 clocks, set to 6 for 512 clocks, set to 7 for 1024 clocks, set to 8 for 2048 clocks, set to 9 for 4096 clocks, set to 10 for 8192 clocks, set to 11 for 16384 clocks, set to 12 for 32768 clocks, set to 13 for 65536 clocks, set to 14 for 131072 clocks, and set to 15 for 262144 clocks.

Table 9-183. EMIF4D_POWER_MANAGEMENT_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	CS_TIM	R/W	0h	Power Mangement timer for Clock Stop. The EMIF will put the external SDRAM in Clock Stop mode after the EMIF is idle for these number of DDR clock cycles and if reg_lp_mode field is set to 1. Set to 0 to immediately enter Clock Stop mode. Set to 1 for 16 clocks, set to 2 for 32 clocks, set to 3 for 64 clocks, set to 4 for 128 clocks, set to 5 for 256 clocks, set to 6 for 512 clocks, set to 7 for 1024 clocks, set to 8 for 2048 clocks, set to 9 for 4096 clocks, set to 10 for 8192 clocks, set to 11 for 16384 clocks, set to 12 for 32768 clocks, set to 13 for 65536 clocks, set to 14 for 131072 clocks, and set to 15 for 262144 clocks.

9.3.4.16 EMIF4D_POWER_MANAGEMENT_CTRL_SHADOW Register (Offset = 3Ch) [reset = 0h]

EMIF4D_POWER_MANAGEMENT_CTRL_SHADOW is shown in Figure 9-158 and described in Table 9-184.

Return to Summary Table.

Figure 9-158. EMIF4D_POWER_MANAGEMENT_CTRL_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD_TIM_SHDW				RESERVED				SR_TIM_SHDW				CS_TIM_SHDW			
R/W-0h				R-0-0h				R/W-0h				R/W-0h			

Table 9-184. EMIF4D_POWER_MANAGEMENT_CTRL_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	
15-12	PD_TIM_SHDW	R/W	0h	Shadow field for reg_pd_tim. This field is loaded into reg_pd_tim field in Power Management Control register when SIdleAck is asserted.
11-8	RESERVED	R-0	0h	
7-4	SR_TIM_SHDW	R/W	0h	Shadow field for reg_sr_tim. This field is loaded into reg_sr_tim field in Power Management Control register when SIdleAck is asserted.
3-0	CS_TIM_SHDW	R/W	0h	Shadow field for reg_cs_tim. This field is loaded into reg_cs_tim field in Power Management Control register when SIdleAck is asserted.

9.3.4.17 EMIF4D_LPDDR2_MODE_REG_DATA Register (Offset = 40h) [reset = 0h]

EMIF4D_LPDDR2_MODE_REG_DATA is shown in Figure 9-159 and described in Table 9-185.

Return to Summary Table.

Figure 9-159. EMIF4D_LPDDR2_MODE_REG_DATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																									VALUE_0						
R-0-0h																									R/W-0h						

Table 9-185. EMIF4D_LPDDR2_MODE_REG_DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R-0	0h	
6-0	VALUE_0	R/W	0h	Mode register value.

9.3.4.18 EMIF4D_LPDDR2_MODE_REG_CONFIG Register (Offset = 50h) [reset = 0h]

EMIF4D_LPDDR2_MODE_REG_CONFIG is shown in Figure 9-160 and described in Table 9-186.

Return to Summary Table.

Figure 9-160. EMIF4D_LPDDR2_MODE_REG_CONFIG Register

31	30	29	28	27	26	25	24
CS	REFRESH_EN	RESERVED					
R/W-0h	R/W-0h	R-0-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
ADDR							
R/W-0h							

Table 9-186. EMIF4D_LPDDR2_MODE_REG_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CS	R/W	0h	Chip select to issue mode register command. Set to 0 for CS0 and set to 1 for CS1.
30	REFRESH_EN	R/W	0h	Refresh Enable after MRW write. If a Mode Data register write occurs with this bit set to 1, the refresh operations will commence.
29-8	RESERVED	R-0	0h	
7-0	ADDR	R/W	0h	Mode register address.

9.3.4.19 EMIF4D_OCP_CONFIG Register (Offset = 54h) [reset = 07770000h]

EMIF4D_OCP_CONFIG is shown in Figure 9-161 and described in Table 9-187.

Return to Summary Table.

Figure 9-161. EMIF4D_OCP_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED				SYS_THRESH_MAX			
R-0-0h				R/W-7h			
23	22	21	20	19	18	17	16
MPU_THRESH_MAX				LL_THRESH_MAX			
R/W-7h				R/W-7h			
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							
R-0-0h							

Table 9-187. EMIF4D_OCP_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R-0	0h	
27-24	SYS_THRESH_MAX	R/W	7h	System OCP Threshold Maximum. The number of commands the system interface can consume in the command FIFO. The value is used to determine when to stop future request, writing a zero will reserve no space for the associated interface. In the event the value is set to zero and a request is seen for that interface, the command FIFO will assume a value of 1. Since the low-latency interface has effectively a higher priority, the only way for the system interface to use all command FIFO entries is to set the reg_ll_thresh_max to zero.
23-20	MPU_THRESH_MAX	R/W	7h	System MPU Threshold Maximum. The number of commands the MPU interface can consume in the command FIFO. The value is used to determine when to stop future request, writing a zero will reserve no space for the associated interface. In the event the value is set to zero and a request is seen for that interface, the command FIFO will assume a value of 1. Since the low-latency interface has effectively a higher priority, the only way for the mpu interface to use all command FIFO entries is to set the reg_ll_thresh_max to zero.
19-16	LL_THRESH_MAX	R/W	7h	Low-latency OCP Threshold Maximum. The number of commands the low latency interface can consume in the command FIFO. The value is used to determine when to stop future request, writing a zero will reserve no space for the associated interface. In the event the value is set to zero and a request is seen for that interface, the command FIFO will assume a value of 1.
15-0	RESERVED	R-0	0h	

9.3.4.20 EMIF4D_OCP_CONFIG_VALUE_1 Register (Offset = 58h) [reset = 9000190Ah]

EMIF4D_OCP_CONFIG_VALUE_1 is shown in Figure 9-162 and described in Table 9-188.

Return to Summary Table.

Figure 9-162. EMIF4D_OCP_CONFIG_VALUE_1 Register

31	30	29	28	27	26	25	24
SYS_BUS_WIDTH		LL_BUS_WIDTH		RESERVED			
R-2h		R-1h		R-0-0h			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
WR_FIFO_DEPTH							
R-19h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMD_FIFO_DEPTH							
R-Ah							

Table 9-188. EMIF4D_OCP_CONFIG_VALUE_1 Register Field Descriptions

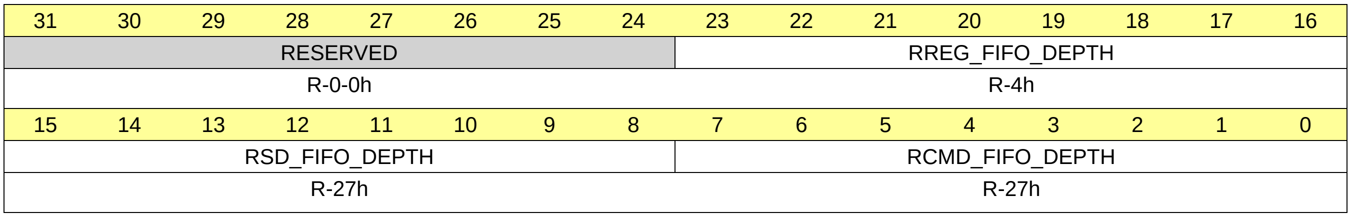
Bit	Field	Type	Reset	Description
31-30	SYS_BUS_WIDTH	R	2h	System OCP data bus width for a particular configuration. 0 = 32 bit wide, 1 = 64 bit wide, 2 = 128 bit wide, and 3 = 256 bit wide
29-28	LL_BUS_WIDTH	R	1h	Low-latency OCP data bus width for a particular configuration. 0 = 32 bit wide, 1 = 64 bit wide, 2 = 128 bit wide, and 3 = 256 bit wide
27-16	RESERVED	R-0	0h	
15-8	WR_FIFO_DEPTH	R	19h	Write Data FIFO depth for a particular configuration.
7-0	CMD_FIFO_DEPTH	R	Ah	Command FIFO depth for a particular configuration.

9.3.4.21 EMIF4D_OCP_CONFIG_VALUE_2 Register (Offset = 5Ch) [reset = 00042727h]

EMIF4D_OCP_CONFIG_VALUE_2 is shown in Figure 9-163 and described in Table 9-189.

Return to Summary Table.

Figure 9-163. EMIF4D_OCP_CONFIG_VALUE_2 Register



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								RREG_FIFO_DEPTH							
R-0-0h								R-4h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RSD_FIFO_DEPTH								RCMD_FIFO_DEPTH							
R-27h								R-27h							

Table 9-189. EMIF4D_OCP_CONFIG_VALUE_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R-0	0h	
23-16	RREG_FIFO_DEPTH	R	4h	Register Read Data FIFO depth for a particular configuration.
15-8	RSD_FIFO_DEPTH	R	27h	SDRAM Read Data FIFO depth for a particular configuration.
7-0	RCMD_FIFO_DEPTH	R	27h	Read Command FIFO depth for a particular configuration.

9.3.4.22 EMIF4D_IODFT_TEST_LOGIC_GLOBAL_CTRL Register (Offset = 60h) [reset = 2011h]

EMIF4D_IODFT_TEST_LOGIC_GLOBAL_CTRL is shown in Figure 9-164 and described in Table 9-190.

Return to Summary Table.

Figure 9-164. EMIF4D_IODFT_TEST_LOGIC_GLOBAL_CTRL Register

31	30	29	28	27	26	25	24
TLEC							
R/W-0h							
23	22	21	20	19	18	17	16
TLEC							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED	MT	ACT_CAP_EN	OPG_LD	RESERVED	RESET_PHY	RESERVED	MMS
R-0-0h	R/W-0h	R/W-1h	R/W-0h	R-0h	R/W-0h	R-0-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED		MC		PC			TM
R-0-0h		R/W-1h		R/W-0h			R/W-1h

Table 9-190. EMIF4D_IODFT_TEST_LOGIC_GLOBAL_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TLEC	R/W	0h	IODFT Test Logic Execution Counter. Contains the number of cycle that the MISR signature will be accumulated. Upon the expiration of the counter the MISR capture will be turned off.
15	RESERVED	R-0	0h	
14	MT	R/W	0h	MISR on/off trigger command. 0h = inactive/no affect. 1h = MISR capture start on the first write or read command to the memory and continues to update the signature until reg_tlec expires, when mc = 3h. 1h = pattern generator starts on the first write or read command to the memory and continues to update the signature until reg_tlec expires, when reg_pc = 1h, 2h, 3h, 5h, 6h, or 7h. These bits are cleared when reg_tlec expires.
13	ACT_CAP_EN	R/W	1h	Active cycles capture enable. If set to a 1 the MISRs and pattern generators will shift only during active cycles. If set to a 0 the MISRs and pattern generators will shift every clock cycle.
12	OPG_LD	R/W	0h	Load pattern generators' initial value. Set to 1 to load an initial value in the pattern generators from reg_tlec.
11	RESERVED	R	0h	
10	RESET_PHY	R/W	0h	Reset DDR PHY.
9	RESERVED	R-0	0h	
8	MMS	R/W	0h	Chooses the source of the MISR input. Set to 0 for output register, and set to 1 for input capture.
7-6	RESERVED	R-0	0h	
5-4	MC	R/W	1h	MISR state. Set to 0 to download results. Set to 1 to hold current value. Set to 2 to load initial value from reg_pc bits. Set to 3 to enable MISR to capture signature.

Table 9-190. EMIF4D_IODFT_TEST_LOGIC_GLOBAL_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-1	PC	R/W	0h	Pattern code. Defines the type of pattern that is selected for the pattern generators. Set to 0 for functional mode and set to 4 to hold current register value. For output pattern generator, set to 1 for random XOR, set to 2 for random XNOR, and set to 3 for an 8 bit shifter. For input pattern generator, set to 5 for random XOR, set to 6 for random XNOR, and set to 7 for an 8 bit shifter.
0	TM	R/W	1h	Functional mode enable. Set to 1 for functional mode, and set to 0 for IODFT mode.

9.3.4.23 EMIF4D_IODFT_TEST_LOGIC_CTRL_MISR_RESULT Register (Offset = 64h) [reset = 0h]

EMIF4D_IODFT_TEST_LOGIC_CTRL_MISR_RESULT is shown in Figure 9-165 and described in Table 9-191.

Return to Summary Table.

Figure 9-165. EMIF4D_IODFT_TEST_LOGIC_CTRL_MISR_RESULT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">DQM_TLMR</td></tr>
<tr><td colspan="2">R-0-0h</td><td colspan="6">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="4">DQM_TLMR</td><td>RESERVED</td><td colspan="3">CTL_TLMR</td></tr>
<tr><td colspan="4">R-0h</td><td>R-0-0h</td><td colspan="3">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">CTL_TLMR</td></tr>
<tr><td colspan="8">R-0h</td></tr>
</table>

Table 9-191. EMIF4D_IODFT_TEST_LOGIC_CTRL_MISR_RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R-0	0h	
21-12	DQM_TLMR	R	0h	This contains the MISR result signature of a given test after the download function is executed. This result is for the DQM signals.
11	RESERVED	R-0	0h	
10-0	CTL_TLMR	R	0h	This contains the MISR result signature of a given test after the download function is executed. This result is for the control signals.

9.3.4.24 EMIF4D_IODFT_TEST_LOGIC_ADDR_MISR_RESULT Register (Offset = 68h) [reset = 0h]

EMIF4D_IODFT_TEST_LOGIC_ADDR_MISR_RESULT is shown in Figure 9-166 and described in Table 9-192.

Return to Summary Table.

Figure 9-166. EMIF4D_IODFT_TEST_LOGIC_ADDR_MISR_RESULT Register

31-21	20-0
RESERVED	ADDR_TLMR
R-0-0h	R-0h

Table 9-192. EMIF4D_IODFT_TEST_LOGIC_ADDR_MISR_RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R-0	0h	
20-0	ADDR_TLMR	R	0h	This contains the MISR result signature of a given test after the download function is executed. This result is for the address signals.

9.3.4.25 EMIF4D_IODFT_TEST_LOGIC_DATA_MISR_RESULT_1 Register (Offset = 6Ch) [reset = 0h]

EMIF4D_IODFT_TEST_LOGIC_DATA_MISR_RESULT_1 is shown in Figure 9-167 and described in Table 9-193.

Return to Summary Table.

Figure 9-167. EMIF4D_IODFT_TEST_LOGIC_DATA_MISR_RESULT_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_TLMR_31_0																															
R-0h																															

Table 9-193. EMIF4D_IODFT_TEST_LOGIC_DATA_MISR_RESULT_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_TLMR_31_0	R	0h	This contains the least significant bits of the MISR result signature of a given test after the download function is executed. This result is for data bus.

9.3.4.26 EMIF4D_IODFT_TEST_LOGIC_DATA_MISR_RESULT_2 Register (Offset = 70h) [reset = 0h]

EMIF4D_IODFT_TEST_LOGIC_DATA_MISR_RESULT_2 is shown in Figure 9-168 and described in Table 9-194.

Return to Summary Table.

Figure 9-168. EMIF4D_IODFT_TEST_LOGIC_DATA_MISR_RESULT_2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_TLMR_63_32																															
R-0h																															

Table 9-194. EMIF4D_IODFT_TEST_LOGIC_DATA_MISR_RESULT_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_TLMR_63_32	R	0h	This contains the middle bits of the MISR result signature of a given test after the download function is executed. This result is for data bus.

9.3.4.27 EMIF4D_IODFT_TEST_LOGIC_DATA_MISR_RESULT_3 Register (Offset = 74h) [reset = 0h]

EMIF4D_IODFT_TEST_LOGIC_DATA_MISR_RESULT_3 is shown in Figure 9-169 and described in Table 9-195.

Return to Summary Table.

Figure 9-169. EMIF4D_IODFT_TEST_LOGIC_DATA_MISR_RESULT_3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					DATA_TLMR_66_64		
R-0-0h					R-0h		

Table 9-195. EMIF4D_IODFT_TEST_LOGIC_DATA_MISR_RESULT_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	
2-0	DATA_TLMR_66_64	R	0h	This contains the most significant bits of the MISR result signature of a given test after the download function is executed. This result is for data bus.

9.3.4.28 EMIF4D_PERFORMANCE_CTR_1 Register (Offset = 80h) [reset = 0h]

EMIF4D_PERFORMANCE_CTR_1 is shown in Figure 9-170 and described in Table 9-196.

Return to Summary Table.

Figure 9-170. EMIF4D_PERFORMANCE_CTR_1 Register

31-0
CTR1
R-0h

Table 9-196. EMIF4D_PERFORMANCE_CTR_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CTR1	R	0h	32 bit counter that can be configured as specified in the Performance Counter Config Register and Performance Counter Master Region Select Register.

9.3.4.29 EMIF4D_PERFORMANCE_CTR_2 Register (Offset = 84h) [reset = 0h]

EMIF4D_PERFORMANCE_CTR_2 is shown in Figure 9-171 and described in Table 9-197.

Return to Summary Table.

Figure 9-171. EMIF4D_PERFORMANCE_CTR_2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTR2																															
R-0h																															

Table 9-197. EMIF4D_PERFORMANCE_CTR_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CTR2	R	0h	32 bit counter that can be configured as specified in the Performance Counter Config Register and Performance Counter Master Region Select Register.

9.3.4.30 EMIF4D_PERFORMANCE_CTR_CONFIG Register (Offset = 88h) [reset = 00010000h]

EMIF4D_PERFORMANCE_CTR_CONFIG is shown in Figure 9-172 and described in Table 9-198.

Return to Summary Table.

Figure 9-172. EMIF4D_PERFORMANCE_CTR_CONFIG Register

31	30	29	28	27	26	25	24
CNTR2_MCONNID_EN	CNTR2_REGION_EN	RESERVED					
R/W-0h	R/W-0h	R-0-0h					
23	22	21	20	19	18	17	16
RESERVED				CNTR2_CFG			
R-0-0h				R/W-1h			
15	14	13	12	11	10	9	8
CNTR1_MCONNID_EN	CNTR1_REGION_EN	RESERVED					
R/W-0h	R/W-0h	R-0-0h					
7	6	5	4	3	2	1	0
RESERVED				CNTR1_CFG			
R-0-0h				R/W-0h			

Table 9-198. EMIF4D_PERFORMANCE_CTR_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CNTR2_MCONNID_EN	R/W	0h	MConnID filter enable for Performance Counter 2 register.
30	CNTR2_REGION_EN	R/W	0h	Chip Select filter enable for Performance Counter 2 register.
29-20	RESERVED	R-0	0h	
19-16	CNTR2_CFG	R/W	1h	Filter configuration for Performance Counter 2. Refer to table for details.
15	CNTR1_MCONNID_EN	R/W	0h	MConnID filter enable for Performance Counter 1 register.
14	CNTR1_REGION_EN	R/W	0h	Chip Select filter enable for Performance Counter 1 register.
13-4	RESERVED	R-0	0h	
3-0	CNTR1_CFG	R/W	0h	Filter configuration for Performance Counter 1. Refer to table for details.

9.3.4.31 EMIF4D_PERFORMANCE_CTR_MASTER_REGION_SELECT Register (Offset = 8Ch) [reset = 0h]

EMIF4D_PERFORMANCE_CTR_MASTER_REGION_SELECT is shown in Figure 9-173 and described in Table 9-199.

Return to Summary Table.

Figure 9-173. EMIF4D_PERFORMANCE_CTR_MASTER_REGION_SELECT Register

31	30	29	28	27	26	25	24
MCONNID2							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED						REGION_SEL2	
R-0-0h						R/W-0h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MCONNID1							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						REGION_SEL1	
R-0-0h						R/W-0h	

Table 9-199. EMIF4D_PERFORMANCE_CTR_MASTER_REGION_SELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MCONNID2	R/W	0h	MConnID for Performance Counter 2 register.
23-18	RESERVED	R-0	0h	
17-16	REGION_SEL2	R/W	0h	MAddrSpace for Performance Counter 2 register.
15-8	MCONNID1	R/W	0h	MConnID for Performance Counter 1 register.
7-2	RESERVED	R-0	0h	
1-0	REGION_SEL1	R/W	0h	MAddrSpace for Performance Counter 1 register.

9.3.4.32 EMIF4D_PERFORMANCE_CTR_TIME Register (Offset = 90h) [reset = 0h]

EMIF4D_PERFORMANCE_CTR_TIME is shown in Figure 9-174 and described in Table 9-200.

Return to Summary Table.

Figure 9-174. EMIF4D_PERFORMANCE_CTR_TIME Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOTAL_TIME																															
R-0h																															

Table 9-200. EMIF4D_PERFORMANCE_CTR_TIME Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TOTAL_TIME	R	0h	32 bit counter that continuously counts number for m_clk cycles elapsed after EMIF is brought out of reset.

9.3.4.33 EMIF4D_MISC_REG Register (Offset = 94h) [reset = 0h]

EMIF4D_MISC_REG is shown in Figure 9-175 and described in Table 9-201.

Return to Summary Table.

Figure 9-175. EMIF4D_MISC_REG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							DLL_CALIB_OS
R-0h							R/W-0h

Table 9-201. EMIF4D_MISC_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	DLL_CALIB_OS	R/W	0h	phy_dll_calib one shot : Setting bit to 1 generates a phy_pll_calib pulse. Bit is self cleared when pll_calib gets generated

9.3.4.34 EMIF4D_DLL_CALIB_CTRL Register (Offset = 98h) [reset = 00090000h]

EMIF4D_DLL_CALIB_CTRL is shown in Figure 9-176 and described in Table 9-202.

Return to Summary Table.

Figure 9-176. EMIF4D_DLL_CALIB_CTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED												ACK_WAIT			
R-0-0h												R/W-9h			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							DLL_CALIB_INTERVAL								
R-0-0h							R/W-0h								

Table 9-202. EMIF4D_DLL_CALIB_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	
19-16	ACK_WAIT	R/W	9h	The ack_wait determines the required wait time after a phy_dll_calib is generated before another command can be sent
15-9	RESERVED	R-0	0h	
8-0	DLL_CALIB_INTERVAL	R/W	0h	The dll_calib_interval field determines the interval between phy_dll_calib generation. This value is multiplied by a precounter of 16 m_clk cycles. Program this field one less the value you are targeting program 1 to achieve interval of 2 (minimum interval supported). Programming zero turns off function. Note the final intervals between dll_calib generation is also a function of ACK_WAIT . Final periodic interval is calculated by: ((dll_calib_interval+1)*16)+ACK_WAIT

9.3.4.35 EMIF4D_DLL_CALIB_CTRL_SHADOW Register (Offset = 9Ch) [reset = 00090000h]

EMIF4D_DLL_CALIB_CTRL_SHADOW is shown in Figure 9-177 and described in Table 9-203.

Return to Summary Table.

Figure 9-177. EMIF4D_DLL_CALIB_CTRL_SHADOW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">ACK_WAIT_SHDW</td></tr>
<tr><td colspan="4">R-0-0h</td><td colspan="4">R/W-9h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>DLL_CALIB_INTERVAL_SHDW</td></tr>
<tr><td colspan="7">R-0-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">DLL_CALIB_INTERVAL_SHDW</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 9-203. EMIF4D_DLL_CALIB_CTRL_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	
19-16	ACK_WAIT_SHDW	R/W	9h	Shadow field for ack_wait. This field is loaded into ack_wait field in dll_calib_ctrl register when SIdleAck is asserted
15-9	RESERVED	R-0	0h	
8-0	DLL_CALIB_INTERVAL_SHDW	R/W	0h	Shadow field for dll_calib_interval. This field is loaded into dll_calib_interval field in the dll_calib_ctrl register when SIdleAck is asserted

9.3.4.36 EMIF4D_END_OF_INTR Register (Offset = A0h) [reset = 0h]

EMIF4D_END_OF_INTR is shown in Figure 9-178 and described in Table 9-204.

Return to Summary Table.

Figure 9-178. EMIF4D_END_OF_INTR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="15">RESERVED</td><td>EOI</td></tr>
<tr><td colspan="15">R-0-0h</td><td>R/W-0h</td></tr>
</table>

Table 9-204. EMIF4D_END_OF_INTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	
0	EOI	R/W	0h	Software End Of Interrupt (EOI) control. Write 0 for system OCP interrupt and write 1 for low-latency OCP interrupt. This field always reads 0 (no EOI memory).

9.3.4.37 EMIF4D_SYSTEM_OCP_INTR_RAW_STS Register (Offset = A4h) [reset = 0h]

EMIF4D_SYSTEM_OCP_INTR_RAW_STS is shown in Figure 9-179 and described in Table 9-205.

Return to Summary Table.

Figure 9-179. EMIF4D_SYSTEM_OCP_INTR_RAW_STS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					DNV_SYS	TA_SYS	ERR_SYS
R-0-0h					R/W1S-0h	R/W1S-0h	R/W1S-0h

Table 9-205. EMIF4D_SYSTEM_OCP_INTR_RAW_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	
2	DNV_SYS	R/W1S	0h	Raw status of system OCP interrupt for LPDDR2 NVM data not valid. Write 1 to set the (raw) status, mostly for debug. Writing a 0 has no effect.
1	TA_SYS	R/W1S	0h	Raw status of system OCP interrupt for SDRAM temperature alert. Write 1 to set the (raw) status, mostly for debug. Writing a 0 has no effect.
0	ERR_SYS	R/W1S	0h	Raw status of system OCP interrupt for command and address error. Write 1 to set the (raw) status, mostly for debug. Writing a 0 has no effect.

9.3.4.38 EMIF4D_LOW_LAT_OCP_INTR_RAW_STS Register (Offset = A8h) [reset = 0h]

EMIF4D_LOW_LAT_OCP_INTR_RAW_STS is shown in Figure 9-180 and described in Table 9-206.

Return to Summary Table.

Figure 9-180. EMIF4D_LOW_LAT_OCP_INTR_RAW_STS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					DNV_LL	TA_LL	ERR_LL
R-0-0h					R/W1S-0h	R/W1S-0h	R/W1S-0h

Table 9-206. EMIF4D_LOW_LAT_OCP_INTR_RAW_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	
2	DNV_LL	R/W1S	0h	Raw status of low-latency OCP interrupt for LPDDR2 NVM data not valid. Write 1 to set the (raw) status, mostly for debug. Writing a 0 has no effect.
1	TA_LL	R/W1S	0h	Raw status of low-latency OCP interrupt for SDRAM temperature alert. Write 1 to set the (raw) status, mostly for debug. Writing a 0 has no effect.
0	ERR_LL	R/W1S	0h	Raw status of low-latency OCP interrupt for command and address error. Write 1 to set the (raw) status, mostly for debug. Writing a 0 has no effect.

9.3.4.39 EMIF4D_SYSTEM_OCP_INTR_STS Register (Offset = ACh) [reset = 0h]

EMIF4D_SYSTEM_OCP_INTR_STS is shown in Figure 9-181 and described in Table 9-207.

Return to Summary Table.

Figure 9-181. EMIF4D_SYSTEM_OCP_INTR_STS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					DNV_SYS	TA_SYS	ERR_SYS
R-0-0h					R/W-0h	R/W-0h	R/W-0h

Table 9-207. EMIF4D_SYSTEM_OCP_INTR_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	
2	DNV_SYS	R/W	0h	Enabled status of system OCP interrupt for LPDDR2 NVM data not valid. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, i.e. even if not enabled). Writing a 0 has no effect.
1	TA_SYS	R/W	0h	Enabled status of system OCP interrupt for SDRAM temperature alert. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, i.e. even if not enabled). Writing a 0 has no effect.
0	ERR_SYS	R/W	0h	Enabled status of system OCP interrupt for command and address error. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, i.e. even if not enabled). Writing a 0 has no effect.

9.3.4.40 EMIF4D_LOW_LAT_OCP_INTR_STS Register (Offset = B0h) [reset = 0h]

EMIF4D_LOW_LAT_OCP_INTR_STS is shown in Figure 9-182 and described in Table 9-208.

Return to Summary Table.

Figure 9-182. EMIF4D_LOW_LAT_OCP_INTR_STS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					DNV_LL	TA_LL	ERR_LL
R-0-0h					R/W-0h	R/W-0h	R/W-0h

Table 9-208. EMIF4D_LOW_LAT_OCP_INTR_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	
2	DNV_LL	R/W	0h	Enabled status of low-latency OCP interrupt for LPDDR2 NVM data not valid. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, i.e. even if not enabled). Writing a 0 has no effect.
1	TA_LL	R/W	0h	Enabled status of low-latency OCP interrupt for SDRAM temperature alert. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, i.e. even if not enabled). Writing a 0 has no effect.
0	ERR_LL	R/W	0h	Enabled status of low-latency OCP interrupt for command and address error. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, i.e. even if not enabled). Writing a 0 has no effect.

9.3.4.41 EMIF4D_SYSTEM_OCP_INTR_EN_SET Register (Offset = B4h) [reset = 0h]

EMIF4D_SYSTEM_OCP_INTR_EN_SET is shown in Figure 9-183 and described in Table 9-209.

Return to Summary Table.

Figure 9-183. EMIF4D_SYSTEM_OCP_INTR_EN_SET Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">RESERVED</td><td>EN_DNV_SYS</td><td>EN_TA_SYS</td><td>EN_ERR_SYS</td></tr>
<tr><td colspan="5">R-0-0h</td><td>R/W1S-0h</td><td>R/W1S-0h</td><td>R/W1S-0h</td></tr>
</table>

Table 9-209. EMIF4D_SYSTEM_OCP_INTR_EN_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	
2	EN_DNV_SYS	R/W1S	0h	Enable set for system OCP interrupt for LPDDR2 NVM data not valid. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.
1	EN_TA_SYS	R/W1S	0h	Enable set for system OCP interrupt for SDRAM temperature alert. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.
0	EN_ERR_SYS	R/W1S	0h	Enable set for system OCP interrupt for command and address error. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.

9.3.4.42 EMIF4D_LOW_LAT_OCP_INTR_EN_SET Register (Offset = B8h) [reset = 0h]

EMIF4D_LOW_LAT_OCP_INTR_EN_SET is shown in Figure 9-184 and described in Table 9-210.

Return to Summary Table.

Figure 9-184. EMIF4D_LOW_LAT_OCP_INTR_EN_SET Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">RESERVED</td><td>EN_DNV_LL</td><td>EN_TA_LL</td><td>EN_ERR_LL</td></tr>
<tr><td colspan="5">R-0-0h</td><td>R/W1S-0h</td><td>R/W1S-0h</td><td>R/W1S-0h</td></tr>
</table>

Table 9-210. EMIF4D_LOW_LAT_OCP_INTR_EN_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	
2	EN_DNV_LL	R/W1S	0h	Enable set for low-latency OCP interrupt for LPDDR2 NVM data not valid. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.
1	EN_TA_LL	R/W1S	0h	Enable set for low-latency OCP interrupt for SDRAM temperature alert. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.
0	EN_ERR_LL	R/W1S	0h	Enable set for low-latency OCP interrupt for command and address error. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.

9.3.4.43 EMIF4D_SYSTEM_OCP_INTR_EN_CLR Register (Offset = BCh) [reset = 0h]

EMIF4D_SYSTEM_OCP_INTR_EN_CLR is shown in Figure 9-185 and described in Table 9-211.

Return to Summary Table.

Figure 9-185. EMIF4D_SYSTEM_OCP_INTR_EN_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					EN_DNV_SYS	EN_TA_SYS	EN_ERR_SYS
R-0-0h					R/W1C-0h	R/W1C-0h	R/W1C-0h

Table 9-211. EMIF4D_SYSTEM_OCP_INTR_EN_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	
2	EN_DNV_SYS	R/W1C	0h	Enable clear for system OCP interrupt for LPDDR2 NVM data not valid. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.
1	EN_TA_SYS	R/W1C	0h	Enable clear for system OCP interrupt for SDRAM temperature alert. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.
0	EN_ERR_SYS	R/W1C	0h	Enable clear for system OCP interrupt for command and address error. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.

9.3.4.44 EMIF4D_LOW_LAT_OCP_INTR_EN_CLR Register (Offset = C0h) [reset = 0h]

EMIF4D_LOW_LAT_OCP_INTR_EN_CLR is shown in Figure 9-186 and described in Table 9-212.

Return to Summary Table.

Figure 9-186. EMIF4D_LOW_LAT_OCP_INTR_EN_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					EN_DNV_LL	EN_TA_LL	EN_ERR_LL
R-0-0h					R/W1C-0h	R/W1C-0h	R/W1C-0h

Table 9-212. EMIF4D_LOW_LAT_OCP_INTR_EN_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	
2	EN_DNV_LL	R/W1C	0h	Enable clear for low-latency OCP interrupt for LPDDR2 NVM data not valid. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.
1	EN_TA_LL	R/W1C	0h	Enable clear for low-latency OCP interrupt for SDRAM temperature alert. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.
0	EN_ERR_LL	R/W1C	0h	Enable clear for low-latency OCP interrupt for command and address error. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.

9.3.4.45 EMIF4D_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG Register (Offset = C8h) [reset = 0h]

EMIF4D_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG is shown in Figure 9-187 and described in Table 9-213.

Return to Summary Table.

Figure 9-187. EMIF4D_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG Register

31	30	29	28	27	26	25	24
ZQ_CS1EN	ZQ_CS0EN	ZQ_DUALCALEN	ZQ_SFEXITEN	RESERVED			
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0-0h			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				ZQ_ZQINIT_MULT		ZQ_ZQCL_MULT	
R-0-0h				R/W-0h		R/W-0h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ZQ_REFINTERVAL							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ZQ_REFINTERVAL							
R/W-0h							

Table 9-213. EMIF4D_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ZQ_CS1EN	R/W	0h	Writing a 1 enables ZQ calibration for CS1.
30	ZQ_CS0EN	R/W	0h	Writing a 1 enables ZQ calibration for CS0.
29	ZQ_DUALCALEN	R/W	0h	ZQ Dual Calibration enable. Allows both ranks to be ZQ calibrated simultaneously. Setting this bit requires both chip selects to have a seerate calibration resistor per device.
28	ZQ_SFEXITEN	R/W	0h	ZQCL on Self Refresh, Active Power-Down, and Precharge Power-Down exit enable. Writing a 1 enables the issuing of ZQCL on Self-Refresh, Active Power-Down, and Precharge Power-Down exit.
27-20	RESERVED	R-0	0h	
19-18	ZQ_ZQINIT_MULT	R/W	0h	Indicates the number of ZQCL intervals that make up a ZQINIT interval, minus one.
17-16	ZQ_ZQCL_MULT	R/W	0h	Indicates the number of ZQCS intervals that make up a ZQCL interval, minus one. ZQCS interval is defined by reg_zq_zqcs in SDRAM Timing 3 Register.
15-0	ZQ_REFINTERVAL	R/W	0h	Number of refresh periods between ZQCS commands. This field supports between one refresh period to 256 ms between ZQCS calibration commands. Refresh period is defined by reg_refresh_rate in SDRAM Refresh Control register.

9.3.4.46 EMIF4D_TEMPERATURE_ALERT_CONFIG Register (Offset = CCh) [reset = 0h]

EMIF4D_TEMPERATURE_ALERT_CONFIG is shown in Figure 9-188 and described in Table 9-214.

Return to Summary Table.

Figure 9-188. EMIF4D_TEMPERATURE_ALERT_CONFIG Register

31	30	29	28	27	26	25	24
TA_CS1EN	TA_CS0EN	RESERVED	TA_SFEXITEN	TA_DEVWDT		TA_DEVCNT	
R/W-0h	R/W-0h	R-0-0h	R/W-0h	R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
RESERVED		TA_REFINTERVAL					
R-0-0h		R/W-0h					
15	14	13	12	11	10	9	8
TA_REFINTERVAL							
R/W-0h							
7	6	5	4	3	2	1	0
TA_REFINTERVAL							
R/W-0h							

Table 9-214. EMIF4D_TEMPERATURE_ALERT_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TA_CS1EN	R/W	0h	Writing a 1 enables temperature alert polling for CS1.
30	TA_CS0EN	R/W	0h	Writing a 1 enables temperature alert polling for CS0.
29	RESERVED	R-0	0h	
28	TA_SFEXITEN	R/W	0h	Temperature Alert Poll on Self-Refresh, Active Power-Down, and Precharge Power-Down exit enable. Writing a 1 enables the issuing of a temperature alert poll on Self-Refresh exit.
27-26	TA_DEVWDT	R/W	0h	This field indicates how wide a physical device is. It is used in conjunction with the reg_ta_devcnt register to determine which byte lanes contain the tempetature alert info. A value of 0 = eight bit wide, 1 = sixteen bit wide, 2 = thirty two bit wide. All others are reserved. If this field is set to 1 and the reg_ta_devcnt field is set to one the byte mask for checking will be 4'b0101.
25-24	TA_DEVCNT	R/W	0h	This field indicates which external byte lanes contain a device for temperature monitoring. A value of 0 = one device, 1 = two devices, 2 = four devices. All other reserved.
23-22	RESERVED	R-0	0h	
21-0	TA_REFINTERVAL	R/W	0h	Number of refresh periods between temperature alert polls. This field supports one refresh period to 10 seconds between temperature alert polls. Refresh period is defined by reg_refresh_rate in SDRAM Refresh Control register.

9.3.4.47 EMIF4D_OCP_ERROR_LOG Register (Offset = D0h) [reset = 0h]

EMIF4D_OCP_ERROR_LOG is shown in Figure 9-189 and described in Table 9-215.

Return to Summary Table.

Figure 9-189. EMIF4D_OCP_ERROR_LOG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
MADDRSPACE		MBURSTSEQ			MCMD		
R-0h		R-0h			R-0h		
7	6	5	4	3	2	1	0
MCONNID							
R-0h							

Table 9-215. EMIF4D_OCP_ERROR_LOG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	
15-14	MADDRSPACE	R	0h	Address space of the first errored transaction. Writing a 1 to the lt field in the Interrupt Raw register or the lt_masked field in the Interrupt Masked register will clear this field.
13-11	MBURSTSEQ	R	0h	Addressing mode of the first errored transaction. Writing a 1 to the lt field in the Interrupt Raw register or the lt_masked field in the Interrupt Masked register will clear this field.
10-8	MCMD	R	0h	Command type of the first errored transaction. Writing a 1 to the lt field in the Interrupt Raw register or the lt_masked field in the Interrupt Masked register will clear this field.
7-0	MCONNID	R	0h	Connection ID of the first errored transaction. Writing a 1 to the lt field in the Interrupt Raw register or the lt_masked field in the Interrupt Masked register will clear this field.

9.3.4.48 EMIF4D_READ_WRITE_LEVELING_RAMP_WINDOW Register (Offset = D4h) [reset = 0h]

EMIF4D_READ_WRITE_LEVELING_RAMP_WINDOW is shown in Figure 9-190 and described in Table 9-216.

Return to Summary Table.

Figure 9-190. EMIF4D_READ_WRITE_LEVELING_RAMP_WINDOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED			RDWRLVLINC_RMP_WIN												
R-0-0h			R/W-0h												

Table 9-216. EMIF4D_READ_WRITE_LEVELING_RAMP_WINDOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R-0	0h	
12-0	RDWRLVLINC_RMP_WIN	R/W	0h	Incremental leveling ramp window in number of refresh periods. The value programmed is minus one the required value. Refresh period is defined by reg_refresh_rate in SDRAM Refresh Control register.

9.3.4.49 EMIF4D_READ_WRITE_LEVELING_RAMP_CTRL Register (Offset = D8h) [reset = 0h]

EMIF4D_READ_WRITE_LEVELING_RAMP_CTRL is shown in Figure 9-191 and described in Table 9-217.

Return to Summary Table.

Figure 9-191. EMIF4D_READ_WRITE_LEVELING_RAMP_CTRL Register

31	30	29	28	27	26	25	24
RDWRLVL_EN	RDWRLVLINC_RMP_PRE						
R/W-0h	R/W-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RDLVLINC_RMP_INT							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RDLVLGATEINC_RMP_INT							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WRLVLINC_RMP_INT							
R/W-0h							

Table 9-217. EMIF4D_READ_WRITE_LEVELING_RAMP_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RDWRLVL_EN	R/W	0h	Read-Write Leveling enable. Set 1 to enable leveling. Set 0 to disable leveling.
30-24	RDWRLVLINC_RMP_PRE	R/W	0h	Incremental leveling pre-scalar in number of refresh periods during ramp window. The value programmed is minus one the required value. Refresh period is defined by reg_refresh_rate in SDRAM Refresh Control register.
23-16	RDLVLINC_RMP_INT	R/W	0h	Incremental read data eye training interval during ramp window. Number of reg_rdwrlvlinc_rmp_pre intervals between incremental read data eye training during ramp window. A value of 0 will disable incremental read data eye training.
15-8	RDLVLGATEINC_RMP_INT	R/W	0h	Incremental read DQS gate training interval during ramp window. Number of reg_rdwrlvlinc_rmp_pre intervals between incremental read DQS gate training during ramp window. A value of 0 will disable incremental read DQS gate training.
7-0	WRLVLINC_RMP_INT	R/W	0h	Incremental write leveling interval during ramp window. Number of reg_rdwrlvlinc_rmp_pre intervals between incremental write leveling during ramp window. A value of 0 will disable incremental write leveling.

9.3.4.50 EMIF4D_READ_WRITE_LEVELING_CTRL Register (Offset = DCh) [reset = 0h]

EMIF4D_READ_WRITE_LEVELING_CTRL is shown in Figure 9-192 and described in Table 9-218.

Return to Summary Table.

Figure 9-192. EMIF4D_READ_WRITE_LEVELING_CTRL Register

31	30	29	28	27	26	25	24
RDWRLVLFULL_START	RDWRLVLINC_PRE						
R/W-0h	R/W-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RDLVLINC_INT							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RDLVLGATEINC_INT							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WRLVLINC_INT							
R/W-0h							

Table 9-218. EMIF4D_READ_WRITE_LEVELING_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RDWRLVLFULL_START	R/W	0h	Full leveling trigger. Writing a 1 to this field triggers full read and write leveling. This bit will self clear to 0.
30-24	RDWRLVLINC_PRE	R/W	0h	Incremental leveling pre-scalar in number of refresh periods. The value programmed is minus one the required value. Refresh period is defined by reg_refresh_rate in SDRAM Refresh Control register.
23-16	RDLVLINC_INT	R/W	0h	Incremental read data eye training interval. Number of reg_rdwrlvlinc_pre intervals between incremental read data eye training. A value of 0 will disable incremental read data eye training.
15-8	RDLVLGATEINC_INT	R/W	0h	Incremental read DQS gate training interval. Number of reg_rdwrlvlinc_pre intervals between incremental read DQS gate training. A value of 0 will disable incremental read DQS gate training.
7-0	WRLVLINC_INT	R/W	0h	Incremental write leveling interval. Number of reg_rdwrlvlinc_pre intervals between incremental write leveling. A value of 0 will disable incremental write leveling.

9.3.4.51 EMIF4D_DDR_PHY_CTRL_1 Register (Offset = E4h) [reset = 0h]

EMIF4D_DDR_PHY_CTRL_1 is shown in Figure 9-193 and described in Table 9-219.

Return to Summary Table.

Figure 9-193. EMIF4D_DDR_PHY_CTRL_1 Register

31	30	29	28	27	26	25	24
RESERVED				RDLVL_MASK	RDLVLGATE_MASK	WRLVL_MASK	RESERVED
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R-0-0h
23	22	21	20	19	18	17	16
RESERVED		PHY_HALF_DELAYS	PHY_CLK_STALL_LEVEL	PHY_DIS_CALIB_RST	PHY_INVERT_CLKOUT	PHY_DLL_LOCK_DIFF	
R-0-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
PHY_DLL_LOCK_DIFF						PHY_FAST_DLL_LOCK	RESERVED
R/W-0h						R/W-0h	R-0-0h
7	6	5	4	3	2	1	0
RESERVED			READ_LAT				
R-0-0h			R/W-0h				

Table 9-219. EMIF4D_DDR_PHY_CTRL_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R-0	0h	
27	RDLVL_MASK	R/W	0h	writing a 1 to this field will mask read data eye training during full leveling command, plus drives reg_phy_use_rd_data_eye_level control low to allow user to use programmed ratio values. Incremental training needs to be disabled using incremental training registers.
26	RDLVLGATE_MASK	R/W	0h	writing a 1 to this field will mask dqs gate training during full leveling command, plus drives reg_phy_use_rd_dqs_level control low to allow user to use programmed ratio values. Incremental training needs to be disabled using incremental training registers.
25	WRLVL_MASK	R/W	0h	writing a 1 to this field will mask write leveling training during full leveling command, plus drives reg_phy_use_wr_level control low to allow user to use programmed ratio values. Incremental training needs to be disabled using incremental training registers.
24-22	RESERVED	R-0	0h	
21	PHY_HALF_DELAYS	R/W	0h	0: Half Delay disabled, DDR DFI clock frequency = DDR DLL frequency 1: Half Delay enabled, DDR DFI clock frequency = 2*DDR DLL frequency
20	PHY_CLK_STALL_LEVEL	R/W	0h	Reserved. Always set to 0.
19	PHY_DIS_CALIB_RST	R/W	0h	Reserved. Always set to 0.
18	PHY_INVERT_CLKOUT	R/W	0h	Inverts the polarity of the DDR clock 0: clock not inverted 1: clock inverted Must be set to 0 for LPDDR2. For DDR3, can be set to 0 or 1 depending on board design (Setting of 1 will work for most board designs).
17-10	PHY_DLL_LOCK_DIFF	R/W	0h	Reserved. Always set to 0x20.
9	PHY_FAST_DLL_LOCK	R/W	0h	Reserved. Always set to 0.

Table 9-219. EMIF4D_DDR_PHY_CTRL_1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8-5	RESERVED	R-0	0h	
4-0	READ_LAT	R/W	0h	This field defines the read latency for the read data from SDRAM in number of DDR clock cycles. This field is used by the EMIF as well as the PHY. The EMIF will expect the first read data to arrive (reg_read_latency + 3) DDR clock cycles from the read command.

9.3.4.52 EMIF4D_DDR_PHY_CTRL_1_SHADOW Register (Offset = E8h) [reset = 0h]

EMIF4D_DDR_PHY_CTRL_1_SHADOW is shown in Figure 9-194 and described in Table 9-220.

Return to Summary Table.

Figure 9-194. EMIF4D_DDR_PHY_CTRL_1_SHADOW Register

31	30	29	28	27	26	25	24
RESERVED				RDLVL_MASK_SHDW	RDLVLGATE_MASK_SHDW	WRLVL_MASK_SHDW	RESERVED
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R-0-0h
23	22	21	20	19	18	17	16
RESERVED		PHY_HALF_DELAYS_SHDW	PHY_CLK_STALL_LEVEL_SHDW	PHY_DIS_CALIB_RST_SHDW	PHY_INVERT_CLKOUT_SHDW	PHY_DLL_LOCK_DIFF_SHDW	
R-0-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
PHY_DLL_LOCK_DIFF_SHDW						PHY_FAST_DLL_LOCK_SHDW	RESERVED
R/W-0h						R/W-0h	R-0-0h
7	6	5	4	3	2	1	0
RESERVED			READ_LAT_SHDW				
R-0-0h			R/W-0h				

Table 9-220. EMIF4D_DDR_PHY_CTRL_1_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R-0	0h	
27	RDLVL_MASK_SHDW	R/W	0h	writing a 1 to this field will mask read data eye training during full leveling command, plus drives reg_phy_use_rd_data_eye_level control low to allow user to use programmed ratio values. Incremental training needs to be disabled using incremental training registers.
26	RDLVLGATE_MASK_SHDW	R/W	0h	writing a 1 to this field will mask dqs gate training during full leveling command, plus drives reg_phy_use_rd_dqs_level control low to allow user to use programmed ratio values. Incremental training needs to be disabled using incremental training registers.
25	WRLVL_MASK_SHDW	R/W	0h	writing a 1 to this field will mask write leveling training during full leveling command, plus drives reg_phy_use_wr_level control low to allow user to use programmed ratio values. Incremental training needs to be disabled using incremental training registers.
24-22	RESERVED	R-0	0h	
21	PHY_HALF_DELAYS_SHDW	R/W	0h	0: Half Delay disabled, DDR DFI clock frequency = DDR DLL frequency 1: Half Delay enabled, DDR DFI clock frequency = 2*DDR DLL frequency
20	PHY_CLK_STALL_LEVEL_SHDW	R/W	0h	Reserved. Always set to 0.
19	PHY_DIS_CALIB_RST_SHDW	R/W	0h	Reserved. Always set to 0.
18	PHY_INVERT_CLKOUT_SHDW	R/W	0h	Inverts the polarity of the DDR clock 0: clock not inverted 1: clock inverted Must be set to 0 for LPDDR2. For DDR3, can be set to 0 or 1 depending on board design (Setting of 1 will work for most board designs).
17-10	PHY_DLL_LOCK_DIFF_SHDW	R/W	0h	Reserved. Always set to 0x20.

Table 9-220. EMIF4D_DDR_PHY_CTRL_1_SHADOW Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	PHY_FAST_DLL_LOCK_SHDW	R/W	0h	Reserved. Always set to 0.
8-5	RESERVED	R-0	0h	
4-0	READ_LAT_SHDW	R/W	0h	Shadow field for reg_read_latency. This field is loaded into reg_read_latency field in DDR PHY Control 1 register when SIdleAck is asserted.

9.3.4.53 EMIF4D_PRIORITY_TO_CLASS_OF_SERVICE_MAPPING Register (Offset = 100h) [reset = 0h]

EMIF4D_PRIORITY_TO_CLASS_OF_SERVICE_MAPPING is shown in Figure 9-195 and described in Table 9-221.

Return to Summary Table.

Figure 9-195. EMIF4D_PRIORITY_TO_CLASS_OF_SERVICE_MAPPING Register

31	30	29	28	27	26	25	24
PRI_COS_MAP_EN	RESERVED						
R/W-0h	R-0-0h						
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
PRI_7_COS		PRI_6_COS		PRI_5_COS		PRI_4_COS	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
PRI_3_COS		PRI_2_COS		PRI_1_COS		PRI_0_COS	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-221. EMIF4D_PRIORITY_TO_CLASS_OF_SERVICE_MAPPING Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PRI_COS_MAP_EN	R/W	0h	Set 1 to enable priority to class of service mapping. Set 0 to disable mapping.
30-16	RESERVED	R-0	0h	
15-14	PRI_7_COS	R/W	0h	Class of service for commands with priority of 7. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.
13-12	PRI_6_COS	R/W	0h	Class of service for commands with priority of 6. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.
11-10	PRI_5_COS	R/W	0h	Class of service for commands with priority of 5. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.
9-8	PRI_4_COS	R/W	0h	Class of service for commands with priority of 4. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.
7-6	PRI_3_COS	R/W	0h	Class of service for commands with priority of 3. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.
5-4	PRI_2_COS	R/W	0h	Class of service for commands with priority of 2. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.
3-2	PRI_1_COS	R/W	0h	Class of service for commands with priority of 1. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.
1-0	PRI_0_COS	R/W	0h	Class of service for commands with priority of 0. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.

9.3.4.54 EMIF4D_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING Register (Offset = 104h) [reset = 0h]

EMIF4D_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING is shown in Figure 9-196 and described in Table 9-222.

Return to Summary Table.

Figure 9-196. EMIF4D_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>CONNID_COS_1_MAP_EN</td><td colspan="7">CONNID_1_COS</td></tr>
<tr><td>R/W-0h</td><td colspan="7">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>CONNID_1_COS</td><td colspan="3">MSK_1_COS</td><td colspan="4">CONNID_2_COS_1</td></tr>
<tr><td>R/W-0h</td><td colspan="3">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="4">CONNID_2_COS_1</td><td colspan="2">MSK_2_COS_1</td><td colspan="2">CONNID_3_COS_1</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">CONNID_3_COS_1</td><td colspan="2">MSK_3_COS_1</td></tr>
<tr><td colspan="6">R/W-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 9-222. EMIF4D_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CONNID_COS_1_MAP_EN	R/W	0h	Set 1 to enable Connection ID to class of service 1 mapping. Set 0 to disable mapping.
30-23	CONNID_1_COS	R/W	0h	Connection ID value 1 for class of service 1.
22-20	MSK_1_COS	R/W	0h	Mask for Connection ID value 1 for class of service 1. Value of 0 will disable masking. Value of 1 will mask Connection ID bit 0. Value of 2 will mask Connection ID bits 1:0. Value of 3 will mask Connection ID bits 2:0. Value of 4 will mask Connection ID bits 3:0. Value of 5 will mask Connection ID bits 4:0. Value of 6 will mask Connection ID bits 5:0. Value of 7 will mask Connection ID bits 6:0.
19-12	CONNID_2_COS_1	R/W	0h	Connection ID value 2 for class of service 1.
11-10	MSK_2_COS_1	R/W	0h	Mask for Connection ID value 2 for class of service 1. Value of 0 will disable masking. Value of 1 will mask Connection ID bit 0. Value of 2 will mask Connection ID bits 1:0. Value of 3 will mask Connection ID bits 2:0.
9-2	CONNID_3_COS_1	R/W	0h	Connection ID value 3 for class of service 1.
1-0	MSK_3_COS_1	R/W	0h	Mask for Connection ID value 3 for class of service 1. Value of 0 will disable masking. Value of 1 will mask Connection ID bit 0. Value of 2 will mask Connection ID bits 1:0. Value of 3 will mask Connection ID bits 2:0.

9.3.4.55 EMIF4D_CONNECTION_ID_TO_CLASS_OF_SERVICE_2_MAPPING Register (Offset = 108h) [reset = 0h]

EMIF4D_CONNECTION_ID_TO_CLASS_OF_SERVICE_2_MAPPING is shown in Figure 9-197 and described in Table 9-223.

Return to Summary Table.

Figure 9-197. EMIF4D_CONNECTION_ID_TO_CLASS_OF_SERVICE_2_MAPPING Register

31	30	29	28	27	26	25	24
CONNID_COS_2_MAP_EN	CONNID_1_COS_2						
R/W-0h	R/W-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CONNID_1_COS_2	MSK_1_COS_2			CONNID_2_COS			
R/W-0h	R/W-0h			R/W-0h			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CONNID_2_COS				MSK_2_COS		CONNID_3_COS_2	
R/W-0h				R/W-0h		R/W-0h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CONNID_3_COS_2						MSK_3_COS_2	
R/W-0h						R/W-0h	

Table 9-223. EMIF4D_CONNECTION_ID_TO_CLASS_OF_SERVICE_2_MAPPING Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CONNID_COS_2_MAP_EN	R/W	0h	Set 1 to enable Connection ID to class of service 2 mapping. Set 0 to disable mapping.
30-23	CONNID_1_COS_2	R/W	0h	Connection ID value 1 for class of service 2.
22-20	MSK_1_COS_2	R/W	0h	Mask for Connection ID value 1 for class of service 2. Value of 0 will disable masking. Value of 1 will mask Connection ID bit 0. Value of 2 will mask Connection ID bits 1:0. Value of 3 will mask Connection ID bits 2:0. Value of 4 will mask Connection ID bits 3:0. Value of 5 will mask Connection ID bits 4:0. Value of 6 will mask Connection ID bits 5:0. Value of 7 will mask Connection ID bits 6:0.
19-12	CONNID_2_COS	R/W	0h	Connection ID value 2 for class of service 2.
11-10	MSK_2_COS	R/W	0h	Mask for Connection ID value 2 for class of service 2. Value of 0 will disable masking. Value of 1 will mask Connection ID bit 0. Value of 2 will mask Connection ID bits 1:0. Value of 3 will mask Connection ID bits 2:0.
9-2	CONNID_3_COS_2	R/W	0h	Connection ID value 3 for class of service 2.
1-0	MSK_3_COS_2	R/W	0h	Mask for Connection ID value 3 for class of service 2. Value of 0 will disable masking. Value of 1 will mask Connection ID bit 0. Value of 2 will mask Connection ID bits 1:0. Value of 3 will mask Connection ID bits 2:0.

9.3.4.56 EMIF4D_ECC_CTRL_REG Register (Offset = 110h) [reset = 0h]

EMIF4D_ECC_CTRL_REG is shown in Figure 9-198 and described in Table 9-224.

Return to Summary Table.

Figure 9-198. EMIF4D_ECC_CTRL_REG Register

31	30	29	28	27	26	25	24
REG_ECC_EN	REG_ECC_ADDR_RGN_PROT	RESERVED					
R/W-0h	R/W-0h	R-0-0h					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						REG_ECC_ADDR_RGN_2_EN	REG_ECC_ADDR_RGN_1_EN
R-0-0h						R/W-0h	R/W-0h

Table 9-224. EMIF4D_ECC_CTRL_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	REG_ECC_EN	R/W	0h	Set 1 to enable ECC. Set 0 to disable ECC.
30	REG_ECC_ADDR_RGN_PROT	R/W	0h	Setting this field to 1 and reg_ecc_en to a 1 will enable ECC calculation for accesses within the address ranges and disable ECC calculation for accesses outside the address ranges. Setting this field to 0 and reg_ecc_en to a 1 will disable ECC calculation for accesses within the address ranges and enable ECC calculation for accesses outside the address ranges. The address ranges can be specified using the ECC Address Range 1 and 2 registers.
29-2	RESERVED	R-0	0h	
1	REG_ECC_ADDR_RGN_2_EN	R/W	0h	Set 1 to enable ECC address range 2. Set 0 to disable ECC address range 2.
0	REG_ECC_ADDR_RGN_1_EN	R/W	0h	Set 1 to enable ECC address range 1. Set 0 to disable ECC address range 1.

9.3.4.57 EMIF4D_ECC_ADDR_RANGE_1 Register (Offset = 114h) [reset = 0h]

EMIF4D_ECC_ADDR_RANGE_1 is shown in Figure 9-199 and described in Table 9-225.

Return to Summary Table.

Figure 9-199. EMIF4D_ECC_ADDR_RANGE_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REG_ECC_END_ADDR_1															
R/W-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_ECC_STRT_ADDR_1															
R/W-0h															

Table 9-225. EMIF4D_ECC_ADDR_RANGE_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	REG_ECC_END_ADDR_1	R/W	0h	End caddress [32:17] for ECC address range 1
15-0	REG_ECC_STRT_ADDR_1	R/W	0h	Start caddress [32:17] for ECC address range 1

9.3.4.58 EMIF4D_ECC_ADDR_RANGE_2 Register (Offset = 118h) [reset = 0h]

EMIF4D_ECC_ADDR_RANGE_2 is shown in Figure 9-200 and described in Table 9-226.

Return to Summary Table.

Figure 9-200. EMIF4D_ECC_ADDR_RANGE_2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REG_ECC_END_ADDR_2															
R/W-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_ECC_STRT_ADDR_2															
R/W-0h															

Table 9-226. EMIF4D_ECC_ADDR_RANGE_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	REG_ECC_END_ADDR_2	R/W	0h	End caddress [32:17] for ECC address range 2
15-0	REG_ECC_STRT_ADDR_2	R/W	0h	Start caddress [32:17] for ECC address range 2

9.3.4.59 EMIF4D_READ_WRITE_EXECUTION_THR Register (Offset = 120h) [reset = C5h]

EMIF4D_READ_WRITE_EXECUTION_THR is shown in Figure 9-201 and described in Table 9-227.

Return to Summary Table.

Figure 9-201. EMIF4D_READ_WRITE_EXECUTION_THR Register

31	30	29	28	27	26	25	24
MFLAG_OVERRIDE	EN_LLBUBBLE	RESERVED					
R/W-0h	R/W-0h	R-0-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED			WR_THRSH				
R-0-0h			R/W-0h				
7	6	5	4	3	2	1	0
RESERVED			RD_THRSH				
R-0-6h			R/W-5h				

Table 9-227. EMIF4D_READ_WRITE_EXECUTION_THR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MFLAG_OVERRIDE	R/W	0h	0 = Use MFLAG, 1 = Use Class of Service
30	EN_LLBUBBLE	R/W	0h	0 = No LL Bubble, 1 = LL Bubble Enabled.
29-13	RESERVED	R-0	0h	
12-8	WR_THRSH	R/W	0h	Write Threshold. Number of SDRAM write bursts after which the EMIF arbitration will switch to executing read commands. The value programmed is always minus one the required number.
7-5	RESERVED	R-0	6h	
4-0	RD_THRSH	R/W	5h	Read Threshold. Number of SDRAM read bursts after which the EMIF arbitration will switch to executing write commands. The value programmed is always minus one the required number.

9.3.4.60 EMIF4D_COS_CONFIG Register (Offset = 124h) [reset = 00FFFFFFh]

EMIF4D_COS_CONFIG is shown in Figure 9-202 and described in Table 9-228.

Return to Summary Table.

Figure 9-202. EMIF4D_COS_CONFIG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								COS_COUNT_1							
R-0-0h								R/W-FFh							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COS_COUNT_2								PR_OLD_COUNT							
R/W-FFh								R/W-FFh							

Table 9-228. EMIF4D_COS_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R-0	0h	
23-16	COS_COUNT_1	R/W	FFh	Priority Raise Counter for class of service 1. Number of m_clk cycles after which the EMIF momentarily raises the priority of the class of service 1 commands in the Command FIFO. A value of N will be equal to N x 16 clocks.
15-8	COS_COUNT_2	R/W	FFh	Priority Raise Counter for class of service 2. Number of m_clk cycles after which the EMIF momentarily raises the priority of the class of service 2 commands in the Command FIFO. A value of N will be equal to N x 16 clocks.
7-0	PR_OLD_COUNT	R/W	FFh	Priority Raise Old Counter. Number of m_clk cycles after which the EMIF momentarily raises the priority of the oldest command in the Command FIFO. A value of N will be equal to N x 16 clocks.

9.3.4.61 EMIF4D_1B_ECC_ERR_CNT Register (Offset = 130h) [reset = 0h]

EMIF4D_1B_ECC_ERR_CNT is shown in Figure 9-203 and described in Table 9-229.

Return to Summary Table.

Figure 9-203. EMIF4D_1B_ECC_ERR_CNT Register

31–0
REG_1B_ECC_ERR_CNT
R/W-0h

Table 9-229. EMIF4D_1B_ECC_ERR_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REG_1B_ECC_ERR_CNT	R/W	0h	32 bit counter that displays number of 1 bit ECC errors. Writing a value will decrement the count by that value. For example, if the count is 0x1234_ABF3, writing 0x1234_ABF3 to this register will clear it.

9.3.4.62 EMIF4D_1B_ECC_ERR_THRSH Register (Offset = 134h) [reset = 0h]

EMIF4D_1B_ECC_ERR_THRSH is shown in Figure 9-204 and described in Table 9-230.

Return to Summary Table.

Figure 9-204. EMIF4D_1B_ECC_ERR_THRSH Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">REG_1B_ECC_ERR_THRSH</td><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td><td colspan="8">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">REG_1B_ECC_ERR_WIN</td></tr>
<tr><td colspan="16">R/W-0h</td></tr>
</table>

Table 9-230. EMIF4D_1B_ECC_ERR_THRSH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	REG_1B_ECC_ERR_THRSH	R/W	0h	1 bit ECC error threshold. The EMIF will generate an interrupt when the 1 bit ECC error count is greater than or equal to this threshold. A value of 0 will disable the generation of the interrupt.
23-16	RESERVED	R-0	0h	
15-0	REG_1B_ECC_ERR_WIN	R/W	0h	1 bit ECC error window in number of refresh periods. The EMIF will generate an interrupt when the 1 bit ECC error count is equal to or greater than the threshold within this window. A value of 0 will disable the window. Refresh period is defined by reg_refresh_rate in SDRAM Refresh Control register. The software can set this bitfield to 0x0 to reset the internal counter.

9.3.4.63 EMIF4D_1B_ECC_ERR_DIST_1 Register (Offset = 138h) [reset = 0h]

EMIF4D_1B_ECC_ERR_DIST_1 is shown in Figure 9-205 and described in Table 9-231.

Return to Summary Table.

Figure 9-205. EMIF4D_1B_ECC_ERR_DIST_1 Register

31-0
REG_1B_ECC_ERR_DIST_1
R/W-0h

Table 9-231. EMIF4D_1B_ECC_ERR_DIST_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REG_1B_ECC_ERR_DIST_1	R/W	0h	1 bit ECC error distribution over data bus bit 31:0. A value of 1 on a bit indicates 1 bit error on the corresponding bit on the data bus. Writing a 1 to any bit will clear that bit. Writing a 0 has no effect.

9.3.4.64 EMIF4D_1B_ECC_ERR_ADDR_LOG Register (Offset = 13Ch) [reset = 0h]

EMIF4D_1B_ECC_ERR_ADDR_LOG is shown in Figure 9-206 and described in Table 9-232.

Return to Summary Table.

Figure 9-206. EMIF4D_1B_ECC_ERR_ADDR_LOG Register

31-0
REG_1B_ECC_ERR_ADDR
R/W-0h

Table 9-232. EMIF4D_1B_ECC_ERR_ADDR_LOG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REG_1B_ECC_ERR_ADDR	R/W	0h	1 bit ECC error address. Most significant bits of the starting address(es) related to the SDRAM reads that had a 1 bit ECC error. This field displays up to four addresses logged in the 4 deep address logging FIFO. Writing a 0x1 will pop one element of the FIFO. Writing a 0x2 will pop all elements of the FIFO. Writing any other alue will have no effect.

9.3.4.65 EMIF4D_2B_ECC_ERR_ADDR_LOG Register (Offset = 140h) [reset = 0h]

EMIF4D_2B_ECC_ERR_ADDR_LOG is shown in Figure 9-207 and described in Table 9-233.

Return to Summary Table.

Figure 9-207. EMIF4D_2B_ECC_ERR_ADDR_LOG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_2B_ECC_ERR_ADDR																															
R/W-0h																															

Table 9-233. EMIF4D_2B_ECC_ERR_ADDR_LOG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REG_2B_ECC_ERR_ADDR	R/W	0h	2 bit ECC error address. Most significant bits of the starting address of the first SDRAM burst that had the 2 bit ECC error. Writing a 1 will clear this field. Writing any other value has no effect.

9.3.4.66 EMIF4D_PHY_STS_1 Register (Offset = 144h) [reset = 0h]

EMIF4D_PHY_STS_1 is shown in Figure 9-208 and described in Table 9-234.

Return to Summary Table.

Figure 9-208. EMIF4D_PHY_STS_1 Register

31	30	29	28	27	26	25	24
RESERVED		PHY_REG_CTRL_DLL_SLAVE_VALUE					
R-0-0h		R-0h					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PHY_REG_CTRL_DLL_SLAVE_VALUE							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PHY_REG_CTRL_DLL_SLAVE_VALUE				RESERVED			PHY_REG_STS_DLL_LOCK
R-0h				R-0-0h			R-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PHY_REG_STS_DLL_LOCK				RESERVED		PHY_REG_CTRL_DLL_LOCK	
R-0h				R-0-0h		R-0h	

Table 9-234. EMIF4D_PHY_STS_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R-0	0h	
29-12	PHY_REG_CTRL_DLL_SLAVE_VALUE	R	0h	DLL Slave Value
11-9	RESERVED	R-0	0h	
8-4	PHY_REG_STS_DLL_LOCK	R	0h	Lock Status for Data DLLs
3-2	RESERVED	R-0	0h	
1-0	PHY_REG_CTRL_DLL_LOCK	R	0h	Lock Status for Command DLLs

9.3.4.67 EMIF4D_PHY_STS_2 Register (Offset = 148h) [reset = 0h]

EMIF4D_PHY_STS_2 is shown in Figure 9-209 and described in Table 9-235.

Return to Summary Table.

Figure 9-209. EMIF4D_PHY_STS_2 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">PHY_REG_STS_DLL_SLAVE_VALUE_LO</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 9-235. EMIF4D_PHY_STS_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PHY_REG_STS_DLL_SLAVE_VALUE_LO	R	0h	Bits 31:0 of Phy_reg_status_dll_slave_value

9.3.4.68 EMIF4D_PHY_STS_3 Register (Offset = 14Ch) [reset = 0h]

EMIF4D_PHY_STS_3 is shown in Figure 9-210 and described in Table 9-236.

Return to Summary Table.

Figure 9-210. EMIF4D_PHY_STS_3 Register

31	30	29	28	27	26	25	24
RESERVED	PHY_REG_RDFIFO_RDPTR						
R-0-0h	R-0h						
23	22	21	20	19	18	17	16
PHY_REG_RDFIFO_RDPTR							
R-0h							
15	14	13	12	11	10	9	8
RESERVED			PHY_REG_STS_DLL_SLAVE_VALUE_HI				
R-0-0h			R-0h				
7	6	5	4	3	2	1	0
PHY_REG_STS_DLL_SLAVE_VALUE_HI							
R-0h							

Table 9-236. EMIF4D_PHY_STS_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0	0h	
30-16	PHY_REG_RDFIFO_RDPTR	R	0h	Read FIFO Read Pointer
15-13	RESERVED	R-0	0h	
12-0	PHY_REG_STS_DLL_SLAVE_VALUE_HI	R	0h	Bits 44:32 of Phy_reg_status_dll_slave_value

9.3.4.69 EMIF4D_PHY_STS_4 Register (Offset = 150h) [reset = 0h]

EMIF4D_PHY_STS_4 is shown in Figure 9-211 and described in Table 9-237.

Return to Summary Table.

Figure 9-211. EMIF4D_PHY_STS_4 Register

31	30	29	28	27	26	25	24
RESERVED	PHY_REG_GATELVL_FSM						
R-0-0h	R-0h						

23	22	21	20	19	18	17	16
PHY_REG_GATELVL_FSM							
R-0h							

15	14	13	12	11	10	9	8
RESERVED	PHY_REG_RDFIFO_WRPTR						
R-0-0h	R-0h						

7	6	5	4	3	2	1	0
PHY_REG_RDFIFO_WRPTR							
R-0h							

Table 9-237. EMIF4D_PHY_STS_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0	0h	
30-16	PHY_REG_GATELVL_FSM	R	0h	Gate Levelling FSM
15	RESERVED	R-0	0h	
14-0	PHY_REG_RDFIFO_WRPTR	R	0h	Read FIFO Write Pointer

9.3.4.70 EMIF4D_PHY_STS_5 Register (Offset = 154h) [reset = 0h]

EMIF4D_PHY_STS_5 is shown in Figure 9-212 and described in Table 9-238.

Return to Summary Table.

Figure 9-212. EMIF4D_PHY_STS_5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												PHY_REG_RD_LEVEL_FSM																			
R-0-0h												R-0h																			

Table 9-238. EMIF4D_PHY_STS_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	
19-0	PHY_REG_RD_LEVEL_FSM	R	0h	Read Levelling FSM

9.3.4.71 EMIF4D_PHY_STS_6 Register (Offset = 158h) [reset = 0h]

EMIF4D_PHY_STS_6 is shown in Figure 9-213 and described in Table 9-239.

Return to Summary Table.

Figure 9-213. EMIF4D_PHY_STS_6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	PHY_REG_WR_LEVEL_FSM														
R-0-0h	R-0h														

Table 9-239. EMIF4D_PHY_STS_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R-0	0h	
14-0	PHY_REG_WR_LEVEL_FSM	R	0h	Writel Levelling FSM

9.3.4.72 EMIF4D_PHY_STS_7 Register (Offset = 15Ch) [reset = 0h]

EMIF4D_PHY_STS_7 is shown in Figure 9-214 and described in Table 9-240.

Return to Summary Table.

Figure 9-214. EMIF4D_PHY_STS_7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_RDLVL_DQS_RATIO1									
R-0-0h						R-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PHY_REG_RDLVL_DQS_RATIO0									
R-0-0h						R-0h									

Table 9-240. EMIF4D_PHY_STS_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_RDLVL_DQS_RATIO1	R	0h	Read levelling DQS ratio1
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_RDLVL_DQS_RATIO0	R	0h	Read levelling DQS ratio0

9.3.4.73 EMIF4D_PHY_STS_8 Register (Offset = 160h) [reset = 0h]

EMIF4D_PHY_STS_8 is shown in Figure 9-215 and described in Table 9-241.

Return to Summary Table.

Figure 9-215. EMIF4D_PHY_STS_8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_RDLVL_DQS_RATIO3									
R-0-0h						R-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PHY_REG_RDLVL_DQS_RATIO2									
R-0-0h						R-0h									

Table 9-241. EMIF4D_PHY_STS_8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_RDLVL_DQS_RATIO3	R	0h	Read levelling DQS ratio3
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_RDLVL_DQS_RATIO2	R	0h	Read levelling DQS ratio2

9.3.4.74 EMIF4D_PHY_STS_9 Register (Offset = 164h) [reset = 0h]

EMIF4D_PHY_STS_9 is shown in Figure 9-216 and described in Table 9-242.

Return to Summary Table.

Figure 9-216. EMIF4D_PHY_STS_9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_RDLVL_DQS_RATIO5									
R-0-0h						R-0h									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PHY_REG_RDLVL_DQS_RATIO4									
R-0-0h						R-0h									

Table 9-242. EMIF4D_PHY_STS_9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_RDLVL_DQS_RATIO5	R	0h	Read Levelling DQS ratio5
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_RDLVL_DQS_RATIO4	R	0h	Read Levelling DQS ratio4

9.3.4.75 EMIF4D_PHY_STS_10 Register (Offset = 168h) [reset = 0h]

EMIF4D_PHY_STS_10 is shown in Figure 9-217 and described in Table 9-243.

Return to Summary Table.

Figure 9-217. EMIF4D_PHY_STS_10 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_RDLVL_DQS_RATIO7									
R-0-0h						R-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PHY_REG_RDLVL_DQS_RATIO6									
R-0-0h						R-0h									

Table 9-243. EMIF4D_PHY_STS_10 Register Field Descriptions

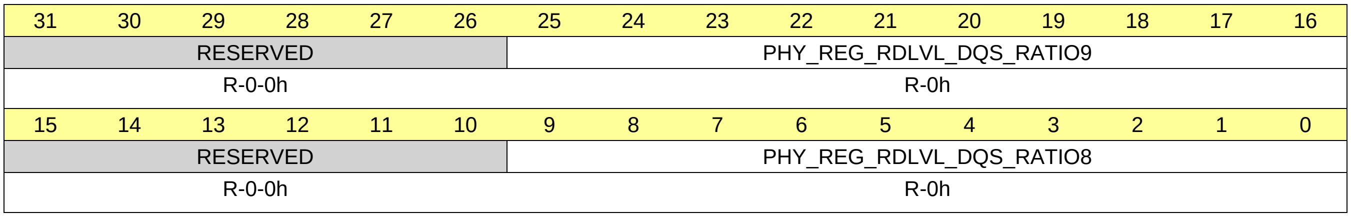
Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_RDLVL_DQS_RATIO7	R	0h	Read levelling DQS ratio7
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_RDLVL_DQS_RATIO6	R	0h	Read levelling DQS ratio6

9.3.4.76 EMIF4D_PHY_STS_11 Register (Offset = 16Ch) [reset = 0h]

EMIF4D_PHY_STS_11 is shown in Figure 9-218 and described in Table 9-244.

Return to Summary Table.

Figure 9-218. EMIF4D_PHY_STS_11 Register



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_RDLVL_DQS_RATIO9									
R-0-0h						R-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PHY_REG_RDLVL_DQS_RATIO8									
R-0-0h						R-0h									

Table 9-244. EMIF4D_PHY_STS_11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_RDLVL_DQS_RATIO9	R	0h	Read levelling DQS ratio9
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_RDLVL_DQS_RATIO8	R	0h	Read levelling DQS ratio8

9.3.4.77 EMIF4D_PHY_STS_12 Register (Offset = 170h) [reset = 0h]

EMIF4D_PHY_STS_12 is shown in Figure 9-219 and described in Table 9-245.

Return to Summary Table.

Figure 9-219. EMIF4D_PHY_STS_12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					PHY_REG_RDLVL_FIFOWEIN_RATIO1										
R-0-0h					R-0h										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PHY_REG_RDLVL_FIFOWEIN_RATIO0										
R-0-0h					R-0h										

Table 9-245. EMIF4D_PHY_STS_12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	PHY_REG_RDLVL_FIFOWEIN_RATIO1	R	0h	Read levelling FIFO Write Enable Ratio1
15-11	RESERVED	R-0	0h	
10-0	PHY_REG_RDLVL_FIFOWEIN_RATIO0	R	0h	Read levelling FIFO Write Enable Ratio0

9.3.4.78 EMIF4D_PHY_STS_13 Register (Offset = 174h) [reset = 0h]

EMIF4D_PHY_STS_13 is shown in Figure 9-220 and described in Table 9-246.

Return to Summary Table.

Figure 9-220. EMIF4D_PHY_STS_13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					PHY_REG_RDLVL_FIFOWEIN_RATIO3										
R-0-0h					R-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					PHY_REG_RDLVL_FIFOWEIN_RATIO2										
R-0-0h					R-0h										

Table 9-246. EMIF4D_PHY_STS_13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	PHY_REG_RDLVL_FIFOWEIN_RATIO3	R	0h	Read levelling FIFO Write Enable Ratio3
15-11	RESERVED	R-0	0h	
10-0	PHY_REG_RDLVL_FIFOWEIN_RATIO2	R	0h	Read levelling FIFO Write Enable Ratio2

9.3.4.79 EMIF4D_PHY_STS_14 Register (Offset = 178h) [reset = 0h]

EMIF4D_PHY_STS_14 is shown in Figure 9-221 and described in Table 9-247.

Return to Summary Table.

Figure 9-221. EMIF4D_PHY_STS_14 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="5">RESERVED</td><td colspan="11">PHY_REG_RDLVL_FIFOWEIN_RATIO5</td></tr>
<tr><td colspan="5">R-0-0h</td><td colspan="11">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">RESERVED</td><td colspan="11">PHY_REG_RDLVL_FIFOWEIN_RATIO4</td></tr>
<tr><td colspan="5">R-0-0h</td><td colspan="11">R-0h</td></tr>
</table>

Table 9-247. EMIF4D_PHY_STS_14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	PHY_REG_RDLVL_FIFOWEIN_RATIO5	R	0h	Read levelling FIFO Write Enable Ratio5
15-11	RESERVED	R-0	0h	
10-0	PHY_REG_RDLVL_FIFOWEIN_RATIO4	R	0h	Read levelling FIFO Write Enable Ratio4

9.3.4.80 EMIF4D_PHY_STS_15 Register (Offset = 17Ch) [reset = 0h]

EMIF4D_PHY_STS_15 is shown in Figure 9-222 and described in Table 9-248.

Return to Summary Table.

Figure 9-222. EMIF4D_PHY_STS_15 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					PHY_REG_RDLVL_FIFOWEIN_RATIO7										
R-0-0h					R-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					PHY_REG_RDLVL_FIFOWEIN_RATIO6										
R-0-0h					R-0h										

Table 9-248. EMIF4D_PHY_STS_15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	PHY_REG_RDLVL_FIFOWEIN_RATIO7	R	0h	Read levelling FIFO Wrie Enable Ratio7
15-11	RESERVED	R-0	0h	
10-0	PHY_REG_RDLVL_FIFOWEIN_RATIO6	R	0h	Read levelling FIFO Wrie Enable Ratio6

9.3.4.81 EMIF4D_PHY_STS_16 Register (Offset = 180h) [reset = 0h]

EMIF4D_PHY_STS_16 is shown in Figure 9-223 and described in Table 9-249.

Return to Summary Table.

Figure 9-223. EMIF4D_PHY_STS_16 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					PHY_REG_RDLVL_FIFOWEIN_RATIO9										
R-0-0h					R-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					PHY_REG_RDLVL_FIFOWEIN_RATIO8										
R-0-0h					R-0h										

Table 9-249. EMIF4D_PHY_STS_16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	PHY_REG_RDLVL_FIFOWEIN_RATIO9	R	0h	Read levelling FIFO Write Enable Ratio9
15-11	RESERVED	R-0	0h	
10-0	PHY_REG_RDLVL_FIFOWEIN_RATIO8	R	0h	Read levelling FIFO Write Enable Ratio8

9.3.4.82 EMIF4D_PHY_STS_17 Register (Offset = 184h) [reset = 0h]

EMIF4D_PHY_STS_17 is shown in Figure 9-224 and described in Table 9-250.

Return to Summary Table.

Figure 9-224. EMIF4D_PHY_STS_17 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WRLVL_DQ_RATIO1									
R-0-0h						R-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PHY_REG_WRLVL_DQ_RATIO0									
R-0-0h						R-0h									

Table 9-250. EMIF4D_PHY_STS_17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WRLVL_DQ_RATIO1	R	0h	Write levelling DQ ratio1
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WRLVL_DQ_RATIO0	R	0h	Write levelling DQ ratio0

9.3.4.83 EMIF4D_PHY_STS_18 Register (Offset = 188h) [reset = 0h]

EMIF4D_PHY_STS_18 is shown in Figure 9-225 and described in Table 9-251.

Return to Summary Table.

Figure 9-225. EMIF4D_PHY_STS_18 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WRLVL_DQ_RATIO3</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WRLVL_DQ_RATIO2</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R-0h</td></tr>
</table>

Table 9-251. EMIF4D_PHY_STS_18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WRLVL_DQ_RATIO3	R	0h	Write levelling DQ ratio3
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WRLVL_DQ_RATIO2	R	0h	Write levelling DQ ratio2

9.3.4.84 EMIF4D_PHY_STS_19 Register (Offset = 18Ch) [reset = 0h]

EMIF4D_PHY_STS_19 is shown in Figure 9-226 and described in Table 9-252.

Return to Summary Table.

Figure 9-226. EMIF4D_PHY_STS_19 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WRLVL_DQ_RATIO5									
R-0-0h						R-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PHY_REG_WRLVL_DQ_RATIO4									
R-0-0h						R-0h									

Table 9-252. EMIF4D_PHY_STS_19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WRLVL_DQ_RATIO5	R	0h	Write levelling DQ ratio5
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WRLVL_DQ_RATIO4	R	0h	Write levelling DQ ratio4

9.3.4.85 EMIF4D_PHY_STS_20 Register (Offset = 190h) [reset = 0h]

EMIF4D_PHY_STS_20 is shown in Figure 9-227 and described in Table 9-253.

Return to Summary Table.

Figure 9-227. EMIF4D_PHY_STS_20 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WRLVL_DQ_RATIO7</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WRLVL_DQ_RATIO6</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R-0h</td></tr>
</table>

Table 9-253. EMIF4D_PHY_STS_20 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WRLVL_DQ_RATIO7	R	0h	Write levelling DQ ratio7
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WRLVL_DQ_RATIO6	R	0h	Write levelling DQ ratio6

9.3.4.86 EMIF4D_PHY_STS_21 Register (Offset = 194h) [reset = 0h]

EMIF4D_PHY_STS_21 is shown in Figure 9-228 and described in Table 9-254.

Return to Summary Table.

Figure 9-228. EMIF4D_PHY_STS_21 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WRLVL_DQ_RATIO9</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WRLVL_DQ_RATIO8</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R-0h</td></tr>
</table>

Table 9-254. EMIF4D_PHY_STS_21 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WRLVL_DQ_RATIO9	R	0h	Write levelling DQ ratio9
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WRLVL_DQ_RATIO8	R	0h	Write levelling DQ ratio8

9.3.4.87 EMIF4D_PHY_STS_22 Register (Offset = 198h) [reset = 0h]

EMIF4D_PHY_STS_22 is shown in Figure 9-229 and described in Table 9-255.

Return to Summary Table.

Figure 9-229. EMIF4D_PHY_STS_22 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WRLVL_DQS_RATIO1</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WRLVL_DQS_RATIO0</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R-0h</td></tr>
</table>

Table 9-255. EMIF4D_PHY_STS_22 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WRLVL_DQS_RATIO1	R	0h	Write levelling DQS ratio 1
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WRLVL_DQS_RATIO0	R	0h	Write levelling DQS ratio 0

9.3.4.88 EMIF4D_PHY_STS_23 Register (Offset = 19Ch) [reset = 0h]

EMIF4D_PHY_STS_23 is shown in Figure 9-230 and described in Table 9-256.

Return to Summary Table.

Figure 9-230. EMIF4D_PHY_STS_23 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WRLVL_DQS_RATIO3									
R-0-0h						R-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PHY_REG_WRLVL_DQS_RATIO2									
R-0-0h						R-0h									

Table 9-256. EMIF4D_PHY_STS_23 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WRLVL_DQS_RATIO3	R	0h	Write levelling DQS ratio3
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WRLVL_DQS_RATIO2	R	0h	Write levelling DQS ratio2

9.3.4.89 EMIF4D_PHY_STS_24 Register (Offset = 1A0h) [reset = 0h]

EMIF4D_PHY_STS_24 is shown in Figure 9-231 and described in Table 9-257.

Return to Summary Table.

Figure 9-231. EMIF4D_PHY_STS_24 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WRLVL_DQS_RATIO5									
R-0-0h						R-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PHY_REG_WRLVL_DQS_RATIO4									
R-0-0h						R-0h									

Table 9-257. EMIF4D_PHY_STS_24 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WRLVL_DQS_RATIO5	R	0h	Write levelling DQS ratio5
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WRLVL_DQS_RATIO4	R	0h	Write levelling DQS ratio4

9.3.4.90 EMIF4D_PHY_STS_25 Register (Offset = 1A4h) [reset = 0h]

EMIF4D_PHY_STS_25 is shown in Figure 9-232 and described in Table 9-258.

Return to Summary Table.

Figure 9-232. EMIF4D_PHY_STS_25 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WRLVL_DQS_RATIO7</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WRLVL_DQS_RATIO6</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R-0h</td></tr>
</table>

Table 9-258. EMIF4D_PHY_STS_25 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WRLVL_DQS_RATIO7	R	0h	Write levelling DQS ratio7
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WRLVL_DQS_RATIO6	R	0h	Write levelling DQS ratio6

9.3.4.91 EMIF4D_PHY_STS_26 Register (Offset = 1A8h) [reset = 0h]

EMIF4D_PHY_STS_26 is shown in Figure 9-233 and described in Table 9-259.

Return to Summary Table.

Figure 9-233. EMIF4D_PHY_STS_26 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WRLVL_DQS_RATIO9</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WRLVL_DQS_RATIO8</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R-0h</td></tr>
</table>

Table 9-259. EMIF4D_PHY_STS_26 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WRLVL_DQS_RATIO9	R	0h	Write levelling DQS ratio9
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WRLVL_DQS_RATIO8	R	0h	Write levelling DQS ratio8

9.3.4.92 EMIF4D_PHY_STS_27 Register (Offset = 1ACh) [reset = 0h]

EMIF4D_PHY_STS_27 is shown in Figure 9-234 and described in Table 9-260.

Return to Summary Table.

Figure 9-234. EMIF4D_PHY_STS_27 Register

31	30	29	28	27	26	25	24
RESERVED		PHY_REG_CTRL_MDLL_UNLOCK_STICKY		RESERVED			PHY_REG_STS_MDLL_UNLOCK_STICKY
R-0-0h		R-0h		R-0-0h			R-0h

23	22	21	20	19	18	17	16
PHY_REG_STS_MDLL_UNLOCK_STICKY				PHY_REG_RDC_FIFO_RST_ERR_CNT			
R-0h				R-0h			

15	14	13	12	11	10	9	8
PHY_REG_RDC_FIFO_RST_ERR_CNT							
R-0h							

7	6	5	4	3	2	1	0
PHY_REG_RDC_FIFO_RST_ERR_CNT							
R-0h							

Table 9-260. EMIF4D_PHY_STS_27 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R-0	0h	
29-28	PHY_REG_CTRL_MDLL_UNLOCK_STICKY	R	0h	Phy control MDLL unlock sticky
27-25	RESERVED	R-0	0h	
24-20	PHY_REG_STS_MDLL_UNLOCK_STICKY	R	0h	Phy data MDLL unlock sticky
19-0	PHY_REG_RDC_FIFO_RST_ERR_CNT	R	0h	RDC FIFO reset error count

9.3.4.93 EMIF4D_PHY_STS_28 Register (Offset = 1B0h) [reset = 0h]

EMIF4D_PHY_STS_28 is shown in Figure 9-235 and described in Table 9-261.

Return to Summary Table.

Figure 9-235. EMIF4D_PHY_STS_28 Register

31	30	29	28	27	26	25	24
RESERVED			PHY_REG_GATELVL_INC_FAIL				
R-0-0h			R-0h				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED			PHY_REG_WRLVL_INC_FAIL				
R-0-0h			R-0h				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED			PHY_REG_RDLVL_INC_FAIL				
R-0-0h			R-0h				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED			PHY_REG_FIFO_WE_IN_MIASALIGNED_STICKY				
R-0-0h			R-0h				

Table 9-261. EMIF4D_PHY_STS_28 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R-0	0h	
28-24	PHY_REG_GATELVL_INC_FAIL	R	0h	Gate levelling failure
23-21	RESERVED	R-0	0h	
20-16	PHY_REG_WRLVL_INC_FAIL	R	0h	Write levelling failure
15-13	RESERVED	R-0	0h	
12-8	PHY_REG_RDLVL_INC_FAIL	R	0h	Read levelling failure
7-5	RESERVED	R-0	0h	
4-0	PHY_REG_FIFO_WE_IN_MIASALIGNED_STICKY	R	0h	FIFO write enable in misaligned stickly

9.3.4.94 EMIF4D_EXT_PHY_CTRL_1 Register (Offset = 200h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_1 is shown in Figure 9-236 and described in Table 9-262.

Return to Summary Table.

Figure 9-236. EMIF4D_EXT_PHY_CTRL_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED											PHY_REG_CTRL_SLAVE_RATIO_1				
R/W-0h											R/W-0h				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PHY_REG_CTRL_SLAVE_RATIO_1						PHY_REG_CTRL_SLAVE_RATIO_0									
R/W-0h						R/W-0h									

Table 9-262. EMIF4D_EXT_PHY_CTRL_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R/W	0h	
20-10	PHY_REG_CTRL_SLAVE_RATIO_1	R/W	0h	PHY Control Slave Ratio for Command Slice 1
9-0	PHY_REG_CTRL_SLAVE_RATIO_0	R/W	0h	PHY Control Slave Ratio for Command Slice 0

9.3.4.95 EMIF4D_EXT_PHY_CTRL_1_SHADOW Register (Offset = 204h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_1_SHADOW is shown in Figure 9-237 and described in Table 9-263.

Return to Summary Table.

Figure 9-237. EMIF4D_EXT_PHY_CTRL_1_SHADOW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="11">RESERVED</td><td colspan="5">PHY_REG_CTRL_SLAVE_RATIO_1</td></tr>
<tr><td colspan="11">R/W-0h</td><td colspan="5">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">PHY_REG_CTRL_SLAVE_RATIO_1</td><td colspan="10">PHY_REG_CTRL_SLAVE_RATIO_0</td></tr>
<tr><td colspan="6">R/W-0h</td><td colspan="10">R/W-0h</td></tr>
</table>

Table 9-263. EMIF4D_EXT_PHY_CTRL_1_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R/W	0h	
20-10	PHY_REG_CTRL_SLAVE_RATIO_1	R/W	0h	PHY Control Slave Ratio for Command Slice 1
9-0	PHY_REG_CTRL_SLAVE_RATIO_0	R/W	0h	PHY Control Slave Ratio for Command Slice 0

9.3.4.96 EMIF4D_EXT_PHY_CTRL_2 Register (Offset = 208h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_2 is shown in Figure 9-238 and described in Table 9-264.

Return to Summary Table.

Figure 9-238. EMIF4D_EXT_PHY_CTRL_2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO1										
R-0-0h					R/W-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO0										
R-0-0h					R/W-0h										

Table 9-264. EMIF4D_EXT_PHY_CTRL_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	PHY_REG_FIFO_WE_SLAVE_RATIO1	R/W	0h	FIFO write enable slave ratio1
15-11	RESERVED	R-0	0h	
10-0	PHY_REG_FIFO_WE_SLAVE_RATIO0	R/W	0h	FIFO write enable slave ratio0

9.3.4.97 EMIF4D_EXT_PHY_CTRL_2_SHADOW Register (Offset = 20Ch) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_2_SHADOW is shown in Figure 9-239 and described in Table 9-265.

Return to Summary Table.

Figure 9-239. EMIF4D_EXT_PHY_CTRL_2_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO1										
R-0-0h					R/W-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO0										
R-0-0h					R/W-0h										

Table 9-265. EMIF4D_EXT_PHY_CTRL_2_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	PHY_REG_FIFO_WE_SLAVE_RATIO1	R/W	0h	FIFO write enable slave ratio1
15-11	RESERVED	R-0	0h	
10-0	PHY_REG_FIFO_WE_SLAVE_RATIO0	R/W	0h	FIFO write enable slave ratio0

9.3.4.98 EMIF4D_EXT_PHY_CTRL_3 Register (Offset = 210h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_3 is shown in Figure 9-240 and described in Table 9-266.

Return to Summary Table.

Figure 9-240. EMIF4D_EXT_PHY_CTRL_3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO3										
R-0-0h					R/W-0h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO2										
R-0-0h					R/W-0h										

Table 9-266. EMIF4D_EXT_PHY_CTRL_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	PHY_REG_FIFO_WE_SLAVE_RATIO3	R/W	0h	FIFO write enable slave ratio3
15-11	RESERVED	R-0	0h	
10-0	PHY_REG_FIFO_WE_SLAVE_RATIO2	R/W	0h	FIFO write enable slave ratio2

9.3.4.99 EMIF4D_EXT_PHY_CTRL_3_SHADOW Register (Offset = 214h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_3_SHADOW is shown in Figure 9-241 and described in Table 9-267.

Return to Summary Table.

Figure 9-241. EMIF4D_EXT_PHY_CTRL_3_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO3										
R-0-0h					R/W-0h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO2										
R-0-0h					R/W-0h										

Table 9-267. EMIF4D_EXT_PHY_CTRL_3_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	PHY_REG_FIFO_WE_SLAVE_RATIO3	R/W	0h	FIFO write enable slave ratio3
15-11	RESERVED	R-0	0h	
10-0	PHY_REG_FIFO_WE_SLAVE_RATIO2	R/W	0h	FIFO write enable slave ratio2

9.3.4.100 EMIF4D_EXT_PHY_CTRL_4 Register (Offset = 218h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_4 is shown in Figure 9-242 and described in Table 9-268.

Return to Summary Table.

Figure 9-242. EMIF4D_EXT_PHY_CTRL_4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO5										
R-0-0h					R/W-0h										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO4										
R-0-0h					R/W-0h										

Table 9-268. EMIF4D_EXT_PHY_CTRL_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	PHY_REG_FIFO_WE_SLAVE_RATIO5	R/W	0h	FIFO write enable slave ratio5
15-11	RESERVED	R-0	0h	
10-0	PHY_REG_FIFO_WE_SLAVE_RATIO4	R/W	0h	FIFO write enable slave ratio4

9.3.4.101 EMIF4D_EXT_PHY_CTRL_4_SHADOW Register (Offset = 21Ch) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_4_SHADOW is shown in Figure 9-243 and described in Table 9-269.

Return to Summary Table.

Figure 9-243. EMIF4D_EXT_PHY_CTRL_4_SHADOW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="5">RESERVED</td><td colspan="11">PHY_REG_FIFO_WE_SLAVE_RATIO5</td></tr>
<tr><td colspan="5">R-0-0h</td><td colspan="11">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">RESERVED</td><td colspan="11">PHY_REG_FIFO_WE_SLAVE_RATIO4</td></tr>
<tr><td colspan="5">R-0-0h</td><td colspan="11">R/W-0h</td></tr>
</table>

Table 9-269. EMIF4D_EXT_PHY_CTRL_4_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	PHY_REG_FIFO_WE_SLAVE_RATIO5	R/W	0h	FIFO write enable slave ratio5
15-11	RESERVED	R-0	0h	
10-0	PHY_REG_FIFO_WE_SLAVE_RATIO4	R/W	0h	FIFO write enable slave ratio4

9.3.4.102 EMIF4D_EXT_PHY_CTRL_5 Register (Offset = 220h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_5 is shown in Figure 9-244 and described in Table 9-270.

Return to Summary Table.

Figure 9-244. EMIF4D_EXT_PHY_CTRL_5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO7										
R-0-0h					R/W-0h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO6										
R-0-0h					R/W-0h										

Table 9-270. EMIF4D_EXT_PHY_CTRL_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	PHY_REG_FIFO_WE_SLAVE_RATIO7	R/W	0h	FIFO wrie enable slave ratio7
15-11	RESERVED	R-0	0h	
10-0	PHY_REG_FIFO_WE_SLAVE_RATIO6	R/W	0h	FIFO wrie enable slave ratio6

9.3.4.103 EMIF4D_EXT_PHY_CTRL_5_SHADOW Register (Offset = 224h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_5_SHADOW is shown in Figure 9-245 and described in Table 9-271.

Return to Summary Table.

Figure 9-245. EMIF4D_EXT_PHY_CTRL_5_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO7										
R-0-0h					R/W-0h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO6										
R-0-0h					R/W-0h										

Table 9-271. EMIF4D_EXT_PHY_CTRL_5_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	PHY_REG_FIFO_WE_SLAVE_RATIO7	R/W	0h	FIFO wrie enable slave ratio7
15-11	RESERVED	R-0	0h	
10-0	PHY_REG_FIFO_WE_SLAVE_RATIO6	R/W	0h	FIFO wrie enable slave ratio6

9.3.4.104 EMIF4D_EXT_PHY_CTRL_6 Register (Offset = 228h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_6 is shown in Figure 9-246 and described in Table 9-272.

Return to Summary Table.

Figure 9-246. EMIF4D_EXT_PHY_CTRL_6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO9										
R-0-0h					R/W-0h										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PHY_REG_FIFO_WE_SLAVE_RATIO8										
R-0-0h					R/W-0h										

Table 9-272. EMIF4D_EXT_PHY_CTRL_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	PHY_REG_FIFO_WE_SLAVE_RATIO9	R/W	0h	FIFO write enable slave ratio9
15-11	RESERVED	R-0	0h	
10-0	PHY_REG_FIFO_WE_SLAVE_RATIO8	R/W	0h	FIFO write enable slave ratio8

9.3.4.105 EMIF4D_EXT_PHY_CTRL_6_SHADOW Register (Offset = 22Ch) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_6_SHADOW is shown in Figure 9-247 and described in Table 9-273.

Return to Summary Table.

Figure 9-247. EMIF4D_EXT_PHY_CTRL_6_SHADOW Register

31-27	26-16
RESERVED	PHY_REG_FIFO_WE_SLAVE_RATIO9
R-0-0h	R/W-0h

15-11	10-0
RESERVED	PHY_REG_FIFO_WE_SLAVE_RATIO8
R-0-0h	R/W-0h

Table 9-273. EMIF4D_EXT_PHY_CTRL_6_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	PHY_REG_FIFO_WE_SLAVE_RATIO9	R/W	0h	FIFO write enable slave ratio9
15-11	RESERVED	R-0	0h	
10-0	PHY_REG_FIFO_WE_SLAVE_RATIO8	R/W	0h	FIFO write enable slave ratio8

9.3.4.106 EMIF4D_EXT_PHY_CTRL_7 Register (Offset = 230h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_7 is shown in Figure 9-248 and described in Table 9-274.

Return to Summary Table.

Figure 9-248. EMIF4D_EXT_PHY_CTRL_7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_RD_DQS_SLAVE_RATIO1									
R-0-0h						R/W-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PHY_REG_RD_DQS_SLAVE_RATIO0									
R-0-0h						R/W-0h									

Table 9-274. EMIF4D_EXT_PHY_CTRL_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_RD_DQS_SLAVE_RATIO1	R/W	0h	Read DQS Slave Ratio1
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_RD_DQS_SLAVE_RATIO0	R/W	0h	Read DQS Slave Ratio0

9.3.4.107 EMIF4D_EXT_PHY_CTRL_7_SHADOW Register (Offset = 234h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_7_SHADOW is shown in Figure 9-249 and described in Table 9-275.

Return to Summary Table.

Figure 9-249. EMIF4D_EXT_PHY_CTRL_7_SHADOW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_RD_DQS_SLAVE_RATIO1</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_RD_DQS_SLAVE_RATIO0</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
</table>

Table 9-275. EMIF4D_EXT_PHY_CTRL_7_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_RD_DQS_SLAVE_RATIO1	R/W	0h	Read DQS Slave Ratio1
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_RD_DQS_SLAVE_RATIO0	R/W	0h	Read DQS Slave Ratio0

9.3.4.108 EMIF4D_EXT_PHY_CTRL_8 Register (Offset = 238h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_8 is shown in Figure 9-250 and described in Table 9-276.

Return to Summary Table.

Figure 9-250. EMIF4D_EXT_PHY_CTRL_8 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_RD_DQS_SLAVE_RATIO3</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_RD_DQS_SLAVE_RATIO2</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
</table>

Table 9-276. EMIF4D_EXT_PHY_CTRL_8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_RD_DQS_SLAVE_RATIO3	R/W	0h	Read DQS Slave Ratio3
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_RD_DQS_SLAVE_RATIO2	R/W	0h	Read DQS Slave Ratio2

9.3.4.109 EMIF4D_EXT_PHY_CTRL_8_SHADOW Register (Offset = 23Ch) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_8_SHADOW is shown in Figure 9-251 and described in Table 9-277.

Return to Summary Table.

Figure 9-251. EMIF4D_EXT_PHY_CTRL_8_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_RD_DQS_SLAVE_RATIO3									
R-0-0h						R/W-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PHY_REG_RD_DQS_SLAVE_RATIO2									
R-0-0h						R/W-0h									

Table 9-277. EMIF4D_EXT_PHY_CTRL_8_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_RD_DQS_SLAVE_RATIO3	R/W	0h	Read DQS Slave Ratio3
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_RD_DQS_SLAVE_RATIO2	R/W	0h	Read DQS Slave Ratio2

9.3.4.110 EMIF4D_EXT_PHY_CTRL_9 Register (Offset = 240h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_9 is shown in Figure 9-252 and described in Table 9-278.

Return to Summary Table.

Figure 9-252. EMIF4D_EXT_PHY_CTRL_9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_RD_DQS_SLAVE_RATIO5									
R-0-0h						R/W-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PHY_REG_RD_DQS_SLAVE_RATIO4									
R-0-0h						R/W-0h									

Table 9-278. EMIF4D_EXT_PHY_CTRL_9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_RD_DQS_SLAVE_RATIO5	R/W	0h	Read DQS Slave Ratio5
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_RD_DQS_SLAVE_RATIO4	R/W	0h	Read DQS Slave Ratio4

9.3.4.111 EMIF4D_EXT_PHY_CTRL_9_SHADOW Register (Offset = 244h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_9_SHADOW is shown in Figure 9-253 and described in Table 9-279.

Return to Summary Table.

Figure 9-253. EMIF4D_EXT_PHY_CTRL_9_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_RD_DQS_SLAVE_RATIO5									
R-0-0h						R/W-0h									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PHY_REG_RD_DQS_SLAVE_RATIO4									
R-0-0h						R/W-0h									

Table 9-279. EMIF4D_EXT_PHY_CTRL_9_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_RD_DQS_SLAVE_RATIO5	R/W	0h	Read DQS Slave Ratio5
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_RD_DQS_SLAVE_RATIO4	R/W	0h	Read DQS Slave Ratio4

9.3.4.112 EMIF4D_EXT_PHY_CTRL_10 Register (Offset = 248h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_10 is shown in Figure 9-254 and described in Table 9-280.

Return to Summary Table.

Figure 9-254. EMIF4D_EXT_PHY_CTRL_10 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_RD_DQS_SLAVE_RATIO7</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_RD_DQS_SLAVE_RATIO6</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
</table>

Table 9-280. EMIF4D_EXT_PHY_CTRL_10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_RD_DQS_SLAVE_RATIO7	R/W	0h	Read DQS Slave Ratio7
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_RD_DQS_SLAVE_RATIO6	R/W	0h	Read DQS Slave Ratio6

9.3.4.113 EMIF4D_EXT_PHY_CTRL_10_SHADOW Register (Offset = 24Ch) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_10_SHADOW is shown in Figure 9-255 and described in Table 9-281.

Return to Summary Table.

Figure 9-255. EMIF4D_EXT_PHY_CTRL_10_SHADOW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_RD_DQS_SLAVE_RATIO7</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_RD_DQS_SLAVE_RATIO6</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
</table>

Table 9-281. EMIF4D_EXT_PHY_CTRL_10_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_RD_DQS_SLAVE_RATIO7	R/W	0h	Read DQS Slave Ratio7
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_RD_DQS_SLAVE_RATIO6	R/W	0h	Read DQS Slave Ratio6

9.3.4.114 EMIF4D_EXT_PHY_CTRL_11 Register (Offset = 250h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_11 is shown in Figure 9-256 and described in Table 9-282.

Return to Summary Table.

Figure 9-256. EMIF4D_EXT_PHY_CTRL_11 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_RD_DQS_SLAVE_RATIO9									
R-0-0h						R/W-0h									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PHY_REG_RD_DQS_SLAVE_RATIO8									
R-0-0h						R/W-0h									

Table 9-282. EMIF4D_EXT_PHY_CTRL_11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_RD_DQS_SLAVE_RATIO9	R/W	0h	Read DQS Slave Ratio9
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_RD_DQS_SLAVE_RATIO8	R/W	0h	Read DQS Slave Ratio8

9.3.4.115 EMIF4D_EXT_PHY_CTRL_11_SHADOW Register (Offset = 254h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_11_SHADOW is shown in Figure 9-257 and described in Table 9-283.

Return to Summary Table.

Figure 9-257. EMIF4D_EXT_PHY_CTRL_11_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_RD_DQS_SLAVE_RATIO9									
R-0-0h						R/W-0h									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PHY_REG_RD_DQS_SLAVE_RATIO8									
R-0-0h						R/W-0h									

Table 9-283. EMIF4D_EXT_PHY_CTRL_11_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_RD_DQS_SLAVE_RATIO9	R/W	0h	Read DQS Slave Ratio9
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_RD_DQS_SLAVE_RATIO8	R/W	0h	Read DQS Slave Ratio8

9.3.4.116 EMIF4D_EXT_PHY_CTRL_12 Register (Offset = 258h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_12 is shown in Figure 9-258 and described in Table 9-284.

Return to Summary Table.

Figure 9-258. EMIF4D_EXT_PHY_CTRL_12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WR_DATA_SLAVE_RATIO1									
R-0-0h						R/W-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PHY_REG_WR_DATA_SLAVE_RATIO0									
R-0-0h						R/W-0h									

Table 9-284. EMIF4D_EXT_PHY_CTRL_12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DATA_SLAVE_RATIO1	R/W	0h	Write Data Slave Ratio1
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DATA_SLAVE_RATIO0	R/W	0h	Write Data Slave Ratio0

9.3.4.117 EMIF4D_EXT_PHY_CTRL_12_SHADOW Register (Offset = 25Ch) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_12_SHADOW is shown in Figure 9-259 and described in Table 9-285.

Return to Summary Table.

Figure 9-259. EMIF4D_EXT_PHY_CTRL_12_SHADOW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WR_DATA_SLAVE_RATIO1</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WR_DATA_SLAVE_RATIO0</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
</table>

Table 9-285. EMIF4D_EXT_PHY_CTRL_12_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DATA_SLAVE_RATIO1	R/W	0h	Write Data Slave Ratio1
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DATA_SLAVE_RATIO0	R/W	0h	Write Data Slave Ratio0

9.3.4.118 EMIF4D_EXT_PHY_CTRL_13 Register (Offset = 260h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_13 is shown in Figure 9-260 and described in Table 9-286.

Return to Summary Table.

Figure 9-260. EMIF4D_EXT_PHY_CTRL_13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WR_DATA_SLAVE_RATIO3									
R-0-0h						R/W-0h									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PHY_REG_WR_DATA_SLAVE_RATIO2									
R-0-0h						R/W-0h									

Table 9-286. EMIF4D_EXT_PHY_CTRL_13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DATA_SLAVE_RATIO3	R/W	0h	Write Data Slave Ratio3
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DATA_SLAVE_RATIO2	R/W	0h	Write Data Slave Ratio2

9.3.4.119 EMIF4D_EXT_PHY_CTRL_13_SHADOW Register (Offset = 264h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_13_SHADOW is shown in Figure 9-261 and described in Table 9-287.

Return to Summary Table.

Figure 9-261. EMIF4D_EXT_PHY_CTRL_13_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WR_DATA_SLAVE_RATIO3									
R-0-0h						R/W-0h									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PHY_REG_WR_DATA_SLAVE_RATIO2									
R-0-0h						R/W-0h									

Table 9-287. EMIF4D_EXT_PHY_CTRL_13_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DATA_SLAVE_RATIO3	R/W	0h	Write Data Slave Ratio3
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DATA_SLAVE_RATIO2	R/W	0h	Write Data Slave Ratio2

9.3.4.120 EMIF4D_EXT_PHY_CTRL_14 Register (Offset = 268h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_14 is shown in Figure 9-262 and described in Table 9-288.

Return to Summary Table.

Figure 9-262. EMIF4D_EXT_PHY_CTRL_14 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WR_DATA_SLAVE_RATIO5									
R-0-0h						R/W-0h									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PHY_REG_WR_DATA_SLAVE_RATIO4									
R-0-0h						R/W-0h									

Table 9-288. EMIF4D_EXT_PHY_CTRL_14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DATA_SLAVE_RATIO5	R/W	0h	Write Data Slave Ratio5
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DATA_SLAVE_RATIO4	R/W	0h	Write Data Slave Ratio4

9.3.4.121 EMIF4D_EXT_PHY_CTRL_14_SHADOW Register (Offset = 26Ch) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_14_SHADOW is shown in Figure 9-263 and described in Table 9-289.

Return to Summary Table.

Figure 9-263. EMIF4D_EXT_PHY_CTRL_14_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WR_DATA_SLAVE_RATIO5									
R-0-0h						R/W-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PHY_REG_WR_DATA_SLAVE_RATIO4									
R-0-0h						R/W-0h									

Table 9-289. EMIF4D_EXT_PHY_CTRL_14_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DATA_SLAVE_RATIO5	R/W	0h	Write Data Slave Ratio5
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DATA_SLAVE_RATIO4	R/W	0h	Write Data Slave Ratio4

9.3.4.122 EMIF4D_EXT_PHY_CTRL_15 Register (Offset = 270h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_15 is shown in Figure 9-264 and described in Table 9-290.

Return to Summary Table.

Figure 9-264. EMIF4D_EXT_PHY_CTRL_15 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WR_DATA_SLAVE_RATIO7</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WR_DATA_SLAVE_RATIO6</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
</table>

Table 9-290. EMIF4D_EXT_PHY_CTRL_15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DATA_SLAVE_RATIO7	R/W	0h	Write Data Slave Ratio7
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DATA_SLAVE_RATIO6	R/W	0h	Write Data Slave Ratio6

9.3.4.123 EMIF4D_EXT_PHY_CTRL_15_SHADOW Register (Offset = 274h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_15_SHADOW is shown in Figure 9-265 and described in Table 9-291.

Return to Summary Table.

Figure 9-265. EMIF4D_EXT_PHY_CTRL_15_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WR_DATA_SLAVE_RATIO7									
R-0-0h						R/W-0h									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PHY_REG_WR_DATA_SLAVE_RATIO6									
R-0-0h						R/W-0h									

Table 9-291. EMIF4D_EXT_PHY_CTRL_15_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DATA_SLAVE_RATIO7	R/W	0h	Write Data Slave Ratio7
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DATA_SLAVE_RATIO6	R/W	0h	Write Data Slave Ratio6

9.3.4.124 EMIF4D_EXT_PHY_CTRL_16 Register (Offset = 278h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_16 is shown in Figure 9-266 and described in Table 9-292.

Return to Summary Table.

Figure 9-266. EMIF4D_EXT_PHY_CTRL_16 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WR_DATA_SLAVE_RATIO9									
R-0-0h						R/W-0h									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PHY_REG_WR_DATA_SLAVE_RATIO8									
R-0-0h						R/W-0h									

Table 9-292. EMIF4D_EXT_PHY_CTRL_16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DATA_SLAVE_RATIO9	R/W	0h	Write Data Slave Ratio9
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DATA_SLAVE_RATIO8	R/W	0h	Write Data Slave Ratio8

9.3.4.125 EMIF4D_EXT_PHY_CTRL_16_SHADOW Register (Offset = 27Ch) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_16_SHADOW is shown in Figure 9-267 and described in Table 9-293.

Return to Summary Table.

Figure 9-267. EMIF4D_EXT_PHY_CTRL_16_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WR_DATA_SLAVE_RATIO9									
R-0-0h						R/W-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PHY_REG_WR_DATA_SLAVE_RATIO8									
R-0-0h						R/W-0h									

Table 9-293. EMIF4D_EXT_PHY_CTRL_16_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DATA_SLAVE_RATIO9	R/W	0h	Write Data Slave Ratio9
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DATA_SLAVE_RATIO8	R/W	0h	Write Data Slave Ratio8

9.3.4.126 EMIF4D_EXT_PHY_CTRL_17 Register (Offset = 280h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_17 is shown in Figure 9-268 and described in Table 9-294.

Return to Summary Table.

Figure 9-268. EMIF4D_EXT_PHY_CTRL_17 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO1									
R-0-0h						R/W-0h									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO0									
R-0-0h						R/W-0h									

Table 9-294. EMIF4D_EXT_PHY_CTRL_17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DQS_SLAVE_RATIO1	R/W	0h	Write Data Slave Ratio1
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DQS_SLAVE_RATIO0	R/W	0h	Write Data Slave Ratio0

9.3.4.127 EMIF4D_EXT_PHY_CTRL_17_SHADOW Register (Offset = 284h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_17_SHADOW is shown in Figure 9-269 and described in Table 9-295.

Return to Summary Table.

Figure 9-269. EMIF4D_EXT_PHY_CTRL_17_SHADOW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WR_DQS_SLAVE_RATIO1</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WR_DQS_SLAVE_RATIO0</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
</table>

Table 9-295. EMIF4D_EXT_PHY_CTRL_17_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DQS_SLAVE_RATIO1	R/W	0h	Write Data Slave Ratio1
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DQS_SLAVE_RATIO0	R/W	0h	Write Data Slave Ratio0

9.3.4.128 EMIF4D_EXT_PHY_CTRL_18 Register (Offset = 288h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_18 is shown in Figure 9-270 and described in Table 9-296.

Return to Summary Table.

Figure 9-270. EMIF4D_EXT_PHY_CTRL_18 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO3									
R-0-0h						R/W-0h									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO2									
R-0-0h						R/W-0h									

Table 9-296. EMIF4D_EXT_PHY_CTRL_18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DQS_SLAVE_RATIO3	R/W	0h	Write Data Slave Ratio3
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DQS_SLAVE_RATIO2	R/W	0h	Write Data Slave Ratio2

9.3.4.129 EMIF4D_EXT_PHY_CTRL_18_SHADOW Register (Offset = 28Ch) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_18_SHADOW is shown in Figure 9-271 and described in Table 9-297.

Return to Summary Table.

Figure 9-271. EMIF4D_EXT_PHY_CTRL_18_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO3									
R-0-0h						R/W-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO2									
R-0-0h						R/W-0h									

Table 9-297. EMIF4D_EXT_PHY_CTRL_18_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DQS_SLAVE_RATIO3	R/W	0h	Write Data Slave Ratio3
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DQS_SLAVE_RATIO2	R/W	0h	Write Data Slave Ratio2

9.3.4.130 EMIF4D_EXT_PHY_CTRL_19 Register (Offset = 290h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_19 is shown in Figure 9-272 and described in Table 9-298.

Return to Summary Table.

Figure 9-272. EMIF4D_EXT_PHY_CTRL_19 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WR_DQS_SLAVE_RATIO5</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">PHY_REG_WR_DQS_SLAVE_RATIO4</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
</table>

Table 9-298. EMIF4D_EXT_PHY_CTRL_19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DQS_SLAVE_RATIO5	R/W	0h	Write Data Slave Ratio5
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DQS_SLAVE_RATIO4	R/W	0h	Write Data Slave Ratio4

9.3.4.131 EMIF4D_EXT_PHY_CTRL_19_SHADOW Register (Offset = 294h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_19_SHADOW is shown in Figure 9-273 and described in Table 9-299.

Return to Summary Table.

Figure 9-273. EMIF4D_EXT_PHY_CTRL_19_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO5									
R-0-0h						R/W-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO4									
R-0-0h						R/W-0h									

Table 9-299. EMIF4D_EXT_PHY_CTRL_19_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DQS_SLAVE_RATIO5	R/W	0h	Write Data Slave Ratio5
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DQS_SLAVE_RATIO4	R/W	0h	Write Data Slave Ratio4

9.3.4.132 EMIF4D_EXT_PHY_CTRL_20 Register (Offset = 298h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_20 is shown in Figure 9-274 and described in Table 9-300.

Return to Summary Table.

Figure 9-274. EMIF4D_EXT_PHY_CTRL_20 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO7									
R-0-0h						R/W-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO6									
R-0-0h						R/W-0h									

Table 9-300. EMIF4D_EXT_PHY_CTRL_20 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DQS_SLAVE_RATIO7	R/W	0h	Write Data Slave Ratio7
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DQS_SLAVE_RATIO6	R/W	0h	Write Data Slave Ratio6

9.3.4.133 EMIF4D_EXT_PHY_CTRL_20_SHADOW Register (Offset = 29Ch) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_20_SHADOW is shown in Figure 9-275 and described in Table 9-301.

Return to Summary Table.

Figure 9-275. EMIF4D_EXT_PHY_CTRL_20_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO7									
R-0-0h						R/W-0h									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO6									
R-0-0h						R/W-0h									

Table 9-301. EMIF4D_EXT_PHY_CTRL_20_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DQS_SLAVE_RATIO7	R/W	0h	Write Data Slave Ratio7
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DQS_SLAVE_RATIO6	R/W	0h	Write Data Slave Ratio6

9.3.4.134 EMIF4D_EXT_PHY_CTRL_21 Register (Offset = 2A0h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_21 is shown in Figure 9-276 and described in Table 9-302.

Return to Summary Table.

Figure 9-276. EMIF4D_EXT_PHY_CTRL_21 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO9									
R-0-0h						R/W-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO8									
R-0-0h						R/W-0h									

Table 9-302. EMIF4D_EXT_PHY_CTRL_21 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DQS_SLAVE_RATIO9	R/W	0h	Write Data Slave Ratio9
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DQS_SLAVE_RATIO8	R/W	0h	Write Data Slave Ratio8

9.3.4.135 EMIF4D_EXT_PHY_CTRL_21_SHADOW Register (Offset = 2A4h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_21_SHADOW is shown in Figure 9-277 and described in Table 9-303.

Return to Summary Table.

Figure 9-277. EMIF4D_EXT_PHY_CTRL_21_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO9									
R-0-0h						R/W-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						PHY_REG_WR_DQS_SLAVE_RATIO8									
R-0-0h						R/W-0h									

Table 9-303. EMIF4D_EXT_PHY_CTRL_21_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	PHY_REG_WR_DQS_SLAVE_RATIO9	R/W	0h	Write Data Slave Ratio9
15-10	RESERVED	R-0	0h	
9-0	PHY_REG_WR_DQS_SLAVE_RATIO8	R/W	0h	Write Data Slave Ratio8

9.3.4.136 EMIF4D_EXT_PHY_CTRL_22 Register (Offset = 2A8h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_22 is shown in Figure 9-278 and described in Table 9-304.

Return to Summary Table.

Figure 9-278. EMIF4D_EXT_PHY_CTRL_22 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="7">RESERVED</td><td colspan="9">PHY_REG_FIFO_WE_IN_DELAY</td></tr>
<tr><td colspan="7">R-0-0h</td><td colspan="9">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="7">RESERVED</td><td colspan="9">PHY_REG_CTRL_SLAVE_DELAY</td></tr>
<tr><td colspan="7">R-0-0h</td><td colspan="9">R/W-0h</td></tr>
</table>

Table 9-304. EMIF4D_EXT_PHY_CTRL_22 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R-0	0h	
24-16	PHY_REG_FIFO_WE_IN_DELAY	R/W	0h	FIFO write enable in delay
15-9	RESERVED	R-0	0h	
8-0	PHY_REG_CTRL_SLAVE_DELAY	R/W	0h	Ctrl slave delay

9.3.4.137 EMIF4D_EXT_PHY_CTRL_22_SHADOW Register (Offset = 2ACh) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_22_SHADOW is shown in Figure 9-279 and described in Table 9-305.

Return to Summary Table.

Figure 9-279. EMIF4D_EXT_PHY_CTRL_22_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED							PHY_REG_FIFO_WE_IN_DELAY								
R-0-0h							R/W-0h								
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							PHY_REG_CTRL_SLAVE_DELAY								
R-0-0h							R/W-0h								

Table 9-305. EMIF4D_EXT_PHY_CTRL_22_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R-0	0h	
24-16	PHY_REG_FIFO_WE_IN_DELAY	R/W	0h	FIFO write enable in delay
15-9	RESERVED	R-0	0h	
8-0	PHY_REG_CTRL_SLAVE_DELAY	R/W	0h	Ctrl slave delay

9.3.4.138 EMIF4D_EXT_PHY_CTRL_23 Register (Offset = 2B0h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_23 is shown in Figure 9-280 and described in Table 9-306.

Return to Summary Table.

Figure 9-280. EMIF4D_EXT_PHY_CTRL_23 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED							PHY_REG_WR_DQS_SLAVE_DELAY								
R-0-0h							R/W-0h								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							PHY_REG_RD_DQS_SLAVE_DELAY								
R-0-0h							R/W-0h								

Table 9-306. EMIF4D_EXT_PHY_CTRL_23 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R-0	0h	
24-16	PHY_REG_WR_DQS_SLAVE_DELAY	R/W	0h	Write DQS Slave delay
15-9	RESERVED	R-0	0h	
8-0	PHY_REG_RD_DQS_SLAVE_DELAY	R/W	0h	Read DQS Slave delay

9.3.4.139 EMIF4D_EXT_PHY_CTRL_23_SHADOW Register (Offset = 2B4h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_23_SHADOW is shown in Figure 9-281 and described in Table 9-307.

Return to Summary Table.

Figure 9-281. EMIF4D_EXT_PHY_CTRL_23_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED							PHY_REG_WR_DQS_SLAVE_DELAY								
R-0-0h							R/W-0h								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							PHY_REG_RD_DQS_SLAVE_DELAY								
R-0-0h							R/W-0h								

Table 9-307. EMIF4D_EXT_PHY_CTRL_23_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R-0	0h	
24-16	PHY_REG_WR_DQS_SLAVE_DELAY	R/W	0h	Write DQS Slave delay
15-9	RESERVED	R-0	0h	
8-0	PHY_REG_RD_DQS_SLAVE_DELAY	R/W	0h	Read DQS Slave delay

9.3.4.140 EMIF4D_EXT_PHY_CTRL_24 Register (Offset = 2B8h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_24 is shown in Figure 9-282 and described in Table 9-308.

Return to Summary Table.

Figure 9-282. EMIF4D_EXT_PHY_CTRL_24 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>RESERVED</td><td colspan="7">REG_PHY_DQ_OFFSET_HI</td></tr>
<tr><td>R-0-0h</td><td colspan="7">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="7">RESERVED</td><td>REG_PHY_GATELVL_INIT_MODE</td></tr>
<tr><td colspan="7">R-0-0h</td><td>R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="3">RESERVED</td><td>REG_PHY_USE_RANK0_DELAYS</td><td colspan="3">RESERVED</td><td>REG_PHY_WR_DATA_SLAVE_DELAY</td></tr>
<tr><td colspan="3">R-0-0h</td><td>R/W-0h</td><td colspan="3">R-0-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">REG_PHY_WR_DATA_SLAVE_DELAY</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 9-308. EMIF4D_EXT_PHY_CTRL_24 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0	0h	
30-24	REG_PHY_DQ_OFFSET_HI	R/W	0h	Phy DQ Offset bits 34:28
23-17	RESERVED	R-0	0h	
16	REG_PHY_GATELVL_INIT_MODE	R/W	0h	Gate levelling init mode
15-13	RESERVED	R-0	0h	
12	REG_PHY_USE_RANK0_DELAYS	R/W	0h	Use rank0 delays
11-9	RESERVED	R-0	0h	
8-0	REG_PHY_WR_DATA_SLAVE_DELAY	R/W	0h	Wirte data slave delay

9.3.4.141 EMIF4D_EXT_PHY_CTRL_24_SHADOW Register (Offset = 2BCh) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_24_SHADOW is shown in Figure 9-283 and described in Table 9-309.

Return to Summary Table.

Figure 9-283. EMIF4D_EXT_PHY_CTRL_24_SHADOW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>RESERVED</td><td colspan="7">REG_PHY_DQ_OFFSET_HI</td></tr>
<tr><td>R-0-0h</td><td colspan="7">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="7">RESERVED</td><td>REG_PHY_GATELVL_INIT_MODE</td></tr>
<tr><td colspan="7">R-0-0h</td><td>R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="3">RESERVED</td><td>REG_PHY_USE_RANK0_DELAYS</td><td colspan="3">RESERVED</td><td>REG_PHY_WR_DATA_SLAVE_DELAY</td></tr>
<tr><td colspan="3">R-0-0h</td><td>R/W-0h</td><td colspan="3">R-0-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">REG_PHY_WR_DATA_SLAVE_DELAY</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 9-309. EMIF4D_EXT_PHY_CTRL_24_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0	0h	
30-24	REG_PHY_DQ_OFFSET_HI	R/W	0h	Phy DQ Offset bits 34:28
23-17	RESERVED	R-0	0h	
16	REG_PHY_GATELVL_INIT_MODE	R/W	0h	Gate levelling init mode
15-13	RESERVED	R-0	0h	
12	REG_PHY_USE_RANK0_DELAYS	R/W	0h	Use rank0 delays
11-9	RESERVED	R-0	0h	
8-0	REG_PHY_WR_DATA_SLAVE_DELAY	R/W	0h	Wirte data slave delay

9.3.4.142 EMIF4D_EXT_PHY_CTRL_25 Register (Offset = 2C0h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_25 is shown in Figure 9-284 and described in Table 9-310.

Return to Summary Table.

Figure 9-284. EMIF4D_EXT_PHY_CTRL_25 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				REG_PHY_DQ_OFFSET											
R-0-0h				R/W-0h											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REG_PHY_DQ_OFFSET															
R/W-0h															

Table 9-310. EMIF4D_EXT_PHY_CTRL_25 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R-0	0h	
27-0	REG_PHY_DQ_OFFSET	R/W	0h	DQ offset

9.3.4.143 EMIF4D_EXT_PHY_CTRL_25_SHADOW Register (Offset = 2C4h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_25_SHADOW is shown in Figure 9-285 and described in Table 9-311.

Return to Summary Table.

Figure 9-285. EMIF4D_EXT_PHY_CTRL_25_SHADOW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="12">REG_PHY_DQ_OFFSET</td></tr>
<tr><td colspan="4">R-0-0h</td><td colspan="12">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">REG_PHY_DQ_OFFSET</td></tr>
<tr><td colspan="16">R/W-0h</td></tr>
</table>

Table 9-311. EMIF4D_EXT_PHY_CTRL_25_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R-0	0h	
27-0	REG_PHY_DQ_OFFSET	R/W	0h	DQ offset

9.3.4.144 EMIF4D_EXT_PHY_CTRL_26 Register (Offset = 2C8h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_26 is shown in Figure 9-286 and described in Table 9-312.

Return to Summary Table.

Figure 9-286. EMIF4D_EXT_PHY_CTRL_26 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					REG_PHY_GATELVL_INIT_RATIO1										
R-0-0h					R/W-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					REG_PHY_GATELVL_INIT_RATIO0										
R-0-0h					R/W-0h										

Table 9-312. EMIF4D_EXT_PHY_CTRL_26 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	REG_PHY_GATELVL_INIT_RATIO1	R/W	0h	Gate levelling init ratio1
15-11	RESERVED	R-0	0h	
10-0	REG_PHY_GATELVL_INIT_RATIO0	R/W	0h	Gate levelling init ratio0

9.3.4.145 EMIF4D_EXT_PHY_CTRL_26_SHADOW Register (Offset = 2CCh) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_26_SHADOW is shown in Figure 9-287 and described in Table 9-313.

Return to Summary Table.

Figure 9-287. EMIF4D_EXT_PHY_CTRL_26_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					REG_PHY_GATELVL_INIT_RATIO1										
R-0-0h					R/W-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					REG_PHY_GATELVL_INIT_RATIO0										
R-0-0h					R/W-0h										

Table 9-313. EMIF4D_EXT_PHY_CTRL_26_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	REG_PHY_GATELVL_INIT_RATIO1	R/W	0h	Gate levelling init ratio1
15-11	RESERVED	R-0	0h	
10-0	REG_PHY_GATELVL_INIT_RATIO0	R/W	0h	Gate levelling init ratio0

9.3.4.146 EMIF4D_EXT_PHY_CTRL_27 Register (Offset = 2D0h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_27 is shown in Figure 9-288 and described in Table 9-314.

Return to Summary Table.

Figure 9-288. EMIF4D_EXT_PHY_CTRL_27 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					REG_PHY_GATELVL_INIT_RATIO3										
R-0-0h					R/W-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					REG_PHY_GATELVL_INIT_RATIO2										
R-0-0h					R/W-0h										

Table 9-314. EMIF4D_EXT_PHY_CTRL_27 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	REG_PHY_GATELVL_INIT_RATIO3	R/W	0h	Gate levelling init ratio3
15-11	RESERVED	R-0	0h	
10-0	REG_PHY_GATELVL_INIT_RATIO2	R/W	0h	Gate levelling init ratio2

9.3.4.147 EMIF4D_EXT_PHY_CTRL_27_SHADOW Register (Offset = 2D4h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_27_SHADOW is shown in Figure 9-289 and described in Table 9-315.

Return to Summary Table.

Figure 9-289. EMIF4D_EXT_PHY_CTRL_27_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					REG_PHY_GATELVL_INIT_RATIO3										
R-0-0h					R/W-0h										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					REG_PHY_GATELVL_INIT_RATIO2										
R-0-0h					R/W-0h										

Table 9-315. EMIF4D_EXT_PHY_CTRL_27_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	REG_PHY_GATELVL_INIT_RATIO3	R/W	0h	Gate levelling init ratio3
15-11	RESERVED	R-0	0h	
10-0	REG_PHY_GATELVL_INIT_RATIO2	R/W	0h	Gate levelling init ratio2

9.3.4.148 EMIF4D_EXT_PHY_CTRL_28 Register (Offset = 2D8h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_28 is shown in Figure 9-290 and described in Table 9-316.

Return to Summary Table.

Figure 9-290. EMIF4D_EXT_PHY_CTRL_28 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					REG_PHY_GATELVL_INIT_RATIO5										
R-0-0h					R/W-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					REG_PHY_GATELVL_INIT_RATIO4										
R-0-0h					R/W-0h										

Table 9-316. EMIF4D_EXT_PHY_CTRL_28 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	REG_PHY_GATELVL_INIT_RATIO5	R/W	0h	Gate levelling init ratio5
15-11	RESERVED	R-0	0h	
10-0	REG_PHY_GATELVL_INIT_RATIO4	R/W	0h	Gate levelling init ratio4

9.3.4.149 EMIF4D_EXT_PHY_CTRL_28_SHADOW Register (Offset = 2DCh) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_28_SHADOW is shown in Figure 9-291 and described in Table 9-317.

Return to Summary Table.

Figure 9-291. EMIF4D_EXT_PHY_CTRL_28_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					REG_PHY_GATELVL_INIT_RATIO5										
R-0-0h					R/W-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					REG_PHY_GATELVL_INIT_RATIO4										
R-0-0h					R/W-0h										

Table 9-317. EMIF4D_EXT_PHY_CTRL_28_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	REG_PHY_GATELVL_INIT_RATIO5	R/W	0h	Gate levelling init ratio5
15-11	RESERVED	R-0	0h	
10-0	REG_PHY_GATELVL_INIT_RATIO4	R/W	0h	Gate levelling init ratio4

9.3.4.150 EMIF4D_EXT_PHY_CTRL_29 Register (Offset = 2E0h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_29 is shown in Figure 9-292 and described in Table 9-318.

Return to Summary Table.

Figure 9-292. EMIF4D_EXT_PHY_CTRL_29 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					REG_PHY_GATELVL_INIT_RATIO7										
R-0-0h					R/W-0h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					REG_PHY_GATELVL_INIT_RATIO6										
R-0-0h					R/W-0h										

Table 9-318. EMIF4D_EXT_PHY_CTRL_29 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	REG_PHY_GATELVL_INIT_RATIO7	R/W	0h	Gate levelling init ratio7
15-11	RESERVED	R-0	0h	
10-0	REG_PHY_GATELVL_INIT_RATIO6	R/W	0h	Gate levelling init ratio6

9.3.4.151 EMIF4D_EXT_PHY_CTRL_29_SHADOW Register (Offset = 2E4h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_29_SHADOW is shown in Figure 9-293 and described in Table 9-319.

Return to Summary Table.

Figure 9-293. EMIF4D_EXT_PHY_CTRL_29_SHADOW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="5">RESERVED</td><td colspan="11">REG_PHY_GATELVL_INIT_RATIO7</td></tr>
<tr><td colspan="5">R-0-0h</td><td colspan="11">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">RESERVED</td><td colspan="11">REG_PHY_GATELVL_INIT_RATIO6</td></tr>
<tr><td colspan="5">R-0-0h</td><td colspan="11">R/W-0h</td></tr>
</table>

Table 9-319. EMIF4D_EXT_PHY_CTRL_29_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	REG_PHY_GATELVL_INIT_RATIO7	R/W	0h	Gate levelling init ratio7
15-11	RESERVED	R-0	0h	
10-0	REG_PHY_GATELVL_INIT_RATIO6	R/W	0h	Gate levelling init ratio6

9.3.4.152 EMIF4D_EXT_PHY_CTRL_30 Register (Offset = 2E8h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_30 is shown in Figure 9-294 and described in Table 9-320.

Return to Summary Table.

Figure 9-294. EMIF4D_EXT_PHY_CTRL_30 Register

31-27	26-16
RESERVED	REG_PHY_GATELVL_INIT_RATIO9
R-0-0h	R/W-0h

15-11	10-0
RESERVED	REG_PHY_GATELVL_INIT_RATIO8
R-0-0h	R/W-0h

Table 9-320. EMIF4D_EXT_PHY_CTRL_30 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	REG_PHY_GATELVL_INIT_RATIO9	R/W	0h	Gate levelling init ratio9
15-11	RESERVED	R-0	0h	
10-0	REG_PHY_GATELVL_INIT_RATIO8	R/W	0h	Gate levelling init ratio8

9.3.4.153 EMIF4D_EXT_PHY_CTRL_30_SHADOW Register (Offset = 2ECh) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_30_SHADOW is shown in Figure 9-295 and described in Table 9-321.

Return to Summary Table.

Figure 9-295. EMIF4D_EXT_PHY_CTRL_30_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					REG_PHY_GATELVL_INIT_RATIO9										
R-0-0h					R/W-0h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					REG_PHY_GATELVL_INIT_RATIO8										
R-0-0h					R/W-0h										

Table 9-321. EMIF4D_EXT_PHY_CTRL_30_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	
26-16	REG_PHY_GATELVL_INIT_RATIO9	R/W	0h	Gate levelling init ratio9
15-11	RESERVED	R-0	0h	
10-0	REG_PHY_GATELVL_INIT_RATIO8	R/W	0h	Gate levelling init ratio8

9.3.4.154 EMIF4D_EXT_PHY_CTRL_31 Register (Offset = 2F0h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_31 is shown in Figure 9-296 and described in Table 9-322.

Return to Summary Table.

Figure 9-296. EMIF4D_EXT_PHY_CTRL_31 Register

31	30	29	28	27	26	25	24
RESERVED						RESERVED	
R-0-0h						R/W-0h	
23	22	21	20	19	18	17	16
RESERVED		REG_PHY_WRLVL_INIT_RATIO1					
R/W-0h		R/W-0h					
15	14	13	12	11	10	9	8
RESERVED						RESERVED	
R-0-0h						R/W-0h	
7	6	5	4	3	2	1	0
RESERVED		REG_PHY_WRLVL_INIT_RATIO0					
R/W-0h		R/W-0h					

Table 9-322. EMIF4D_EXT_PHY_CTRL_31 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-22	RESERVED	R/W	0h	
21-16	REG_PHY_WRLVL_INIT_RATIO1	R/W	0h	Write levelling init ratio1
15-10	RESERVED	R-0	0h	
9-6	RESERVED	R/W	0h	
5-0	REG_PHY_WRLVL_INIT_RATIO0	R/W	0h	Write levelling init ratio0

9.3.4.155 EMIF4D_EXT_PHY_CTRL_31_SHADOW Register (Offset = 2F4h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_31_SHADOW is shown in Figure 9-297 and described in Table 9-323.

Return to Summary Table.

Figure 9-297. EMIF4D_EXT_PHY_CTRL_31_SHADOW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">RESERVED</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">REG_PHY_WRLVL_INIT_RATIO1</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">RESERVED</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="2">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">REG_PHY_WRLVL_INIT_RATIO0</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="6">R/W-0h</td></tr>
</table>

Table 9-323. EMIF4D_EXT_PHY_CTRL_31_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-22	RESERVED	R/W	0h	
21-16	REG_PHY_WRLVL_INIT_RATIO1	R/W	0h	Write levelling init ratio1
15-10	RESERVED	R-0	0h	
9-6	RESERVED	R/W	0h	
5-0	REG_PHY_WRLVL_INIT_RATIO0	R/W	0h	Write levelling init ratio0

9.3.4.156 EMIF4D_EXT_PHY_CTRL_32 Register (Offset = 2F8h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_32 is shown in Figure 9-298 and described in Table 9-324.

Return to Summary Table.

Figure 9-298. EMIF4D_EXT_PHY_CTRL_32 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						REG_PHY_WRLVL_INIT_RATIO3									
R-0-0h						R/W-0h									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						REG_PHY_WRLVL_INIT_RATIO2									
R-0-0h						R/W-0h									

Table 9-324. EMIF4D_EXT_PHY_CTRL_32 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	REG_PHY_WRLVL_INIT_RATIO3	R/W	0h	Write levelling init ratio3
15-10	RESERVED	R-0	0h	
9-0	REG_PHY_WRLVL_INIT_RATIO2	R/W	0h	Write levelling init ratio2

9.3.4.157 EMIF4D_EXT_PHY_CTRL_32_SHADOW Register (Offset = 2FCh) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_32_SHADOW is shown in Figure 9-299 and described in Table 9-325.

Return to Summary Table.

Figure 9-299. EMIF4D_EXT_PHY_CTRL_32_SHADOW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">REG_PHY_WRLVL_INIT_RATIO3</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">REG_PHY_WRLVL_INIT_RATIO2</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
</table>

Table 9-325. EMIF4D_EXT_PHY_CTRL_32_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	REG_PHY_WRLVL_INIT_RATIO3	R/W	0h	Write levelling init ratio3
15-10	RESERVED	R-0	0h	
9-0	REG_PHY_WRLVL_INIT_RATIO2	R/W	0h	Write levelling init ratio2

9.3.4.158 EMIF4D_EXT_PHY_CTRL_33 Register (Offset = 300h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_33 is shown in Figure 9-300 and described in Table 9-326.

Return to Summary Table.

Figure 9-300. EMIF4D_EXT_PHY_CTRL_33 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						REG_PHY_WRLVL_INIT_RATIO5									
R-0-0h						R/W-0h									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						REG_PHY_WRLVL_INIT_RATIO4									
R-0-0h						R/W-0h									

Table 9-326. EMIF4D_EXT_PHY_CTRL_33 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	REG_PHY_WRLVL_INIT_RATIO5	R/W	0h	Write levelling init ratio5
15-10	RESERVED	R-0	0h	
9-0	REG_PHY_WRLVL_INIT_RATIO4	R/W	0h	Write levelling init ratio4

9.3.4.159 EMIF4D_EXT_PHY_CTRL_33_SHADOW Register (Offset = 304h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_33_SHADOW is shown in Figure 9-301 and described in Table 9-327.

Return to Summary Table.

Figure 9-301. EMIF4D_EXT_PHY_CTRL_33_SHADOW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">REG_PHY_WRLVL_INIT_RATIO5</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">REG_PHY_WRLVL_INIT_RATIO4</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
</table>

Table 9-327. EMIF4D_EXT_PHY_CTRL_33_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	REG_PHY_WRLVL_INIT_RATIO5	R/W	0h	Write levelling init ratio5
15-10	RESERVED	R-0	0h	
9-0	REG_PHY_WRLVL_INIT_RATIO4	R/W	0h	Write levelling init ratio4

9.3.4.160 EMIF4D_EXT_PHY_CTRL_34 Register (Offset = 308h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_34 is shown in Figure 9-302 and described in Table 9-328.

Return to Summary Table.

Figure 9-302. EMIF4D_EXT_PHY_CTRL_34 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						REG_PHY_WRLVL_INIT_RATIO7									
R-0-0h						R/W-0h									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						REG_PHY_WRLVL_INIT_RATIO6									
R-0-0h						R/W-0h									

Table 9-328. EMIF4D_EXT_PHY_CTRL_34 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	REG_PHY_WRLVL_INIT_RATIO7	R/W	0h	Write levelling init ratio7
15-10	RESERVED	R-0	0h	
9-0	REG_PHY_WRLVL_INIT_RATIO6	R/W	0h	Write levelling init ratio6

9.3.4.161 EMIF4D_EXT_PHY_CTRL_34_SHADOW Register (Offset = 30Ch) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_34_SHADOW is shown in Figure 9-303 and described in Table 9-329.

Return to Summary Table.

Figure 9-303. EMIF4D_EXT_PHY_CTRL_34_SHADOW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">REG_PHY_WRLVL_INIT_RATIO7</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">REG_PHY_WRLVL_INIT_RATIO6</td></tr>
<tr><td colspan="6">R-0-0h</td><td colspan="10">R/W-0h</td></tr>
</table>

Table 9-329. EMIF4D_EXT_PHY_CTRL_34_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	REG_PHY_WRLVL_INIT_RATIO7	R/W	0h	Write levelling init ratio7
15-10	RESERVED	R-0	0h	
9-0	REG_PHY_WRLVL_INIT_RATIO6	R/W	0h	Write levelling init ratio6

9.3.4.162 EMIF4D_EXT_PHY_CTRL_35 Register (Offset = 310h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_35 is shown in Figure 9-304 and described in Table 9-330.

Return to Summary Table.

Figure 9-304. EMIF4D_EXT_PHY_CTRL_35 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						REG_PHY_WRLVL_INIT_RATIO9									
R-0-0h						R/W-0h									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						REG_PHY_WRLVL_INIT_RATIO8									
R-0-0h						R/W-0h									

Table 9-330. EMIF4D_EXT_PHY_CTRL_35 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	REG_PHY_WRLVL_INIT_RATIO9	R/W	0h	Write levelling init ratio9
15-10	RESERVED	R-0	0h	
9-0	REG_PHY_WRLVL_INIT_RATIO8	R/W	0h	Write levelling init ratio8

9.3.4.163 EMIF4D_EXT_PHY_CTRL_35_SHADOW Register (Offset = 314h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_35_SHADOW is shown in Figure 9-305 and described in Table 9-331.

Return to Summary Table.

Figure 9-305. EMIF4D_EXT_PHY_CTRL_35_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						REG_PHY_WRLVL_INIT_RATIO9									
R-0-0h						R/W-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						REG_PHY_WRLVL_INIT_RATIO8									
R-0-0h						R/W-0h									

Table 9-331. EMIF4D_EXT_PHY_CTRL_35_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	
25-16	REG_PHY_WRLVL_INIT_RATIO9	R/W	0h	Write levelling init ratio9
15-10	RESERVED	R-0	0h	
9-0	REG_PHY_WRLVL_INIT_RATIO8	R/W	0h	Write levelling init ratio8

9.3.4.164 EMIF4D_EXT_PHY_CTRL_36 Register (Offset = 318h) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_36 is shown in Figure 9-306 and described in Table 9-332.

Return to Summary Table.

Figure 9-306. EMIF4D_EXT_PHY_CTRL_36 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="5">RESERVED</td><td>REG_PHY_RDC_FIFO_RST_ERR_CNT_CLR</td><td>REG_PHY_MDLL_UNLOCK_CLR</td><td>REG_PHY_FIFO_WE_IN_MISALIGNED_CLR</td></tr>
<tr><td colspan="5">R-0-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">REG_PHY_WRLVL_NUM_OF_DQ0</td><td colspan="4">REG_PHY_GATELVL_NUM_OF_DQ0</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 9-332. EMIF4D_EXT_PHY_CTRL_36 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	
10	REG_PHY_RDC_FIFO_RST_ERR_CNT_CLR	R/W	0h	RDC FIFO Reset Error Count Clear
9	REG_PHY_MDLL_UNLOCK_CLR	R/W	0h	MDLL Unlock Clear
8	REG_PHY_FIFO_WE_IN_MISALIGNED_CLR	R/W	0h	FIFO Write Enable In Misaligned Clear
7-4	REG_PHY_WRLVL_NUM_OF_DQ0	R/W	0h	Write levelling number of DQ0
3-0	REG_PHY_GATELVL_NUM_OF_DQ0	R/W	0h	Gate levelling number of DQ0

9.3.4.165 EMIF4D_EXT_PHY_CTRL_36_SHADOW Register (Offset = 31Ch) [reset = 0h]

EMIF4D_EXT_PHY_CTRL_36_SHADOW is shown in Figure 9-307 and described in Table 9-333.

Return to Summary Table.

Figure 9-307. EMIF4D_EXT_PHY_CTRL_36_SHADOW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="5">RESERVED</td><td>REG_PHY_RDC_FIFO_RST_ERR_CNT_CLR</td><td>REG_PHY_MDLL_UNLOCK_CLR</td><td>REG_PHY_FIFO_WE_IN_MISALIGNED_CLR</td></tr>
<tr><td colspan="5">R-0-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">REG_PHY_WRLVL_NUM_OF_DQ0</td><td colspan="4">REG_PHY_GATELVL_NUM_OF_DQ0</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 9-333. EMIF4D_EXT_PHY_CTRL_36_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	
10	REG_PHY_RDC_FIFO_RST_ERR_CNT_CLR	R/W	0h	RDC FIFO Reset Error Count Clear
9	REG_PHY_MDLL_UNLOCK_CLR	R/W	0h	MDLL Unlock Clear
8	REG_PHY_FIFO_WE_IN_MISALIGNED_CLR	R/W	0h	FIFO Write Enable In Misaligned Clear
7-4	REG_PHY_WRLVL_NUM_OF_DQ0	R/W	0h	Write levelling number of DQ0
3-0	REG_PHY_GATELVL_NUM_OF_DQ0	R/W	0h	Gate levelling number of DQ0

9.4 ELM

9.4.1 Introduction

Non-managed NAND flash memories can be dense and nonvolatile in their own nature, but error-prone. When reading from NAND flash memories, some level of error-correction is required. In the case of NAND modules with no internal correction capability, sometimes referred to as bare NANDs, the correction process is delegated to the memory controller.

The general-purpose memory controller (GPMC) probes data read from an external NAND flash and uses this to compute checksum-like information, called syndrome polynomials, on a per-block basis. Each syndrome polynomial gives a status of the read operations for a full block, including 512 bytes of data, parity bits, and an optional spare-area data field, with a maximum block size of 1023 bytes. Computation is based on a Bose-ChaudhurI-Hocquenghem (BCH) algorithm. The error-location module (ELM) extracts error addresses from these syndrome polynomials.

Based on the syndrome polynomial value, the ELM can detect errors, compute the number of errors, and give the location of each error bit. The actual data is not required to complete the error-correction algorithm. Errors can be reported anywhere in the NAND flash block, including in the parity bits.

The maximum acceptable number of errors that can be corrected depends on a programmable configuration parameter. 4-, 8-, and 16-bit error-correction levels are supported. The ELM relies on a static and fixed definition of the generator polynomial for each error-correction level that corresponds to the generator polynomials defined in the GPMC (there are three fixed polynomial for the three correction error levels). A larger number of errors than the programmed error-correction level may be detected, but the ELM cannot correct them all. The offending block is then tagged as uncorrectable in the associated computation exit status register. If the computation is successful, that is, if the number of errors detected does not exceed the maximum value authorized for the chosen correction capability, the exit status register contains the information on the number of detected errors.

When the error-location process completes, an interrupt is triggered to inform the central processing unit (CPU) that its status can be checked. The number of detected errors and their locations in the NAND block can be retrieved from the module through register accesses.

9.4.1.1 ELM Features

The ELM has the following features:

- 4, 8, and 16 bits per 512-byte block error-location based on BCH algorithms
- Eight simultaneous processing contexts
- Page-based and continuous modes
- Interrupt generation on error-location process completion:
  - When the full page has been processed in page mode
  - For each syndrome polynomial in continuous mode

9.4.1.2 Unsupported ELM Features

There are no unsupported ELM features in this device.

9.4.2 Integration

The error location module (ELM) is used to extract error addresses from syndrome polynomials generated using a BCH algorithm. Each of these polynomials gives a status of the read operations for a 512 bytes block from a NAND flash and its associated BCH parity bits, plus optionally some spare area information.

The ELM is intended to be used in conjunction with the GPMC. Syndrome polynomials generated on-the-fly when reading a NAND Flash page and stored in GPMC registers are passed to the ELM module. The MPU can then easily correct the data block by flipping the bits pointed to by the ELM error locations outputs.

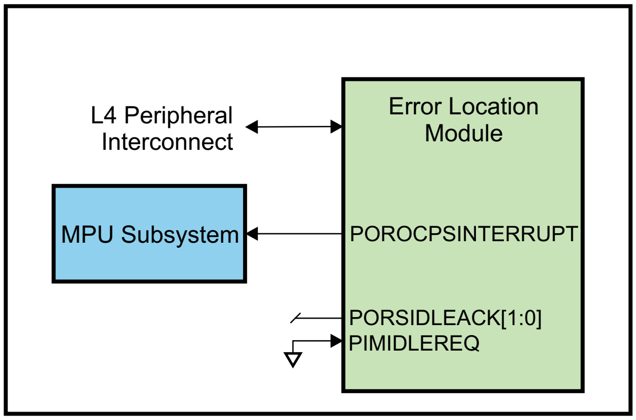


Figure 9-308. ELM Integration

9.4.2.1 ELM Connectivity Attributes

The general connectivity for the ELM module in this device is summarized in Table 9-334.

Table 9-334. ELM Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	L4PER_L4LS_GCLK
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	1 interrupt to MPU Subsystem
DMA Requests	None
Physical Address	L4 Peripheral slave port

9.4.2.2 ELM Clock and Reset Management

The ELM operates from a single OCP interface clock.

Table 9-335. ELM Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
Functional/interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk From PRCM

9.4.2.3 ELM Pin List

The ELM module does not include any external interface pins.

9.4.3 Functional Description

The ELM is designed around the error-location engine, which handles the computation based on the input syndrome polynomials.

The ELM maps the error-location engine to a standard interconnect interface by using a set of registers to control inputs and outputs.

9.4.3.1 ELM Software Reset

To perform a software reset, write a 1 to the ELM_SYSCONFIG[1] SOFTRESET bit. The ELM_SYSSTS[0] RESETDONE bit indicates that the software reset is complete when its value is 1. When the software reset completes, the ELM_SYSCONFIG[1] SOFTRESET bit is automatically reset.

9.4.3.2 ELM Power Management

Table 9-336 describes the power-management features available to the ELM module.

Table 9-336. Local Power Management Features

Feature	Registers	Description
Clock autogating	ELM_SYSCONFIG[0] AUTOGATING bit	This bit allows a local power optimization inside the module by gating the ELM_FCLK clock upon the interface activity.
Slave idle modes	ELM_SYSCONFIG[4:3] SIDLEMODE bit field	Force-idle, No-idle, and Smart-idle modes are available.
Clock activity	ELM_SYSCONFIG[8] CLOCKACTIVITY bit	The clock can be switched-off or maintained during the wake-up period.
Master Standby modes	N/A	
Global Wake-up Enable	N/A	
Wake-up Sources Enable	N/A	

CAUTION

The PRCM module has no hardware means of reading CLOCKACTIVITY settings. Thus, software must ensure consistent programming between the ELM CLOCKACTIVITY and ELM clock PRCM control bits.

9.4.3.3 ELM Interrupt Requests

Table 9-337 lists the event flags, and their masks, that can cause module interrupts.

Table 9-337. Events

Event Flag	Event Mask	Map to	Description
ELM_IRQSTS[8] PAGE_VALID	ELM_IRQEN[8] PAGE_MASK	ELM_IRQ	Page interrupt
ELM_IRQSTS[7] LOC_VALID_7	ELM_IRQEN[7] LOCATION_MASK_7	ELM_IRQ	Error-location interrupt for syndrome polynomial 7
ELM_IRQSTS[6] LOC_VALID_6	ELM_IRQEN[6] LOCATION_MASK_6	ELM_IRQ	Error-location interrupt for syndrome polynomial 6
ELM_IRQSTS[5] LOC_VALID_5	ELM_IRQEN[5] LOCATION_MASK_5	ELM_IRQ	Error-location interrupt for syndrome polynomial 5
ELM_IRQSTS[4] LOC_VALID_4	ELM_IRQEN[4] LOCATION_MASK_4	ELM_IRQ	Error-location interrupt for syndrome polynomial 4
ELM_IRQSTS[3] LOC_VALID_3	ELM_IRQEN[3] LOCATION_MASK_3	ELM_IRQ	Error-location interrupt for syndrome polynomial 3
ELM_IRQSTS[2] LOC_VALID_2	ELM_IRQEN[2] LOCATION_MASK_2	ELM_IRQ	Error-location interrupt for syndrome polynomial 2

Table 9-337. Events (continued)

Event Flag	Event Mask	Map to	Description
ELM_IRQSTS[1] LOC_VALID_1	ELM_IRQEN[1] LOCATION_MASK_1	ELM_IRQ	Error-location interrupt for syndrome polynomial 1
ELM_IRQSTS[0] LOC_VALID_0	ELM_IRQEN[0] LOCATION_MASK_0	ELM_IRQ	Error-location interrupt for syndrome polynomial 0

9.4.3.4 Processing Initialization

ELM_LOCATION_CONFIG global setting parameters must be set before using the error-location engine. The ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL bit defines the error-correction level used (4-,8-, or 16-bit error-correction). The ELM_LOCATION_CONFIG[26:16] ECC_SIZE bit field defines the maximum buffer length beyond which the engine processing no longer looks for errors.

The CPU can choose to use the ELM in continuous mode or page mode. If all ELM_PAGE_CTRL[i] SECTOR_i bits are reset (i is the syndrome polynomial number, i = 0 to 7), continuous mode is used. In any other case, page mode is implicitly selected.

- Continuous mode: Each syndrome polynomial is processed independently – results for a syndrome can be retrieved and acknowledged at any time, whatever the status of the other seven processing contexts.
- Page mode: Syndrome polynomials are grouped into atomic entities – only one page can be processed at any given time, even if all eight contexts are not used for this page. Unused contexts are lost and cannot be affected to any other processing. The full page must be acknowledged and cleared before moving to the next page.

For completion interrupts to be generated correctly, all ELM_IRQEN[i] LOCATION_MASK_i bits (i = 0 to 7) must be forced to 0 when in page mode, and set to 1 in continuous mode. Additionally, the ELM_IRQEN[8] PAGE_MASK bit must be set to 1 when in page mode.

The CPU initiates error-location processing by writing a syndrome polynomial into one of the eight possible register sets. Each of these register sets includes seven registers: ELM_SYNDROME_FRAGMENT_0_i to ELM_SYNDROME_FRAGMENT_6_i. The first six registers can be written in any order, but ELM_SYNDROME_FRAGMENT_6_i must be written last because it includes the validity bit, which instructs the ELM that this syndrome polynomial must be processed (the ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID bit).

As soon as one validity bit is asserted (ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID = 0x1, with i = 0 to 7), error-location processing can start for the corresponding syndrome polynomial. The associated ELM_LOCATION_STS_i and ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i registers are not reset (i = 0 to 7). The software must not consider them until the corresponding ELM_IRQSTS[i] LOC_VALID_i bit is set.

9.4.3.5 Processing Sequence

While the error-location engine is busy processing one syndrome polynomial, further syndrome polynomials can be written. They are processed when the current processing completes.

The engine completes early when:

- No error is detected; that is, when the ELM_LOCATION_STS_i[8] ECC_CORRECTABLE bit is set to 1 and the ELM_LOCATION_STS_i[4:0] ECC_NB_ERRORS bit field is set to 0x0.
- Too many errors are detected; that is, when the ELM_LOCATION_STS_i[8] ECC_CORRECTABLE bit is set to 0 while the ELM_LOCATION_STS_i[4:0] ECC_NB_ERRORS bit field is set with the value output by the error-location engine. The reported number of errors is not ensured if ECC_CORRECTABLE is 0.

If the engine completes early, the associated error-location registers ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i are not updated (i = 0 to 7).

In all other cases, the engine goes through the entire error-location process. Each time an error-location is found, it is logged in the associated ECC_ERROR_LOCATION bit field. The first error detected is logged in the ELM_ERROR_LOCATION_0_i[12:0] ECC_ERROR_LOCATION bit field; the second in the ELM_ERROR_LOCATION_1_i[12:0] ECC_ERROR_LOCATION bit field, and so on.

Table 9-338. ELM_LOCATION_STS_i Value Decoding Table

ECC_CORRECTABLE Value	ECC_NB_ERRORS Value	Status	Number of Errors Detected	Action Required
1	0	OK	0	None
1	≠ 0	OK	ECC_NB_ERRORS	Correct the data buffer read based on the ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i results.
0	Any	Failed	Unknown	Software-dependant

9.4.3.6 Processing Completion

When the processing for a given syndrome polynomial completes, its ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID bit is reset. It must not be set again until the exit status registers, ELM_LOCATION_STS_i (i = 0 to 7), for this processing are checked. Failure to comply with this rule leads to potential loss of the first polynomial process data output.

The error-location engine signals the process completion to the ELM. When this event is detected, the corresponding ELM_IRQSTS[i] LOC_VALID_i bit is set (i = 0 to 7). The processing-exit status is available from the associated ELM_LOCATION_STS_i register, and error locations are stored in order in the ECC_ERROR_LOCATION fields. The software must only read valid error-location registers based on the number of errors detected and located.

Immediately after the error-location engine completes, a new syndrome polynomial can be processed, if any is available, as reported by the ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID validity bit, depending on the configured error-correction level. If several syndrome polynomials are available, a round-robin arbitration is used to select one for processing.

In continuous mode (that is, all bits in ELM_PAGE_CTRL are reset), an interrupt is triggered whenever a ELM_IRQSTS[i] LOC_VALID_i bit is asserted. The CPU must read the ELM_IRQSTS register to determine which polynomial is processed and retrieve the exit status and error locations (ELM_LOCATION_STS_i and ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i). When done, the CPU must clear the corresponding ELM_IRQSTS[i] LOC_VALID_i bit by writing it to 1. Other status bits must be written to 0 so that other interrupts are not unintentionally cleared. When using this mode, the ELM_IRQSTS[8] PAGE_VALID interrupt is never triggered.

In page mode, the module does not trigger interrupts for the processing completion of each polynomial because the ELM_IRQEN[i] LOCATION_MASK_i bits are cleared. A page is defined using the ELM_PAGE_CTRL register. Each SECTOR_i bit set means the corresponding polynomial i is part of the page processing. A page is fully processed when all tagged polynomials have been processed, as logged in the ELM_IRQSTS[i] LOC_VALID_i bit fields. The module triggers an ELM_IRQSTS[8] PAGE_VALID interrupt whenever it detects that the full page has been processed. To make sure the next page can be correctly processed, all status bits in the ELM_IRQSTS register must be cleared by using a single atomic-write access.

NOTE: Do not modify page setting parameters in the ELM_PAGE_CTRL register unless the engine is idle, no polynomial input is valid, and all interrupts have been cleared.

Because no polynomial-level interrupt is triggered in page mode, polynomials cleared in the ELM_PAGE_CTRL[i] SECTOR_i bit fields (i = 0 to 7) are processed as usual, but are essentially ignored. The CPU must manually poll the ELM_IRQSTS bits to check for their status.

9.4.4 Basic Programming Model

9.4.4.1 ELM Low Level Programming Model

9.4.4.1.1 Processing Initialization

Table 9-339. ELM Processing Initialization

Step	Register/ Bit Field / Programming Model	Value
Resets the module	ELM_SYSCONFIG[1] SOFTRESET	0x1
Wait until reset is done.	ELM_SYSSTS[0] RESETDONE	0x1
Configure the slave interface power management.	ELM_SYSCONFIG[4:3] SIDLEMODE	Set value
Defines the error-correction level used	ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL	Set value
Defines the maximum buffer length	ELM_LOCATION_CONFIG[26:16] ECC_SIZE	Set value
Sets the ELM in continuous mode or page mode	ELM_PAGE_CTRL	Set value
If continuous mode is used	All ELM_PAGE_CTRL[i] SECTOR_i (i = 0 to 7)	0x0
Enables interrupt for syndrome polynomial i	ELM_IRQEN[i] LOCATION_MASK_i	0x1
else (page mode is used)	One syndrome polynomial i is set ELM_PAGE_CTRL[i] SECTOR_i (i = 0 to 7)	0x1
Disable all interrupts for syndrome polynomial and enable PAGE_MASK interrupt.	All ELM_IRQEN[i] LOCATION_MASK_i = 0x0 and ELM_IRQEN[8] PAGE_MASK = 0x1	Set value
endif		Set value
Set the input syndrome polynomial i.	ELM_SYNDROME_FRAGMENT_0_i	Set value
	ELM_SYNDROME_FRAGMENT_1_i	Set value
	ELM_SYNDROME_FRAGMENT_5_i	Set value
	ELM_SYNDROME_FRAGMENT_6_i	Set value
Initiates the computation process	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID	0x1

9.4.4.1.2 Read Results

The engine goes through the entire error-location process and results can be read. Table 9-340 and Table 9-341 describe the processing completion for continuous and page modes, respectively.

Table 9-340. ELM Processing Completion for Continuous Mode

Step	Register/ Bit Field / Programming Model	Value
Wait until process is complete for syndrome polynomial i: Wait until the ELM_IRQ interrupt is generated, or poll the status register.		
Read for which i the error-location process is complete.	ELM_IRQSTS[i] LOC_VALID_i	0x1
if the process fails (too many errors)	ELM_LOCATION_STS_i[8] ECC_CORRECTABLE	0x0
It is software dependant.		
else (process successful, the engine completes)	ELM_LOCATION_STS_i[8] ECC_CORRECTABLE	0x1
Read the number of errors.	ELM_LOCATION_STS_i[4:0] ECC_NB_ERRORS	
Read the error-location bit addresses for syndrome polynomial i of the ECC_NB_ERRORS first registers. It is the software responsibility to correct errors in the data buffer.	ELM_ERROR_LOCATION_0_i[12:0] ECC_ERROR_LOCATION	
	ELM_ERROR_LOCATION_1_i[12:0] ECC_ERROR_LOCATION	
	...	
	ELM_ERROR_LOCATION_15_i[12:0] ECC_ERROR_LOCATION	

Table 9-340. ELM Processing Completion for Continuous Mode (continued)

Step	Register/ Bit Field / Programming Model	Value
endif		
Clear the corresponding i interrupt.	ELM_IRQSTS[i] LOC_VALID_i	0x1

A new syndrome polynomial can be processed after the end of processing (ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID = 0x0) and after the exit status register check (ELM_LOCATION_STS_i).

Table 9-341. ELM Processing Completion for Page Mode

Step	Register/ Bit Field / Programming Model	Value
Wait until process is complete for syndrome polynomial i: Wait until the ELM_IRQ interrupt is generated, or poll the status register.		
Wait for page completed interrupt: All error locations are valid.	ELM_IRQSTS[8] PAGE_VALID	0x1
Repeat the following actions the necessary number of times. That is, once for each valid defined block in the page.		
Read the process exit status.	ELM_LOCATION_STS_i[8] ECC_CORRECTABLE	
if the process fails (too many errors)	ELM_LOCATION_STS_i[8] ECC_CORRECTABLE	0x0
It is software dependant.		
else (process successful, the engine completes)	ELM_LOCATION_STS_i[8] ECC_CORRECTABLE	0x1
Read the number of errors.	ELM_LOCATION_STS_i[4:0] ECC_NB_ERRORS	
Read the error-location bit addresses for syndrome polynomial i of the ECC_NB_ERRORS first registers.	ELM_ERROR_LOCATION_0_i[12:0] ECC_ERROR_LOCATION	
	ELM_ERROR_LOCATION_1_i[12:0] ECC_ERROR_LOCATION	
	...	
	ELM_ERROR_LOCATION_15_i[12:0] ECC_ERROR_LOCATION	
endif		
End Repeat		
Clear the ELM_IRQSTS register.	ELM_IRQSTS	0x1FF

Next page can be correctly processed after a page is fully processed, when all tagged polynomials have been processed (ELM_IRQSTS[i] LOC_VALID_i = 0x1 for all syndrome polynomials i used in the page).

9.4.4.2 Use Case: ELM Used in Continuous Mode

In this example, the ELM module is programmed for an 8-bit error-correction capability in continuous mode. After reading a 528-byte NAND flash sector (512B data plus 16B spare area) with a 16-bit interface, a non-zero polynomial syndrome is reported from the GPMC (Polynomial syndrome 0 is used in the ELM):

- P = 0x0A16ABE115E44F767BFB0D0980

Table 9-342. Use Case: Continuous Mode

Step	Register/ Bit Field / Programming Model	Value
Resets the module	ELM_SYSCONFIG[1] SOFTRESET	0x1
Wait until reset is done.	ELM_SYSSTS[0] RESETDONE	0x1
Configure the slave interface power management: Smart idle is used.	ELM_SYSCONFIG[4:3] SIDLEMODE	0x2
Defines the error-correction level used: 8 bits	ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL	0x1

Table 9-342. Use Case: Continuous Mode (continued)

Step	Register/ Bit Field / Programming Model	Value
Defines the maximum buffer length: 528 bytes (2x528 = 1056)	ELM_LOCATION_CONFIG[26:16] ECC_SIZE	0x420
Sets the ELM in continuous mode	ELM_PAGE_CTRL	0
Enables interrupt for syndrome polynomial 0	ELM_IRQEN[0] LOCATION_MASK_0	0x1
Set the input syndrome polynomial 0.	ELM_SYNDROME_FRAGMENT_0_i (i=0)	0xFB0D0980
	ELM_SYNDROME_FRAGMENT_1_i (i=0)	0xE44F767B
	ELM_SYNDROME_FRAGMENT_2_i (i=0)	0x16ABE115
	ELM_SYNDROME_FRAGMENT_3_i (i=0)	0x0000000A
Initiates the computation process	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (i=0)	0x1
Wait until process is complete for syndrome polynomial 0: IRQ_ELM is generated or poll the status register.		
Read that error-location process is complete for syndrome polynomial 0.	ELM_IRQSTS[0] LOC_VALID_0	0x1
Read the process exit status: All errors were successfully located.	ELM_LOCATION_STS_i[8] ECC_CORRECTABLE (i=0)	0x1
Read the number of errors: Four errors detected.	ELM_LOCATION_STS_i[4:0] ECC_NB_ERRORS (i=0)	0x4
Read the error-location bit addresses for syndrome polynomial 0 of the 4 first registers: Errors are located in the data buffer at decimal addresses 431, 1062, 1909, 3452.	ELM_ERROR_LOCATION_0_i (i=0)	0x1AF
	ELM_ERROR_LOCATION_1_i (i=0)	0x426
	ELM_ERROR_LOCATION_2_i (i=0)	0x775
	ELM_ERROR_LOCATION_3_i (i=0)	0xD7C
Clear the corresponding interrupt for polynomial 0.	ELM_IRQSTS[0] LOC_VALID_0	0x1

The NAND flash data in the sector are seen as a polynomial of degree 4223 (number of bits in a 528 byte buffer minus 1), with each data bit being a coefficient in the polynomial. When reading from a NAND flash using the GPMC module, computation of the polynomial syndrome assumes that the first NAND word read at address 0x0 contains the highest-order coefficient in the message. Furthermore, in the 16-bit NAND word, bits are ordered from bit 7 to bit 0, then from bit 15 to bit 8. Based on this convention, an address table of the data buffer can be built. NAND memory addresses in Table 9-343 are given in decimal format.

Table 9-343. 16-bit NAND Sector Buffer Address Map

NAND Memory Address	Message bit addresses in the memory word															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	4215	4214	4213	4212	4211	4210	4209	4208	4223	4222	4221	4220	4219	4218	4217	4216
1	4175	4174	4173	4172	4171	4170	4169	4168	4183	4182	4181	4180	4179	4178	4177	4176
...																
47	3463	3462	3461	3460	3459	3458	3457	3456	3471	3470	3469	3468	3467	3466	3465	3464
48	3447	3446	3445	3444	3443	3442	3441	3440	3455	3454	3453	3452	3451	3450	3449	3448
49	3431	3430	3429	3428	3427	3426	3425	3424	3439	3438	3437	3436	3435	3434	3433	3432
50	3415	3414	3413	3412	3411	3410	3409	3408	3423	3422	3421	3420	3419	3418	3417	3416
...																
255	135	134	133	132	131	130	129	128	143	142	141	140	139	138	137	136
256	119	118	117	116	115	114	113	112	127	126	125	124	123	122	121	120
257	103	102	101	100	99	98	97	96	111	110	109	108	107	106	105	104
258	87	86	85	84	83	82	81	80	95	94	93	92	91	90	89	88
259	71	70	69	68	67	66	65	64	79	78	77	76	75	74	73	72
260	55	54	53	52	51	50	49	48	63	62	61	60	59	58	57	56
261	39	38	37	36	35	34	33	32	47	46	45	44	43	42	41	40
262	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24
263	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8

The table can now be used to determine which bits in the buffer were incorrect and must be flipped. In this example, the first bit to be flipped is bit 4 from the 49th byte read from memory. It is up to the processor to correctly map this word to the copied buffer and to flip this bit. The same process must be repeated for all detected errors.

9.4.4.3 Use Case: ELM Used in Page Mode

In this example, the ELM module is programmed for an 16-bit error-correction capability in page mode. After reading a 528-byte NAND flash sector (512B data plus 16B spare area) with a 16-bit interface, four non-zero polynomial syndromes are reported from the GPMC (Polynomial syndrome 0, 1, 2, and 3 are used in the ELM):

- P0 = 0xE8B0 12ADDB5A318E05BE B0693DB28330B5CC A329AA05E0B718EF
- P1 = 0xBAD0 49A0D932C22E6669 0948DF08BE093336 79C6BA10E5F935EB
- P2 = 0x69D9 B86ABCD5EC3697FA A6498FEE54556EA0 1579EF7D60BA3189
- P3 = 0x0

Table 9-344. Use Case: Page Mode

Step	Register/ Bit Field / Programming Model	Value
Resets the module	ELM_SYSCONFIG[1] SOFTRESET	0x1
Wait until reset is done.	ELM_SYSSTS[0] RESETDONE	0x1
Configure the slave interface power management: Smart idle is used.	ELM_SYSCONFIG[4:3] SIDLEMODE	0x2
Defines the error-correction level used: 16 bits	ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL	0x2
Defines the maximum buffer length: 528 bytes	ELM_LOCATION_CONFIG[26:16] ECC_SIZE	0x420
Sets the ELM in page mode (4 blocks in a page)	ELM_PAGE_CTRL[0] SECTOR_0	0x1
	ELM_PAGE_CTRL[1] SECTOR_1	0x1
	ELM_PAGE_CTRL[2] SECTOR_2	0x1
	ELM_PAGE_CTRL[3] SECTOR_3	0x1

Table 9-344. Use Case: Page Mode (continued)

Step	Register/ Bit Field / Programming Model	Value
Disable all interrupts for syndrome polynomial and enable PAGE_MASK interrupt.	ELM_IRQEN	0x100
Set the input syndrome polynomial 0.	ELM_SYNDROME_FRAGMENT_0_i (i=0)	0xE0B718EF
	ELM_SYNDROME_FRAGMENT_1_i (i=0)	0xA329AA05
	ELM_SYNDROME_FRAGMENT_2_i (i=0)	0x8330B5CC
	ELM_SYNDROME_FRAGMENT_3_i (i=0)	0xB0693DB2
	ELM_SYNDROME_FRAGMENT_4_i (i=0)	0x318E05BE
	ELM_SYNDROME_FRAGMENT_5_i (i=0)	0x12ADDB5A
	ELM_SYNDROME_FRAGMENT_6_i (i=0)	0xE8B0
Set the input syndrome polynomial 1.	ELM_SYNDROME_FRAGMENT_0_i (i=1)	0xE5F935EB
	ELM_SYNDROME_FRAGMENT_1_i (i=1)	0x79C6BA10
	ELM_SYNDROME_FRAGMENT_2_i (i=1)	0xBE093336
	ELM_SYNDROME_FRAGMENT_3_i (i=1)	0x0948DF08
	ELM_SYNDROME_FRAGMENT_4_i (i=1)	0xC22E6669
	ELM_SYNDROME_FRAGMENT_5_i (i=1)	0x49A0D932
	ELM_SYNDROME_FRAGMENT_6_i (i=1)	0xBAD0
Set the input syndrome polynomial 2.	ELM_SYNDROME_FRAGMENT_0_i (i=2)	0x60BA3189
	ELM_SYNDROME_FRAGMENT_1_i (i=2)	0x1579EF7D
	ELM_SYNDROME_FRAGMENT_2_i (i=2)	0x54556EA0
	ELM_SYNDROME_FRAGMENT_3_i (i=2)	0xA6498FEE
	ELM_SYNDROME_FRAGMENT_4_i (i=2)	0xEC3697FA
	ELM_SYNDROME_FRAGMENT_5_i (i=2)	0xB86ABCD5
	ELM_SYNDROME_FRAGMENT_6_i (i=2)	0x69D9
Set the input syndrome polynomial 3.	ELM_SYNDROME_FRAGMENT_0_i (i=3)	0x0
	ELM_SYNDROME_FRAGMENT_1_i (i=3)	0x0
	ELM_SYNDROME_FRAGMENT_2_i (i=3)	0x0
	ELM_SYNDROME_FRAGMENT_3_i (i=3)	0x0
	ELM_SYNDROME_FRAGMENT_4_i (i=3)	0x0
	ELM_SYNDROME_FRAGMENT_5_i (i=3)	0x0
	ELM_SYNDROME_FRAGMENT_6_i (i=3)	0x0
Initiates the computation process for syndrome polynomial 0	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (i=0)	0x1
Initiates the computation process for syndrome polynomial 1	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (i=1)	0x1
Initiates the computation process for syndrome polynomial 2	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (i=2)	0x1
Initiates the computation process for syndrome polynomial 3	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (i=3)	0x1
Wait until process is complete for syndrome polynomial 0, 1, 2, and 3: Wait until the ELM_IRQ interrupt is generated or poll the status register.		
Wait for page completed interrupt: All error locations are valid.	ELM_IRQSTS[8] PAGE_VALID	0x1
Read the process exit status for syndrome polynomial 0: All errors were successfully located.	ELM_LOCATION_STS_i[8] ECC_CORRECTABLE (i=0)	0x1
Read the process exit status for syndrome polynomial 1: All errors were successfully located.	ELM_LOCATION_STS_i[8] ECC_CORRECTABLE (i=1)	0x1

Table 9-344. Use Case: Page Mode (continued)

Step	Register/ Bit Field / Programming Model	Value
Read the process exit status for syndrome polynomial 2: All errors were successfully located.	ELM_LOCATION_STS_i[8] ECC_CORRECTABLE (i=2)	0x1
Read the process exit status for syndrome polynomial 3: All errors were successfully located.	ELM_LOCATION_STS_i[8] ECC_CORRECTABLE (i=3)	0x1
Read the number of errors for syndrome polynomial 0: 4 errors detected.	ELM_LOCATION_STS_i[4:0] ECC_NB_ERRORS (i=0)	0x4
Read the number of errors for syndrome polynomial 1: 2 errors detected.	ELM_LOCATION_STS_i[4:0] ECC_NB_ERRORS (i=1)	0x2
Read the number of errors for syndrome polynomial 2: 1 error detected.	ELM_LOCATION_STS_i[4:0] ECC_NB_ERRORS (i=2)	0x1
Read the number of errors for syndrome polynomial 3: 0 errors detected.	ELM_LOCATION_STS_i[4:0] ECC_NB_ERRORS (i=3)	0x0
Read the error-location bit addresses for syndrome polynomial 0 of the 4 first registers:	ELM_ERROR_LOCATION_0_i (i=0)	0x1FE
	ELM_ERROR_LOCATION_1_i (i=0)	0x617
	ELM_ERROR_LOCATION_2_i (i=0)	0x650
	ELM_ERROR_LOCATION_3_i (i=0)	0xA83
Read the error-location bit addresses for syndrome polynomial 1 of the 2 first registers:	ELM_ERROR_LOCATION_0_i (i=1)	0x104E
	ELM_ERROR_LOCATION_1_i (i=1)	0x4
Read the errors location bit addresses for syndrome polynomial 2 of the first registers:	ELM_ERROR_LOCATION_0_i (i=1)	0x3E8
Clear the ELM_IRQSTS register.	ELM_IRQSTS	0x1FF

9.4.5 *ELM Registers*

Table 9-345 lists the memory-mapped registers for the ELM. All register offset addresses not listed in Table 9-345 should be considered as reserved locations and the register contents should not be modified.

Table 9-345. ELM Registers

Offset	Acronym	Register Name	Section
0h	ELM_REVISION	ELM Revision Register	Section 9.4.5.1
10h	ELM_SYSCONFIG	ELM System Configuration Register	Section 9.4.5.2
14h	ELM_SYSSTS	ELM System Status Register	Section 9.4.5.3
18h	ELM_IRQSTS	ELM Interrupt Status Register	Section 9.4.5.4
1Ch	ELM_IRQEN	ELM Interrupt Enable Register	Section 9.4.5.5
20h	ELM_LOCATION_CONFIG	ELM Location Configuration Register	Section 9.4.5.6
80h	ELM_PAGE_CTRL	ELM Page Definition Register	Section 9.4.5.7
400h	ELM_SYNDROME_FRAGMENT_0_0	ELM_SYNDROME_FRAGMENT_0_i Register	Section 9.4.5.8
404h	ELM_SYNDROME_FRAGMENT_1_0	ELM_SYNDROME_FRAGMENT_1_i Register	Section 9.4.5.9
408h	ELM_SYNDROME_FRAGMENT_2_0	ELM_SYNDROME_FRAGMENT_2_i Register	Section 9.4.5.10
40Ch	ELM_SYNDROME_FRAGMENT_3_0	ELM_SYNDROME_FRAGMENT_3_i Register	Section 9.4.5.11
410h	ELM_SYNDROME_FRAGMENT_4_0	ELM_SYNDROME_FRAGMENT_4_i Register	Section 9.4.5.12
414h	ELM_SYNDROME_FRAGMENT_5_0	ELM_SYNDROME_FRAGMENT_5_i Register	Section 9.4.5.13
418h	ELM_SYNDROME_FRAGMENT_6_0	ELM_SYNDROME_FRAGMENT_6_i Register	Section 9.4.5.14
440h	ELM_SYNDROME_FRAGMENT_0_1	ELM_SYNDROME_FRAGMENT_0_i Register	Section 9.4.5.8
444h	ELM_SYNDROME_FRAGMENT_1_1	ELM_SYNDROME_FRAGMENT_1_i Register	Section 9.4.5.9
448h	ELM_SYNDROME_FRAGMENT_2_1	ELM_SYNDROME_FRAGMENT_2_i Register	Section 9.4.5.10

Table 9-345. ELM Registers (continued)

Offset	Acronym	Register Name	Section
44Ch	ELM_SYNDROME_FRAGMENT_3_1	ELM_SYNDROME_FRAGMENT_3_i Register	Section 9.4.5.11
450h	ELM_SYNDROME_FRAGMENT_4_1	ELM_SYNDROME_FRAGMENT_4_i Register	Section 9.4.5.12
454h	ELM_SYNDROME_FRAGMENT_5_1	ELM_SYNDROME_FRAGMENT_5_i Register	Section 9.4.5.13
458h	ELM_SYNDROME_FRAGMENT_6_1	ELM_SYNDROME_FRAGMENT_6_i Register	Section 9.4.5.14
480h	ELM_SYNDROME_FRAGMENT_0_2	ELM_SYNDROME_FRAGMENT_0_i Register	Section 9.4.5.8
484h	ELM_SYNDROME_FRAGMENT_1_2	ELM_SYNDROME_FRAGMENT_1_i Register	Section 9.4.5.9
488h	ELM_SYNDROME_FRAGMENT_2_2	ELM_SYNDROME_FRAGMENT_2_i Register	Section 9.4.5.10
48Ch	ELM_SYNDROME_FRAGMENT_3_2	ELM_SYNDROME_FRAGMENT_3_i Register	Section 9.4.5.11
490h	ELM_SYNDROME_FRAGMENT_4_2	ELM_SYNDROME_FRAGMENT_4_i Register	Section 9.4.5.12
494h	ELM_SYNDROME_FRAGMENT_5_2	ELM_SYNDROME_FRAGMENT_5_i Register	Section 9.4.5.13
498h	ELM_SYNDROME_FRAGMENT_6_2	ELM_SYNDROME_FRAGMENT_6_i Register	Section 9.4.5.14
4C0h	ELM_SYNDROME_FRAGMENT_0_3	ELM_SYNDROME_FRAGMENT_0_i Register	Section 9.4.5.8
4C4h	ELM_SYNDROME_FRAGMENT_1_3	ELM_SYNDROME_FRAGMENT_1_i Register	Section 9.4.5.9
4C8h	ELM_SYNDROME_FRAGMENT_2_3	ELM_SYNDROME_FRAGMENT_2_i Register	Section 9.4.5.10
4CCh	ELM_SYNDROME_FRAGMENT_3_3	ELM_SYNDROME_FRAGMENT_3_i Register	Section 9.4.5.11
4D0h	ELM_SYNDROME_FRAGMENT_4_3	ELM_SYNDROME_FRAGMENT_4_i Register	Section 9.4.5.12
4D4h	ELM_SYNDROME_FRAGMENT_5_3	ELM_SYNDROME_FRAGMENT_5_i Register	Section 9.4.5.13
4D8h	ELM_SYNDROME_FRAGMENT_6_3	ELM_SYNDROME_FRAGMENT_6_i Register	Section 9.4.5.14
500h	ELM_SYNDROME_FRAGMENT_0_4	ELM_SYNDROME_FRAGMENT_0_i Register	Section 9.4.5.8
504h	ELM_SYNDROME_FRAGMENT_1_4	ELM_SYNDROME_FRAGMENT_1_i Register	Section 9.4.5.9
508h	ELM_SYNDROME_FRAGMENT_2_4	ELM_SYNDROME_FRAGMENT_2_i Register	Section 9.4.5.10
50Ch	ELM_SYNDROME_FRAGMENT_3_4	ELM_SYNDROME_FRAGMENT_3_i Register	Section 9.4.5.11
510h	ELM_SYNDROME_FRAGMENT_4_4	ELM_SYNDROME_FRAGMENT_4_i Register	Section 9.4.5.12
514h	ELM_SYNDROME_FRAGMENT_5_4	ELM_SYNDROME_FRAGMENT_5_i Register	Section 9.4.5.13
518h	ELM_SYNDROME_FRAGMENT_6_4	ELM_SYNDROME_FRAGMENT_6_i Register	Section 9.4.5.14
540h	ELM_SYNDROME_FRAGMENT_0_5	ELM_SYNDROME_FRAGMENT_0_i Register	Section 9.4.5.8
544h	ELM_SYNDROME_FRAGMENT_1_5	ELM_SYNDROME_FRAGMENT_1_i Register	Section 9.4.5.9
548h	ELM_SYNDROME_FRAGMENT_2_5	ELM_SYNDROME_FRAGMENT_2_i Register	Section 9.4.5.10
54Ch	ELM_SYNDROME_FRAGMENT_3_5	ELM_SYNDROME_FRAGMENT_3_i Register	Section 9.4.5.11
550h	ELM_SYNDROME_FRAGMENT_4_5	ELM_SYNDROME_FRAGMENT_4_i Register	Section 9.4.5.12
554h	ELM_SYNDROME_FRAGMENT_5_5	ELM_SYNDROME_FRAGMENT_5_i Register	Section 9.4.5.13
558h	ELM_SYNDROME_FRAGMENT_6_5	ELM_SYNDROME_FRAGMENT_6_i Register	Section 9.4.5.14
580h	ELM_SYNDROME_FRAGMENT_0_6	ELM_SYNDROME_FRAGMENT_0_i Register	Section 9.4.5.8
584h	ELM_SYNDROME_FRAGMENT_1_6	ELM_SYNDROME_FRAGMENT_1_i Register	Section 9.4.5.9
588h	ELM_SYNDROME_FRAGMENT_2_6	ELM_SYNDROME_FRAGMENT_2_i Register	Section 9.4.5.10
58Ch	ELM_SYNDROME_FRAGMENT_3_6	ELM_SYNDROME_FRAGMENT_3_i Register	Section 9.4.5.11
590h	ELM_SYNDROME_FRAGMENT_4_6	ELM_SYNDROME_FRAGMENT_4_i Register	Section 9.4.5.12
594h	ELM_SYNDROME_FRAGMENT_5_6	ELM_SYNDROME_FRAGMENT_5_i Register	Section 9.4.5.13
598h	ELM_SYNDROME_FRAGMENT_6_6	ELM_SYNDROME_FRAGMENT_6_i Register	Section 9.4.5.14
5C0h	ELM_SYNDROME_FRAGMENT_0_7	ELM_SYNDROME_FRAGMENT_0_i Register	Section 9.4.5.8
5C4h	ELM_SYNDROME_FRAGMENT_1_7	ELM_SYNDROME_FRAGMENT_1_i Register	Section 9.4.5.9
5C8h	ELM_SYNDROME_FRAGMENT_2_7	ELM_SYNDROME_FRAGMENT_2_i Register	Section 9.4.5.10
5CCh	ELM_SYNDROME_FRAGMENT_3_7	ELM_SYNDROME_FRAGMENT_3_i Register	Section 9.4.5.11
5D0h	ELM_SYNDROME_FRAGMENT_4_7	ELM_SYNDROME_FRAGMENT_4_i Register	Section 9.4.5.12
5D4h	ELM_SYNDROME_FRAGMENT_5_7	ELM_SYNDROME_FRAGMENT_5_i Register	Section 9.4.5.13
5D8h	ELM_SYNDROME_FRAGMENT_6_7	ELM_SYNDROME_FRAGMENT_6_i Register	Section 9.4.5.14
800h	ELM_LOCATION_STS_0	ELM_LOCATION_STATUS_i Register	Section 9.4.5.15

Table 9-345. ELM Registers (continued)

Offset	Acronym	Register Name	Section
880h	ELM_ERROR_LOCATION_0_0	ELM_ERROR_LOCATION_0_i Register	Section 9.4.5.16
884h	ELM_ERROR_LOCATION_1_0	ELM_ERROR_LOCATION_1_i Register	Section 9.4.5.17
888h	ELM_ERROR_LOCATION_2_0	ELM_ERROR_LOCATION_2_i Register	Section 9.4.5.18
88Ch	ELM_ERROR_LOCATION_3_0	ELM_ERROR_LOCATION_3_i Register	Section 9.4.5.19
890h	ELM_ERROR_LOCATION_4_0	ELM_ERROR_LOCATION_4_i Register	Section 9.4.5.20
894h	ELM_ERROR_LOCATION_5_0	ELM_ERROR_LOCATION_5_i Register	Section 9.4.5.21
898h	ELM_ERROR_LOCATION_6_0	ELM_ERROR_LOCATION_6_i Register	Section 9.4.5.22
89Ch	ELM_ERROR_LOCATION_7_0	ELM_ERROR_LOCATION_7_i Register	Section 9.4.5.23
8A0h	ELM_ERROR_LOCATION_8_0	ELM_ERROR_LOCATION_8_i Register	Section 9.4.5.24
8A4h	ELM_ERROR_LOCATION_9_0	ELM_ERROR_LOCATION_9_i Register	Section 9.4.5.25
8A8h	ELM_ERROR_LOCATION_10_0	ELM_ERROR_LOCATION_10_i Register	Section 9.4.5.26
8ACh	ELM_ERROR_LOCATION_11_0	ELM_ERROR_LOCATION_11_i Register	Section 9.4.5.27
8B0h	ELM_ERROR_LOCATION_12_0	ELM_ERROR_LOCATION_12_i Register	Section 9.4.5.28
8B4h	ELM_ERROR_LOCATION_13_0	ELM_ERROR_LOCATION_13_i Register	Section 9.4.5.29
8B8h	ELM_ERROR_LOCATION_14_0	ELM_ERROR_LOCATION_14_i Register	Section 9.4.5.30
8BCh	ELM_ERROR_LOCATION_15_0	ELM_ERROR_LOCATION_15_i Register	Section 9.4.5.31
900h	ELM_LOCATION_STS_1	ELM_LOCATION_STATUS_i Register	Section 9.4.5.15
980h	ELM_ERROR_LOCATION_0_1	ELM_ERROR_LOCATION_0_i Register	Section 9.4.5.16
984h	ELM_ERROR_LOCATION_1_1	ELM_ERROR_LOCATION_1_i Register	Section 9.4.5.17
988h	ELM_ERROR_LOCATION_2_1	ELM_ERROR_LOCATION_2_i Register	Section 9.4.5.18
98Ch	ELM_ERROR_LOCATION_3_1	ELM_ERROR_LOCATION_3_i Register	Section 9.4.5.19
990h	ELM_ERROR_LOCATION_4_1	ELM_ERROR_LOCATION_4_i Register	Section 9.4.5.20
994h	ELM_ERROR_LOCATION_5_1	ELM_ERROR_LOCATION_5_i Register	Section 9.4.5.21
998h	ELM_ERROR_LOCATION_6_1	ELM_ERROR_LOCATION_6_i Register	Section 9.4.5.22
99Ch	ELM_ERROR_LOCATION_7_1	ELM_ERROR_LOCATION_7_i Register	Section 9.4.5.23
9A0h	ELM_ERROR_LOCATION_8_1	ELM_ERROR_LOCATION_8_i Register	Section 9.4.5.24
9A4h	ELM_ERROR_LOCATION_9_1	ELM_ERROR_LOCATION_9_i Register	Section 9.4.5.25
9A8h	ELM_ERROR_LOCATION_10_1	ELM_ERROR_LOCATION_10_i Register	Section 9.4.5.26
9ACh	ELM_ERROR_LOCATION_11_1	ELM_ERROR_LOCATION_11_i Register	Section 9.4.5.27
9B0h	ELM_ERROR_LOCATION_12_1	ELM_ERROR_LOCATION_12_i Register	Section 9.4.5.28
9B4h	ELM_ERROR_LOCATION_13_1	ELM_ERROR_LOCATION_13_i Register	Section 9.4.5.29
9B8h	ELM_ERROR_LOCATION_14_1	ELM_ERROR_LOCATION_14_i Register	Section 9.4.5.30
9BCh	ELM_ERROR_LOCATION_15_1	ELM_ERROR_LOCATION_15_i Register	Section 9.4.5.31
A00h	ELM_LOCATION_STS_2	ELM_LOCATION_STATUS_i Register	Section 9.4.5.15
A80h	ELM_ERROR_LOCATION_0_2	ELM_ERROR_LOCATION_0_i Register	Section 9.4.5.16
A84h	ELM_ERROR_LOCATION_1_2	ELM_ERROR_LOCATION_1_i Register	Section 9.4.5.17
A88h	ELM_ERROR_LOCATION_2_2	ELM_ERROR_LOCATION_2_i Register	Section 9.4.5.18
A8Ch	ELM_ERROR_LOCATION_3_2	ELM_ERROR_LOCATION_3_i Register	Section 9.4.5.19
A90h	ELM_ERROR_LOCATION_4_2	ELM_ERROR_LOCATION_4_i Register	Section 9.4.5.20
A94h	ELM_ERROR_LOCATION_5_2	ELM_ERROR_LOCATION_5_i Register	Section 9.4.5.21
A98h	ELM_ERROR_LOCATION_6_2	ELM_ERROR_LOCATION_6_i Register	Section 9.4.5.22
A9Ch	ELM_ERROR_LOCATION_7_2	ELM_ERROR_LOCATION_7_i Register	Section 9.4.5.23
AA0h	ELM_ERROR_LOCATION_8_2	ELM_ERROR_LOCATION_8_i Register	Section 9.4.5.24
AA4h	ELM_ERROR_LOCATION_9_2	ELM_ERROR_LOCATION_9_i Register	Section 9.4.5.25
AA8h	ELM_ERROR_LOCATION_10_2	ELM_ERROR_LOCATION_10_i Register	Section 9.4.5.26
AACh	ELM_ERROR_LOCATION_11_2	ELM_ERROR_LOCATION_11_i Register	Section 9.4.5.27
AB0h	ELM_ERROR_LOCATION_12_2	ELM_ERROR_LOCATION_12_i Register	Section 9.4.5.28

Table 9-345. ELM Registers (continued)

Offset	Acronym	Register Name	Section
AB4h	ELM_ERROR_LOCATION_13_2	ELM_ERROR_LOCATION_13_i Register	Section 9.4.5.29
AB8h	ELM_ERROR_LOCATION_14_2	ELM_ERROR_LOCATION_14_i Register	Section 9.4.5.30
ABCh	ELM_ERROR_LOCATION_15_2	ELM_ERROR_LOCATION_15_i Register	Section 9.4.5.31
B00h	ELM_LOCATION_STS_3	ELM_LOCATION_STATUS_i Register	Section 9.4.5.15
B80h	ELM_ERROR_LOCATION_0_3	ELM_ERROR_LOCATION_0_i Register	Section 9.4.5.16
B84h	ELM_ERROR_LOCATION_1_3	ELM_ERROR_LOCATION_1_i Register	Section 9.4.5.17
B88h	ELM_ERROR_LOCATION_2_3	ELM_ERROR_LOCATION_2_i Register	Section 9.4.5.18
B8Ch	ELM_ERROR_LOCATION_3_3	ELM_ERROR_LOCATION_3_i Register	Section 9.4.5.19
B90h	ELM_ERROR_LOCATION_4_3	ELM_ERROR_LOCATION_4_i Register	Section 9.4.5.20
B94h	ELM_ERROR_LOCATION_5_3	ELM_ERROR_LOCATION_5_i Register	Section 9.4.5.21
B98h	ELM_ERROR_LOCATION_6_3	ELM_ERROR_LOCATION_6_i Register	Section 9.4.5.22
B9Ch	ELM_ERROR_LOCATION_7_3	ELM_ERROR_LOCATION_7_i Register	Section 9.4.5.23
BA0h	ELM_ERROR_LOCATION_8_3	ELM_ERROR_LOCATION_8_i Register	Section 9.4.5.24
BA4h	ELM_ERROR_LOCATION_9_3	ELM_ERROR_LOCATION_9_i Register	Section 9.4.5.25
BA8h	ELM_ERROR_LOCATION_10_3	ELM_ERROR_LOCATION_10_i Register	Section 9.4.5.26
BACh	ELM_ERROR_LOCATION_11_3	ELM_ERROR_LOCATION_11_i Register	Section 9.4.5.27
BB0h	ELM_ERROR_LOCATION_12_3	ELM_ERROR_LOCATION_12_i Register	Section 9.4.5.28
BB4h	ELM_ERROR_LOCATION_13_3	ELM_ERROR_LOCATION_13_i Register	Section 9.4.5.29
BB8h	ELM_ERROR_LOCATION_14_3	ELM_ERROR_LOCATION_14_i Register	Section 9.4.5.30
BBCh	ELM_ERROR_LOCATION_15_3	ELM_ERROR_LOCATION_15_i Register	Section 9.4.5.31
C00h	ELM_LOCATION_STS_4	ELM_LOCATION_STATUS_i Register	Section 9.4.5.15
C80h	ELM_ERROR_LOCATION_0_4	ELM_ERROR_LOCATION_0_i Register	Section 9.4.5.16
C84h	ELM_ERROR_LOCATION_1_4	ELM_ERROR_LOCATION_1_i Register	Section 9.4.5.17
C88h	ELM_ERROR_LOCATION_2_4	ELM_ERROR_LOCATION_2_i Register	Section 9.4.5.18
C8Ch	ELM_ERROR_LOCATION_3_4	ELM_ERROR_LOCATION_3_i Register	Section 9.4.5.19
C90h	ELM_ERROR_LOCATION_4_4	ELM_ERROR_LOCATION_4_i Register	Section 9.4.5.20
C94h	ELM_ERROR_LOCATION_5_4	ELM_ERROR_LOCATION_5_i Register	Section 9.4.5.21
C98h	ELM_ERROR_LOCATION_6_4	ELM_ERROR_LOCATION_6_i Register	Section 9.4.5.22
C9Ch	ELM_ERROR_LOCATION_7_4	ELM_ERROR_LOCATION_7_i Register	Section 9.4.5.23
CA0h	ELM_ERROR_LOCATION_8_4	ELM_ERROR_LOCATION_8_i Register	Section 9.4.5.24
CA4h	ELM_ERROR_LOCATION_9_4	ELM_ERROR_LOCATION_9_i Register	Section 9.4.5.25
CA8h	ELM_ERROR_LOCATION_10_4	ELM_ERROR_LOCATION_10_i Register	Section 9.4.5.26
CACh	ELM_ERROR_LOCATION_11_4	ELM_ERROR_LOCATION_11_i Register	Section 9.4.5.27
CB0h	ELM_ERROR_LOCATION_12_4	ELM_ERROR_LOCATION_12_i Register	Section 9.4.5.28
CB4h	ELM_ERROR_LOCATION_13_4	ELM_ERROR_LOCATION_13_i Register	Section 9.4.5.29
CB8h	ELM_ERROR_LOCATION_14_4	ELM_ERROR_LOCATION_14_i Register	Section 9.4.5.30
CBCh	ELM_ERROR_LOCATION_15_4	ELM_ERROR_LOCATION_15_i Register	Section 9.4.5.31
D00h	ELM_LOCATION_STS_5	ELM_LOCATION_STATUS_i Register	Section 9.4.5.15
D80h	ELM_ERROR_LOCATION_0_5	ELM_ERROR_LOCATION_0_i Register	Section 9.4.5.16
D84h	ELM_ERROR_LOCATION_1_5	ELM_ERROR_LOCATION_1_i Register	Section 9.4.5.17
D88h	ELM_ERROR_LOCATION_2_5	ELM_ERROR_LOCATION_2_i Register	Section 9.4.5.18
D8Ch	ELM_ERROR_LOCATION_3_5	ELM_ERROR_LOCATION_3_i Register	Section 9.4.5.19
D90h	ELM_ERROR_LOCATION_4_5	ELM_ERROR_LOCATION_4_i Register	Section 9.4.5.20
D94h	ELM_ERROR_LOCATION_5_5	ELM_ERROR_LOCATION_5_i Register	Section 9.4.5.21
D98h	ELM_ERROR_LOCATION_6_5	ELM_ERROR_LOCATION_6_i Register	Section 9.4.5.22
D9Ch	ELM_ERROR_LOCATION_7_5	ELM_ERROR_LOCATION_7_i Register	Section 9.4.5.23
DA0h	ELM_ERROR_LOCATION_8_5	ELM_ERROR_LOCATION_8_i Register	Section 9.4.5.24

Table 9-345. ELM Registers (continued)

Offset	Acronym	Register Name	Section
DA4h	ELM_ERROR_LOCATION_9_5	ELM_ERROR_LOCATION_9_i Register	Section 9.4.5.25
DA8h	ELM_ERROR_LOCATION_10_5	ELM_ERROR_LOCATION_10_i Register	Section 9.4.5.26
DACh	ELM_ERROR_LOCATION_11_5	ELM_ERROR_LOCATION_11_i Register	Section 9.4.5.27
DB0h	ELM_ERROR_LOCATION_12_5	ELM_ERROR_LOCATION_12_i Register	Section 9.4.5.28
DB4h	ELM_ERROR_LOCATION_13_5	ELM_ERROR_LOCATION_13_i Register	Section 9.4.5.29
DB8h	ELM_ERROR_LOCATION_14_5	ELM_ERROR_LOCATION_14_i Register	Section 9.4.5.30
DBCh	ELM_ERROR_LOCATION_15_5	ELM_ERROR_LOCATION_15_i Register	Section 9.4.5.31
E00h	ELM_LOCATION_STS_6	ELM_LOCATION_STATUS_i Register	Section 9.4.5.15
E80h	ELM_ERROR_LOCATION_0_6	ELM_ERROR_LOCATION_0_i Register	Section 9.4.5.16
E84h	ELM_ERROR_LOCATION_1_6	ELM_ERROR_LOCATION_1_i Register	Section 9.4.5.17
E88h	ELM_ERROR_LOCATION_2_6	ELM_ERROR_LOCATION_2_i Register	Section 9.4.5.18
E8Ch	ELM_ERROR_LOCATION_3_6	ELM_ERROR_LOCATION_3_i Register	Section 9.4.5.19
E90h	ELM_ERROR_LOCATION_4_6	ELM_ERROR_LOCATION_4_i Register	Section 9.4.5.20
E94h	ELM_ERROR_LOCATION_5_6	ELM_ERROR_LOCATION_5_i Register	Section 9.4.5.21
E98h	ELM_ERROR_LOCATION_6_6	ELM_ERROR_LOCATION_6_i Register	Section 9.4.5.22
E9Ch	ELM_ERROR_LOCATION_7_6	ELM_ERROR_LOCATION_7_i Register	Section 9.4.5.23
EA0h	ELM_ERROR_LOCATION_8_6	ELM_ERROR_LOCATION_8_i Register	Section 9.4.5.24
EA4h	ELM_ERROR_LOCATION_9_6	ELM_ERROR_LOCATION_9_i Register	Section 9.4.5.25
EA8h	ELM_ERROR_LOCATION_10_6	ELM_ERROR_LOCATION_10_i Register	Section 9.4.5.26
EACh	ELM_ERROR_LOCATION_11_6	ELM_ERROR_LOCATION_11_i Register	Section 9.4.5.27
EB0h	ELM_ERROR_LOCATION_12_6	ELM_ERROR_LOCATION_12_i Register	Section 9.4.5.28
EB4h	ELM_ERROR_LOCATION_13_6	ELM_ERROR_LOCATION_13_i Register	Section 9.4.5.29
EB8h	ELM_ERROR_LOCATION_14_6	ELM_ERROR_LOCATION_14_i Register	Section 9.4.5.30
EBCh	ELM_ERROR_LOCATION_15_6	ELM_ERROR_LOCATION_15_i Register	Section 9.4.5.31
F00h	ELM_LOCATION_STS_7	ELM_LOCATION_STATUS_i Register	Section 9.4.5.15
F80h	ELM_ERROR_LOCATION_0_7	ELM_ERROR_LOCATION_0_i Register	Section 9.4.5.16
F84h	ELM_ERROR_LOCATION_1_7	ELM_ERROR_LOCATION_1_i Register	Section 9.4.5.17
F88h	ELM_ERROR_LOCATION_2_7	ELM_ERROR_LOCATION_2_i Register	Section 9.4.5.18
F8Ch	ELM_ERROR_LOCATION_3_7	ELM_ERROR_LOCATION_3_i Register	Section 9.4.5.19
F90h	ELM_ERROR_LOCATION_4_7	ELM_ERROR_LOCATION_4_i Register	Section 9.4.5.20
F94h	ELM_ERROR_LOCATION_5_7	ELM_ERROR_LOCATION_5_i Register	Section 9.4.5.21
F98h	ELM_ERROR_LOCATION_6_7	ELM_ERROR_LOCATION_6_i Register	Section 9.4.5.22
F9Ch	ELM_ERROR_LOCATION_7_7	ELM_ERROR_LOCATION_7_i Register	Section 9.4.5.23
FA0h	ELM_ERROR_LOCATION_8_7	ELM_ERROR_LOCATION_8_i Register	Section 9.4.5.24
FA4h	ELM_ERROR_LOCATION_9_7	ELM_ERROR_LOCATION_9_i Register	Section 9.4.5.25
FA8h	ELM_ERROR_LOCATION_10_7	ELM_ERROR_LOCATION_10_i Register	Section 9.4.5.26
FACh	ELM_ERROR_LOCATION_11_7	ELM_ERROR_LOCATION_11_i Register	Section 9.4.5.27
FB0h	ELM_ERROR_LOCATION_12_7	ELM_ERROR_LOCATION_12_i Register	Section 9.4.5.28
FB4h	ELM_ERROR_LOCATION_13_7	ELM_ERROR_LOCATION_13_i Register	Section 9.4.5.29
FB8h	ELM_ERROR_LOCATION_14_7	ELM_ERROR_LOCATION_14_i Register	Section 9.4.5.30
FBCh	ELM_ERROR_LOCATION_15_7	ELM_ERROR_LOCATION_15_i Register	Section 9.4.5.31

9.4.5.1 ELM_REVISION Register (offset = 0h) [reset = 0h]

ELM_REVISION is shown in Figure 9-309 and described in Table 9-346.

This register contains the IP revision code.

Figure 9-309. ELM_REVISION Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">REVISION</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 9-346. ELM_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REVISION	R	0h	IP Revision, value 0 to FFFF FFFFh.

9.4.5.2 ELM_SYSCONFIG Register (offset = 10h) [reset = 11h]

ELM_SYSCONFIG is shown in Figure 9-310 and described in Table 9-347.

This register allows controlling various parameters of the OCP interface.

Figure 9-310. ELM_SYSCONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							CLOCKACTIVITYOCPZ
R-0h							R/W-0h
7	6	5	4	3	2	1	0
RESERVED			SIDLEMODE		RESERVED	SOFTRESET	AUTOGATING
R-0h			R/W-2h		R-0h	R/W-0h	R/W-1h

Table 9-347. ELM_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	CLOCKACTIVITYOCPZ	R/W	0h	OCP Clock activity when module is in IDLE mode (during wake up mode period). 0h (R/W) = OCP Clock can be switch-off 1h (R/W) = OCP Clock is maintained during wake up period
7-5	RESERVED	R	0h	
4-3	SIDLEMODE	R/W	2h	Slave interface power management (IDLE req/ack control). 0h (R/W) = FORCE Idle. IDLE request is acknowledged unconditionally and immediately. (Default Dumb mode for safety) 1h (R/W) = NO idle. IDLE request is never acknowledged. 2h (R/W) = SMART Idle. The acknowledgment to an IDLE request is given based on the internal activity. 3h (R/W) = Reserved - do not use
2	RESERVED	R	0h	
1	SOFTRESET	R/W	0h	Module software reset. This bit is automatically reset by hardware (During reads, it always returns 0.). It has same effect as the OCP hardware reset. 0h (R/W) = Normal mode. 1h (R/W) = Start soft reset sequence.
0	AUTOGATING	R/W	1h	Internal OCP clock gating strategy. (No module visible impact other than saving power.) 0h (R/W) = OCP clock is free-running. 1h (R/W) = Automatic internal OCP clock gating strategy is applied based on the OCP interface activity.

9.4.5.3 ELM_SYSSTS Register (offset = 14h) [reset = 0h]

ELM_SYSSTS is shown in Figure 9-311 and described in Table 9-348.

Figure 9-311. ELM_SYSSTS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="7">RESERVED</td><td>RESETDONE</td></tr>
<tr><td colspan="7">R-0h</td><td>R-0h</td></tr>
</table>

Table 9-348. ELM_SYSSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	RESETDONE	R	0h	Internal reset monitoring (OCP domain). Undefined since: From hardware perspective, the reset state is 0. From software user perspective, when the accessible module is 1. 0h (R/W) = Reset is on-going 1h (R/W) = Reset is done (completed)

9.4.5.4 ELM_IRQSTS Register (offset = 18h) [reset = 0h]

ELM_IRQSTS is shown in Figure 9-312 and described in Table 9-349.

This register doubles as a status register for the error-location processes.

Figure 9-312. ELM_IRQSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							PAGE_VALID
R-0h							R/W-0h
7	6	5	4	3	2	1	0
LOC_VALID_7	LOC_VALID_6	LOC_VALID_5	LOC_VALID_4	LOC_VALID_3	LOC_VALID_2	LOC_VALID_1	LOC_VALID_0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-349. ELM_IRQSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	PAGE_VALID	R/W	0h	Error-location status for a full page, based on the mask definition. 0h (R/W) = Write: No effect. 1h (R/W) = Read: All error locations valid.
7	LOC_VALID_7	R/W	0h	Error-location status for syndrome polynomial 7. 0h (R/W) = Write: No effect. 1h (R/W) = Read: Error-location process completed.
6	LOC_VALID_6	R/W	0h	Error-location status for syndrome polynomial 6. 0h (R/W) = Write: No effect. 1h (R/W) = Read: Error-location process completed.
5	LOC_VALID_5	R/W	0h	Error-location status for syndrome polynomial 5. 0h (R/W) = Write: No effect. 1h (R/W) = Read: Error-location process completed.
4	LOC_VALID_4	R/W	0h	Error-location status for syndrome polynomial 4. 0h (R/W) = Write: No effect. 1h (R/W) = Read: Error-location process completed.
3	LOC_VALID_3	R/W	0h	Error-location status for syndrome polynomial 3. 0h (R/W) = Write: No effect. 1h (R/W) = Read: Error-location process completed.
2	LOC_VALID_2	R/W	0h	Error-location status for syndrome polynomial 2. 0h (R/W) = Write: No effect. 1h (R/W) = Read: Error-location process completed.
1	LOC_VALID_1	R/W	0h	Error-location status for syndrome polynomial 1. 0h (R/W) = Write: No effect. 1h (R/W) = Read: Error-location process completed.
0	LOC_VALID_0	R/W	0h	Error-location status for syndrome polynomial 0. 0h (R/W) = No syndrome processed or process in progress. 1h (R/W) = Clear interrupt.

9.4.5.5 ELM_IRQEN Register (offset = 1Ch) [reset = 0h]

ELM_IRQEN is shown in Figure 9-313 and described in Table 9-350.

Figure 9-313. ELM_IRQEN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							PAGE_MASK
R-0h							R/W-0h
7	6	5	4	3	2	1	0
LOCATION_MASK_7	LOCATION_MASK_6	LOCATION_MASK_5	LOCATION_MASK_4	LOCATION_MASK_3	LOCATION_MASK_2	LOCATION_MASK_1	LOCATION_MASK_0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-350. ELM_IRQEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	PAGE_MASK	R/W	0h	Page interrupt mask bit 0h (R/W) = Disable interrupt. 1h (R/W) = Enable interrupt.
7	LOCATION_MASK_7	R/W	0h	Error-location interrupt mask bit for syndrome polynomial 7. 0h (R/W) = Disable interrupt. 1h (R/W) = Enable interrupt.
6	LOCATION_MASK_6	R/W	0h	Error-location interrupt mask bit for syndrome polynomial 6. 0h (R/W) = Disable interrupt. 1h (R/W) = Enable interrupt.
5	LOCATION_MASK_5	R/W	0h	Error-location interrupt mask bit for syndrome polynomial 5. 0h (R/W) = Disable interrupt. 1h (R/W) = Enable interrupt.
4	LOCATION_MASK_4	R/W	0h	Error-location interrupt mask bit for syndrome polynomial 4. 0h (R/W) = Disable interrupt. 1h (R/W) = Enable interrupt.
3	LOCATION_MASK_3	R/W	0h	Error-location interrupt mask bit for syndrome polynomial 3. 0h (R/W) = Disable interrupt. 1h (R/W) = Enable interrupt.
2	LOCATION_MASK_2	R/W	0h	Error-location interrupt mask bit for syndrome polynomial 2. 0h (R/W) = Disable interrupt. 1h (R/W) = Enable interrupt.
1	LOCATION_MASK_1	R/W	0h	Error-location interrupt mask bit for syndrome polynomial 1. 0h (R/W) = Disable interrupt. 1h (R/W) = Enable interrupt.
0	LOCATION_MASK_0	R/W	0h	Error-location interrupt mask bit for syndrome polynomial 0. 0h (R/W) = Disable interrupt. 1h (R/W) = Enable interrupt.

9.4.5.6 ELM_LOCATION_CONFIG Register (offset = 20h) [reset = 0h]

ELM_LOCATION_CONFIG is shown in Figure 9-314 and described in Table 9-351.

Figure 9-314. ELM_LOCATION_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED					ECC_SIZE		
R-0h					R/W-0h		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ECC_SIZE							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						ECC_BCH_LEVEL	
R-0h						R/W-0h	

Table 9-351. ELM_LOCATION_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	
26-16	ECC_SIZE	R/W	0h	Maximum size of the buffers for which the error-location engine is used, in number of nibbles (4 bits entities), value 0 to 7FFh.
15-2	RESERVED	R	0h	
1-0	ECC_BCH_LEVEL	R/W	0h	Error correction level. 0h (R/W) = 4 bits. 1h (R/W) = 8 bits. 2h (R/W) = 16 bits. 3h (R/W) = Reserved.

9.4.5.7 ELM_PAGE_CTRL Register (offset = 80h) [reset = 0h]

ELM_PAGE_CTRL is shown in Figure 9-315 and described in Table 9-352.

Figure 9-315. ELM_PAGE_CTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>SECTOR_7</td><td>SECTOR_6</td><td>SECTOR_5</td><td>SECTOR_4</td><td>SECTOR_3</td><td>SECTOR_2</td><td>SECTOR_1</td><td>SECTOR_0</td></tr>
<tr><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 9-352. ELM_PAGE_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	SECTOR_7	R/W	0h	Set to 1 if syndrome polynomial 7 is part of the page in page mode. Must be 0 in continuous mode.
6	SECTOR_6	R/W	0h	Set to 1 if syndrome polynomial 6 is part of the page in page mode. Must be 0 in continuous mode.
5	SECTOR_5	R/W	0h	Set to 1 if syndrome polynomial 5 is part of the page in page mode. Must be 0 in continuous mode.
4	SECTOR_4	R/W	0h	Set to 1 if syndrome polynomial 4 is part of the page in page mode. Must be 0 in continuous mode.
3	SECTOR_3	R/W	0h	Set to 1 if syndrome polynomial 3 is part of the page in page mode. Must be 0 in continuous mode.
2	SECTOR_2	R/W	0h	Set to 1 if syndrome polynomial 2 is part of the page in page mode. Must be 0 in continuous mode.
1	SECTOR_1	R/W	0h	Set to 1 if syndrome polynomial 1 is part of the page in page mode. Must be 0 in continuous mode.
0	SECTOR_0	R/W	0h	Set to 1 if syndrome polynomial 0 is part of the page in page mode. Must be 0 in continuous mode.

9.4.5.8 ELM_SYNDROME_FRAGMENT_0_0 Register (offset = 400h + [i * 40h]) [reset = 0h]

ELM_SYNDROME_FRAGMENT_0_0 is shown in Figure 9-316 and described in Table 9-353.

Figure 9-316. ELM_SYNDROME_FRAGMENT_0_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNDROME_0																															
R/W-0h																															

Table 9-353. ELM_SYNDROME_FRAGMENT_0_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SYNDROME_0	R/W	0h	Syndrome bits 0 to 31, value 0 to FFFF FFFFh.

9.4.5.9 ELM_SYNDROME_FRAGMENT_1_0 Register (offset = 404h + [i * 40h]) [reset = 0h]

ELM_SYNDROME_FRAGMENT_1_0 is shown in Figure 9-317 and described in Table 9-354.

Figure 9-317. ELM_SYNDROME_FRAGMENT_1_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNDROME_1																															
R/W-0h																															

Table 9-354. ELM_SYNDROME_FRAGMENT_1_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SYNDROME_1	R/W	0h	Syndrome bits 32 to 63, value 0 to FFFF FFFFh.

9.4.5.10 ELM_SYNDROME_FRAGMENT_2_0 Register (offset = 408h + [i * 40h]) [reset = 0h]

ELM_SYNDROME_FRAGMENT_2_0 is shown in Figure 9-318 and described in Table 9-355.

Figure 9-318. ELM_SYNDROME_FRAGMENT_2_0 Register

31-0
SYNDROME_2
R/W-0h

Table 9-355. ELM_SYNDROME_FRAGMENT_2_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SYNDROME_2	R/W	0h	Syndrome bits 64 to 95, value 0 to FFFF FFFFh.

9.4.5.11 ELM_SYNDROME_FRAGMENT_3_0 Register (offset = 40Ch + [i * 40h]) [reset = 0h]

ELM_SYNDROME_FRAGMENT_3_0 is shown in Figure 9-319 and described in Table 9-356.

Figure 9-319. ELM_SYNDROME_FRAGMENT_3_0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">SYNDROME_3</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 9-356. ELM_SYNDROME_FRAGMENT_3_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SYNDROME_3	R/W	0h	Syndrome bits 96 to 127, value 0 to FFFF FFFFh.

9.4.5.12 ELM_SYNDROME_FRAGMENT_4_0 Register (offset = 410h + [i * 40h]) [reset = 0h]

ELM_SYNDROME_FRAGMENT_4_0 is shown in Figure 9-320 and described in Table 9-357.

Figure 9-320. ELM_SYNDROME_FRAGMENT_4_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNDROME_4																															
R/W-0h																															

Table 9-357. ELM_SYNDROME_FRAGMENT_4_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SYNDROME_4	R/W	0h	Syndrome bits 128 to 159, value 0 to FFFF FFFFh.

9.4.5.13 ELM_SYNDROME_FRAGMENT_5_0 Register (offset = 414h + [i * 40h]) [reset = 0h]

ELM_SYNDROME_FRAGMENT_5_0 is shown in Figure 9-321 and described in Table 9-358.

Figure 9-321. ELM_SYNDROME_FRAGMENT_5_0 Register

31–0
SYNDROME_5
R/W-0h

Table 9-358. ELM_SYNDROME_FRAGMENT_5_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SYNDROME_5	R/W	0h	Syndrome bits 160 to 191, value 0 to FFFF FFFFh.

9.4.5.14 ELM_SYNDROME_FRAGMENT_6_0 Register (offset = 418h + [i * 40h]) [reset = 0h]

ELM_SYNDROME_FRAGMENT_6_0 is shown in Figure 9-322 and described in Table 9-359.

Figure 9-322. ELM_SYNDROME_FRAGMENT_6_0 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							SYNDROME_VALID
R-0h							R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SYNDROME_6							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SYNDROME_6							
R/W-0h							

Table 9-359. ELM_SYNDROME_FRAGMENT_6_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	SYNDROME_VALID	R/W	0h	Syndrome valid bit. 0h (R/W) = This syndrome polynomial should not be processed. 1h (R/W) = This syndrome polynomial must be processed.
15-0	SYNDROME_6	R/W	0h	Syndrome bits 192 to 207, value 0 to FFFFh.

9.4.5.15 ELM_LOCATION_STS_0 Register (offset = 800h + [i * 100h]) [reset = 0h]

ELM_LOCATION_STS_0 is shown in Figure 9-323 and described in Table 9-360.

Figure 9-323. ELM_LOCATION_STS_0 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							ECC_CORRECTABL
R-0h							R-0h
7	6	5	4	3	2	1	0
RESERVED			ECC_NB_ERRORS				
R-0h			R-0h				

Table 9-360. ELM_LOCATION_STS_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	ECC_CORRECTABL	R	0h	Error-location process exit status. 0h (R/W) = ECC error-location process failed. Number of errors and error locations are invalid. 1h (R/W) = All errors were successfully located. Number of errors and error locations are valid.
7-5	RESERVED	R	0h	
4-0	ECC_NB_ERRORS	R	0h	Number of errors detected and located, value 0 to 1Fh.

9.4.5.16 ELM_ERROR_LOCATION_0_0 Register (offset = 880h + [i * 100h]) [reset = 0h]

ELM_ERROR_LOCATION_0_0 is shown in Figure 9-324 and described in Table 9-361.

Figure 9-324. ELM_ERROR_LOCATION_0_0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">RESERVED</td><td colspan="13">ECC_ERROR_LOCATION</td></tr>
<tr><td colspan="3">R-0h</td><td colspan="13">R-0h</td></tr>
</table>

Table 9-361. ELM_ERROR_LOCATION_0_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12-0	ECC_ERROR_LOCATION	R	0h	Error-location bit address, 0 to 1FFFh.

9.4.5.17 ELM_ERROR_LOCATION_1_0 Register (offset = 884h + [i * 100h]) [reset = 0h]

ELM_ERROR_LOCATION_1_0 is shown in Figure 9-325 and described in Table 9-362.

Figure 9-325. ELM_ERROR_LOCATION_1_0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">RESERVED</td><td colspan="13">ECC_ERROR_LOCATION</td></tr>
<tr><td colspan="3">R-0h</td><td colspan="13">R-0h</td></tr>
</table>

Table 9-362. ELM_ERROR_LOCATION_1_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12-0	ECC_ERROR_LOCATION	R	0h	Error-location bit address, 0 to 1FFFh.

9.4.5.18 ELM_ERROR_LOCATION_2_0 Register (offset = 888h + [i * 100h]) [reset = 0h]

ELM_ERROR_LOCATION_2_0 is shown in Figure 9-326 and described in Table 9-363.

Figure 9-326. ELM_ERROR_LOCATION_2_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED			ECC_ERROR_LOCATION												
R-0h			R-0h												

Table 9-363. ELM_ERROR_LOCATION_2_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12-0	ECC_ERROR_LOCATION	R	0h	Error-location bit address, 0 to 1FFFh.

9.4.5.19 ELM_ERROR_LOCATION_3_0 Register (offset = 88Ch + [i * 100h]) [reset = 0h]

ELM_ERROR_LOCATION_3_0 is shown in Figure 9-327 and described in Table 9-364.

Figure 9-327. ELM_ERROR_LOCATION_3_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			ECC_ERROR_LOCATION												
R-0h			R-0h												

Table 9-364. ELM_ERROR_LOCATION_3_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12-0	ECC_ERROR_LOCATION	R	0h	Error-location bit address, 0 to 1FFFh.

9.4.5.20 ELM_ERROR_LOCATION_4_0 Register (offset = 890h + [i * 100h]) [reset = 0h]

ELM_ERROR_LOCATION_4_0 is shown in Figure 9-328 and described in Table 9-365.

Figure 9-328. ELM_ERROR_LOCATION_4_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			ECC_ERROR_LOCATION												
R-0h			R-0h												

Table 9-365. ELM_ERROR_LOCATION_4_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12-0	ECC_ERROR_LOCATION	R	0h	Error-location bit address, 0 to 1FFFh.

9.4.5.21 ELM_ERROR_LOCATION_5_0 Register (offset = 894h + [i * 100h]) [reset = 0h]

ELM_ERROR_LOCATION_5_0 is shown in Figure 9-329 and described in Table 9-366.

Figure 9-329. ELM_ERROR_LOCATION_5_0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">RESERVED</td><td colspan="13">ECC_ERROR_LOCATION</td></tr>
<tr><td colspan="3">R-0h</td><td colspan="13">R-0h</td></tr>
</table>

Table 9-366. ELM_ERROR_LOCATION_5_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12-0	ECC_ERROR_LOCATION	R	0h	Error-location bit address, 0 to 1FFFh.

9.4.5.22 ELM_ERROR_LOCATION_6_0 Register (offset = 898h + [i * 100h]) [reset = 0h]

ELM_ERROR_LOCATION_6_0 is shown in Figure 9-330 and described in Table 9-367.

Figure 9-330. ELM_ERROR_LOCATION_6_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			ECC_ERROR_LOCATION												
R-0h			R-0h												

Table 9-367. ELM_ERROR_LOCATION_6_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12-0	ECC_ERROR_LOCATION	R	0h	Error-location bit address, 0 to 1FFFh.

9.4.5.23 ELM_ERROR_LOCATION_7_0 Register (offset = 89Ch + [i * 100h]) [reset = 0h]

ELM_ERROR_LOCATION_7_0 is shown in Figure 9-331 and described in Table 9-368.

Figure 9-331. ELM_ERROR_LOCATION_7_0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">RESERVED</td><td colspan="13">ECC_ERROR_LOCATION</td></tr>
<tr><td colspan="3">R-0h</td><td colspan="13">R-0h</td></tr>
</table>

Table 9-368. ELM_ERROR_LOCATION_7_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12-0	ECC_ERROR_LOCATION	R	0h	Error-location bit address, 0 to 1FFFh.

9.4.5.24 ELM_ERROR_LOCATION_8_0 Register (offset = 8A0h + [i * 100h]) [reset = 0h]

ELM_ERROR_LOCATION_8_0 is shown in Figure 9-332 and described in Table 9-369.

Figure 9-332. ELM_ERROR_LOCATION_8_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			ECC_ERROR_LOCATION												
R-0h			R-0h												

Table 9-369. ELM_ERROR_LOCATION_8_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12-0	ECC_ERROR_LOCATION	R	0h	Error-location bit address, 0 to 1FFFh.

9.4.5.25 ELM_ERROR_LOCATION_9_0 Register (offset = 8A4h + [i * 100h]) [reset = 0h]

ELM_ERROR_LOCATION_9_0 is shown in Figure 9-333 and described in Table 9-370.

Figure 9-333. ELM_ERROR_LOCATION_9_0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">RESERVED</td><td colspan="13">ECC_ERROR_LOCATION</td></tr>
<tr><td colspan="3">R-0h</td><td colspan="13">R-0h</td></tr>
</table>

Table 9-370. ELM_ERROR_LOCATION_9_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12-0	ECC_ERROR_LOCATION	R	0h	Error-location bit address, 0 to 1FFFh.

9.4.5.26 ELM_ERROR_LOCATION_10_0 Register (offset = 8A8h + [i * 100h]) [reset = 0h]

ELM_ERROR_LOCATION_10_0 is shown in Figure 9-334 and described in Table 9-371.

Figure 9-334. ELM_ERROR_LOCATION_10_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED			ECC_ERROR_LOCATION												
R-0h			R-0h												

Table 9-371. ELM_ERROR_LOCATION_10_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12-0	ECC_ERROR_LOCATION	R	0h	Error-location bit address, 0 to 1FFFh.

9.4.5.27 ELM_ERROR_LOCATION_11_0 Register (offset = 8ACh + [i * 100h]) [reset = 0h]

ELM_ERROR_LOCATION_11_0 is shown in Figure 9-335 and described in Table 9-372.

Figure 9-335. ELM_ERROR_LOCATION_11_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			ECC_ERROR_LOCATION												
R-0h			R-0h												

Table 9-372. ELM_ERROR_LOCATION_11_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12-0	ECC_ERROR_LOCATION	R	0h	Error-location bit address, 0 to 1FFFh.

9.4.5.28 ELM_ERROR_LOCATION_12_0 Register (offset = 8B0h + [i * 100h]) [reset = 0h]

ELM_ERROR_LOCATION_12_0 is shown in Figure 9-336 and described in Table 9-373.

Figure 9-336. ELM_ERROR_LOCATION_12_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			ECC_ERROR_LOCATION												
R-0h			R-0h												

Table 9-373. ELM_ERROR_LOCATION_12_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12-0	ECC_ERROR_LOCATION	R	0h	Error-location bit address, 0 to 1FFFh.

9.4.5.29 ELM_ERROR_LOCATION_13_0 Register (offset = 8B4h + [i * 100h]) [reset = 0h]

ELM_ERROR_LOCATION_13_0 is shown in Figure 9-337 and described in Table 9-374.

Figure 9-337. ELM_ERROR_LOCATION_13_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			ECC_ERROR_LOCATION												
R-0h			R-0h												

Table 9-374. ELM_ERROR_LOCATION_13_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12-0	ECC_ERROR_LOCATION	R	0h	Error-location bit address, 0 to 1FFFh.

9.4.5.30 ELM_ERROR_LOCATION_14_0 Register (offset = 8B8h + [i * 100h]) [reset = 0h]

ELM_ERROR_LOCATION_14_0 is shown in Figure 9-338 and described in Table 9-375.

Figure 9-338. ELM_ERROR_LOCATION_14_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			ECC_ERROR_LOCATION												
R-0h			R-0h												

Table 9-375. ELM_ERROR_LOCATION_14_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12-0	ECC_ERROR_LOCATION	R	0h	Error-location bit address, 0 to 1FFFh.

9.4.5.31 ELM_ERROR_LOCATION_15_0 Register (offset = 8BCh + [i * 100h]) [reset = 0h]

ELM_ERROR_LOCATION_15_0 is shown in Figure 9-339 and described in Table 9-376.

Figure 9-339. ELM_ERROR_LOCATION_15_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED			ECC_ERROR_LOCATION												
R-0h			R-0h												

Table 9-376. ELM_ERROR_LOCATION_15_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12-0	ECC_ERROR_LOCATION	R	0h	Error-location bit address, 0 to 1FFFh.

Enhanced Direct Memory Access (EDMA)

This chapter describes the EDMA of the device.

Topic **Page**

10.1 Introduction

The enhanced direct memory access (EDMA3) controller's primary purpose is to service user-programmed data transfers between two memory-mapped slave endpoints on the device.

Typical usage includes, but is not limited to the following:

- Servicing software-driven paging transfers (e.g., transfers from external memory to internal device memory).
- Servicing event-driven peripherals, such as a serial port.
- Performing sorting or sub-frame extraction of various data structures.
- Offloading data transfers from the main device CPU(s).

The EDMA3 controller consists of two principal blocks:

- EDMA3 channel controller (EDMA3CC).
- EDMA3 transfer controller(s) (EDMA3TC).

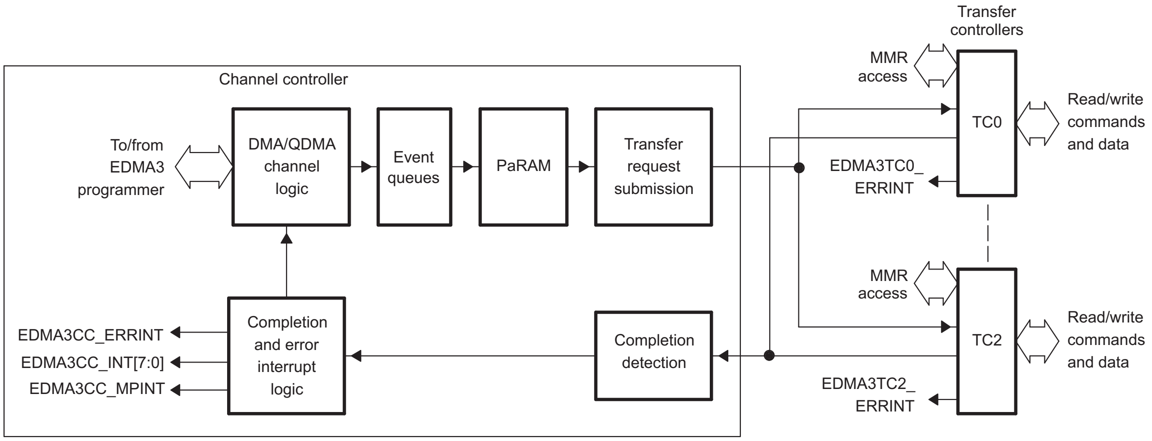
The EDMA3 channel controller serves as the user interface for the EDMA3 controller. The EDMA3CC includes parameter RAM (PaRAM), channel control registers, and interrupt control registers. The EDMA3CC serves to prioritize incoming software requests or events from peripherals and submits transfer requests (TRs) to the transfer controller.

The EDMA3 transfer controllers are slaves to the EDMA3 channel controller that is responsible for data movement. The transfer controller issues read/write commands to the source and destination addresses that are programmed for a given transfer. The operation is transparent to user.

10.1.1 EDMA3 Controller Block Diagram

Figure 10-1 shows a block diagram for the EDMA3 controller.

Figure 10-1. EDMA3 Controller Block Diagram



10.1.2 *Third-Party Channel Controller (TPCC) Overview*

10.1.2.1 TPCC Features

The general features of the TPCC module are:

- Up to 64 DMA Channels
  - Channels triggered by:
    - Event Synchronization
    - Manual Synchronization (CPU write to ‘Event Set Register’)
    - Chain Synchronization (completion of one transfer chains to the next)
  - Parameterizable support for programmable DMA Channel to PaRAM mapping
- Up to 8 QDMA Channels
  - QDMA Channels are triggered automatically upon writing to PaRAM
  - Support for programmable QDMA Channel to PaRAM mapping
- Up to 64 Event Inputs
- Up to 8 Interrupt outputs for multi-core support
- Up to 256 PaRAM entries
  - Each PaRAM entry can be used as DMA Entry (up to 64), QDMA Entry (up to 8), or Link Entry (remaining)
- 8 Priority Levels for mapping CC/TC priority relative to priority of other masters in the system.
- Up to 3 Event Queues
- 16 Event Entries per Event Queue
- Supports three-transfer dimensions
  - A-synchronized transfers—one dimension serviced per event
  - AB-synchronized transfers—two dimensions serviced per event
  - Independent Indexes on Source and Destination
  - Does not support direct submission of 3D transfer to TC
  - Chaining feature allows 3D transfer based on single event
- Increment and FIFO transfer addressing modes (TC feature)
- Linking mechanism allows automatic PaRAM Entry update
- Transfer Completion Signaling between TC and CC for Chaining and Interrupt generation.
- Programmable assignment of Priority to TC channel.
- Proxied Memory Protection for TR submission
- Parameterizable support for Active Memory Protection for accesses to PaRAM and registers.
- Queue Watermarking
- Missed Event Detection
- Error and status recording to facilitate debug
- Single Clock domain for all interfaces
- Parameterizable number of Write Completion interfaces (up to 8) (set to number of TC Channels)
- AET Event generation

10.1.2.2 Unsupported TPCC Features

- This device does not support AET event generation because the output is not connected
- This device does not use the global completion interrupt. Only regional completion interrupts are supported
- The channel controller only supports 4 memory protection regions 0-3
- The channel controller only supports 4 shadow regions 0-3

- Only 2 region completion interrupts are connected at the system level. See Table 10-1 for more information
- Only 256 PaRAM Entries are supported
- Only 3 Event Queues are supported

10.1.3 Third-Party Transfer Controller (TPTC) Overview

10.1.3.1 TPTC Features

The TPTC module includes the following features:

- Up to eight independent channels
- External event control use model (TPCC)
- Read and Write Master ports per Channel 64- or 128-bit configuration.
- Parameterizable FIFO size
- Up to four in-flight Transfer Requests
- Proxied Memory protection for data transfers
- Programmable Priority levels (up to 8)
- Background programmation capability
- Supports 2-dimensional transfers with independent indexes on Source and Destination.
- Support for increment or FIFO-mode transfers
- Interrupt and error support
- Single clock domain for all interfaces

10.1.3.2 Unsupported TPTC Features

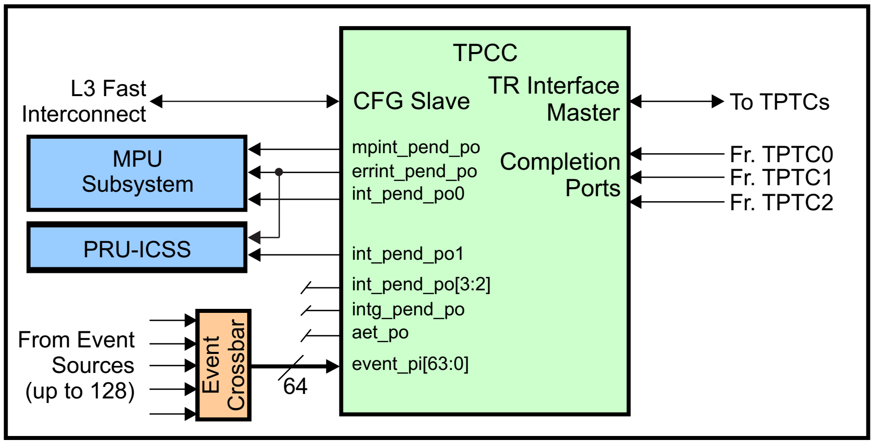
- TPTCx supports 512 byte FIFO size

10.2 Integration

10.2.1 Third-Party Channel Controller (TPCC) Integration

This device uses the TPCC peripheral to provide control over its third-party transfer channels (TPTCs). Figure 10-2 shows the integration of the TPCC module.

Figure 10-2. TPCC Integration



10.2.1.1 TPCC Connectivity Attributes

The general connectivity attributes of the TPCC are summarized in Table 10-1.

Table 10-1. TPCC Connectivity Attributes

Attributes	Type
Power domain	Peripheral Domain
Clock domain	PD_PER_L3_GCLK
Reset signals	PER_DOM_RST_N
Idle/Wakeup signals	Smart Idle
Interrupt request	4 Regional Completion Interrupts: int_pend_po0 (EDMACOMPINT) – to MPU Subsystem int_pend_po1 (tpcc_int_pend_po1) – to PRU-ICSS Int_pend_po[3:2] - unused Error Interrupt: errint_po (EDMAERRINT) – to MPU Subsystem, PRU-ICSS Memory Protection Error Interrupt: mpint_p0 (EDMAMPERR) – to MPU Subsystem
DMA request	none
Physical address	L3 Fast slave port

10.2.1.2 TPCC Clock and Reset Management

The TPCC operates from a single clock and runs at the L3_Fast clock rate.

Table 10-2. TPCC Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
tpcc_clk_pi Interface / Functional clock	200 MHz	CORE_CLKOUTM4	pd_per_l3_gclk From PRCM

10.2.1.3 TPCC Pin List

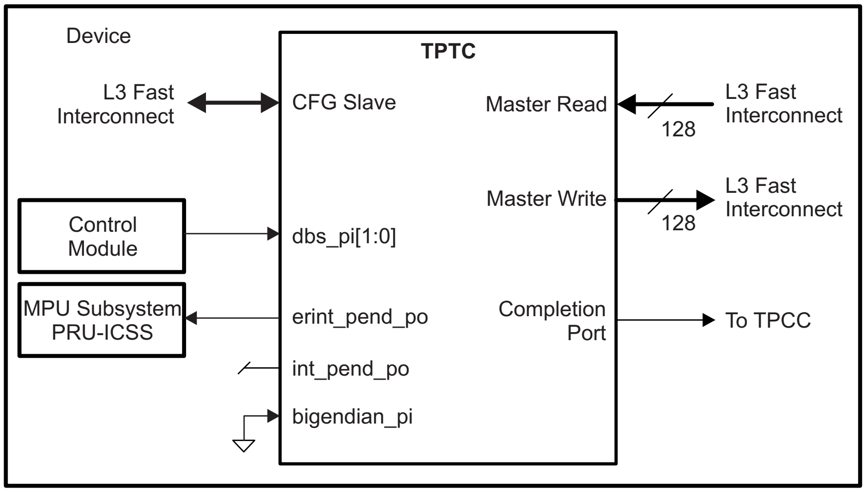
The TPCC module does not include any external interface pins.

10.2.2 Third-Party Transfer Controller (TPTC) Integration

This device uses three TPTC peripherals, TC0–TC2, to perform EDMA transfers between slave peripherals. TC3 is not supported.

The submission of transfer requests to the TPTCs is controlled by the TPCC. Figure 10-3 shows the integration of the TPTC modules

Figure 10-3. TPTC Integration



10.2.2.1 TPTC Connectivity Attributes

The general connectivity attributes for the TPTCs are shown in Table 10-3.

Table 10-3. TPTC Connectivity Attributes

Attributes	Type
Power domain	Peripheral Domain
Clock domain	PD_PER_L3_GCLK
Reset signals	PER_DOM_RST_N
Idle/Wakeup signals	Standby (2 ports) Smart Idle
Interrupt request	Error interrupt per instance erint_pend_po (TCERRINTx) – to MPU Subsystem and PRU-ICSS (tptc_erint_pend_po, TPTC0 only)
DMA request	none
Physical address	L3 Fast slave port

10.2.2.2 TPTC Clock and Reset Management

The TPTC operates from a single clock and runs at the L3_Fast clock rate.

Table 10-4. TPTC Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
tptc_clk_pi Interface / Functional clock	200 MHz	CORE_CLKOUTM4	pd_per_l3_gclk From PRCM

10.2.2.3 TPTC Pin List

The TPTC module does not include any external interface pins.

10.3 Functional Description

This chapter discusses the architecture of the EDMA3 controller.

10.3.1 Functional Overview

10.3.1.1 EDMA3 Channel Controller (EDMA3CC)

Figure 10-4 shows a functional block diagram of the EDMA3 channel controller (EDMA3CC).

The main blocks of the EDMA3CC are as follows:

- Parameter RAM (PaRAM): The PaRAM maintains parameter sets for channel and reload parameter sets. You must write the PaRAM with the transfer context for the desired channels and link parameter sets. EDMA3CC processes sets based on a trigger event and submits a transfer request (TR) to the transfer controller.
- EDMA3 event and interrupt processing registers: Allows mapping of events to parameter sets, enable/disable events, enable/disable interrupt conditions, and clearing interrupts.
- Completion detection: The completion detect block detects completion of transfers by the EDMA3TC and/or slave peripherals. You can optionally use completion of transfers to chain trigger new transfers or to assert interrupts.
- Event queues: Event queues form the interface between the event detection logic and the transfer request submission logic.
- Memory protection registers: Memory protection registers define the accesses (privilege level and requestor(s)) that are allowed to access the DMA channel shadow region view(s) and regions of PaRAM.

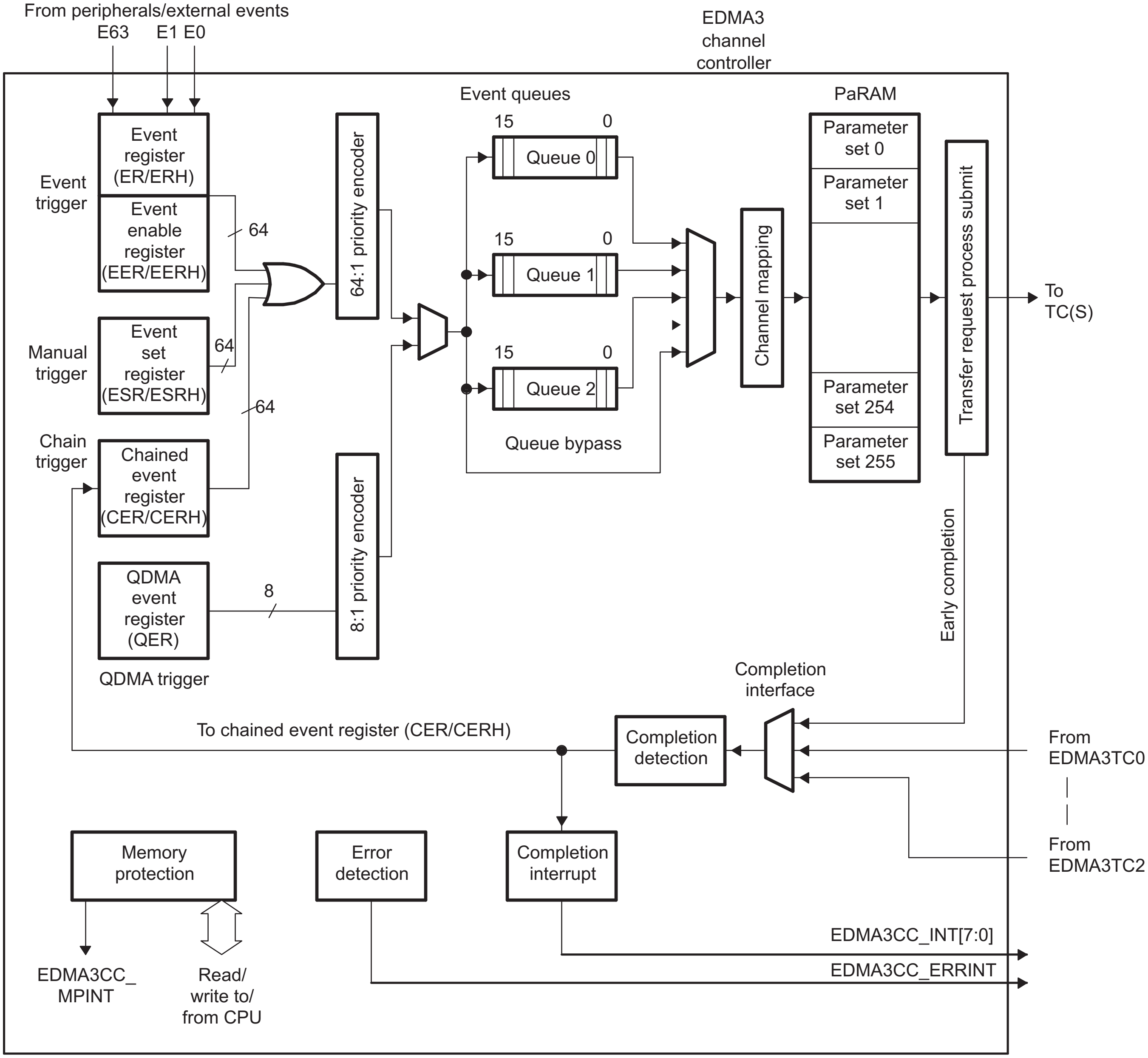
Other functions include the following:

- Region registers: Region registers allow DMA resources (DMA channels and interrupts) to be assigned to unique regions that different EDMA3 programmers own (for example, ARM).
- Debug registers: Debug registers allow debug visibility by providing registers to read the queue status, controller status, and missed event status.

The EDMA3CC includes two channel types: DMA channels (64 channels) and QDMA channels (8 channels).

Each channel is associated with a given event queue/transfer controller and with a given PaRAM set. The main thing that differentiates a DMA channel from a QDMA channel is the method that the system uses to trigger transfers. See Section 10.3.4.

Figure 10-4. EDMA3 Channel Controller (EDMA3CC) Block Diagram



A trigger event is needed to initiate a transfer. A trigger event may be due to an external event, manual write to the event set register, or chained event for DMA channels. QDMA channels auto-trigger when a write to the trigger word that you program occurs on the associated PaRAM set. All such trigger events are logged into appropriate registers upon recognition.

Once a trigger event is recognized, the appropriate event gets queued in the EDMA3CC event queue. The assignment of each DMA/QDMA channel to an event queue is programmable. Each queue is 16 events deep; therefore, you can queue up to 16 events (on a single queue) in the EDMA3CC at a time. Additional pending events that are mapped to a full queue are queued when the event queue space becomes available. See Section 10.3.11.

If events on different channels are detected simultaneously, the events are queued based on a fixed priority arbitration scheme with the DMA channels being higher priority events than the QDMA channels. Among the two groups of channels, the lowest-numbered channel is the highest priority.

Each event in the event queue is processed in FIFO order. When the head of the queue is reached, the PaRAM associated with that channel is read to determine the transfer details. The TR submission logic evaluates the validity of the TR and is responsible for submitting a valid transfer request (TR) to the appropriate EDMA3TC (based on the event queue to the EDMA3TC association, Q0 goes to TC0 , Q1 goes to TC1, Q2 goes to TC2, and Q3 goes to TC3). For more information, refer to Section 10.3.3.

The EDMA3TC receives the request and is responsible for data movement, as specified in the transfer request packet (TRP), other necessary tasks like buffering, and ensuring transfers are carried out in an optimal fashion wherever possible. For more information on EDMA3TC, refer to Section 10.3.1.2.

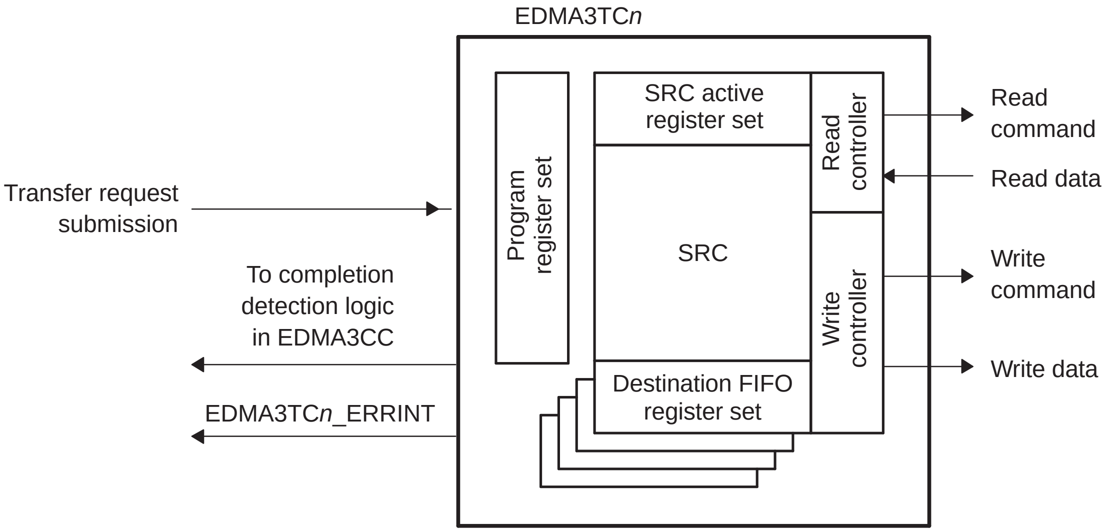
If you have decided to receive an interrupt or to chain to another channel on completion of the current transfer, the EDMA3TC signals completion to the EDMA3CC completion detection logic when the transfer is complete. You can alternately choose to trigger completion when a TR leaves the EDMA3CC boundary, rather than wait for all of the data transfers to complete. Based on the setting of the EDMA3CC interrupt registers, the completion interrupt generation logic is responsible for generating EDMA3CC completion interrupts to the CPU. For more information, refer to Section 10.3.5.

Additionally, the EDMA3CC also has an error detection logic that causes an error interrupt generation on various error conditions (like missed events, exceeding event queue thresholds, etc.). For more information on error interrupts, refer to Section 10.3.9.4.

10.3.1.2 EDMA3 Transfer Controller (EDMA3TC)

Section 10.3.9.4 shows a functional block diagram of the EDMA3 transfer controller (EDMA3TC).

Figure 10-5. EDMA3 Transfer Controller (EDMA3TC) Block Diagram



The main blocks of the EDMA3TC are:

- DMA program register set: The DMA program register set stores the transfer requests received from the EDMA3 channel controller (EDMA3CC).
- DMA source active register set: The DMA source active register set stores the context for the DMA transfer request currently in progress in the read controller.
- Read controller: The read controller issues read commands to the source address.
- Destination FIFO register set: The destination (DST) FIFO register set stores the context for the DMA transfer request(s) currently in progress in the write controller.
- Write controller: The write controller issues write commands/write data to the destination slave.
- Data FIFO: The data FIFO exists for holding temporary in-flight data.
- Completion interface: The completion interface sends completion codes to the EDMA3CC when a transfer completes, and generates interrupts and chained events (also, see Section 10.3.1.1 for more information on transfer completion reporting).

When the EDMA3TC is idle and receives its first TR, DMA program register set receives the TR, where it transitions to the DMA source active set and the destination FIFO register set immediately. The second TR (if pending from EDMA3CC) is loaded into the DMA program set, ensuring it can start as soon as possible when the active transfer completes. As soon as the current active set is exhausted, the TR is loaded from the DMA program register set into the DMA source active register set as well as to the appropriate entry in the destination FIFO register set.

The read controller issues read commands governed by the rules of command fragmentation and optimization. These are issued only when the data FIFO has space available for the data read. When sufficient data is in the data FIFO, the write controller starts issuing a write command again following the rules for command fragmentation and optimization. For more information on command fragmentation and optimization, refer to Section 10.3.12.1.1.

Depending on the number of entries, the read controller can process up to two or four transfer requests ahead of the destination subject to the amount of free data FIFO.

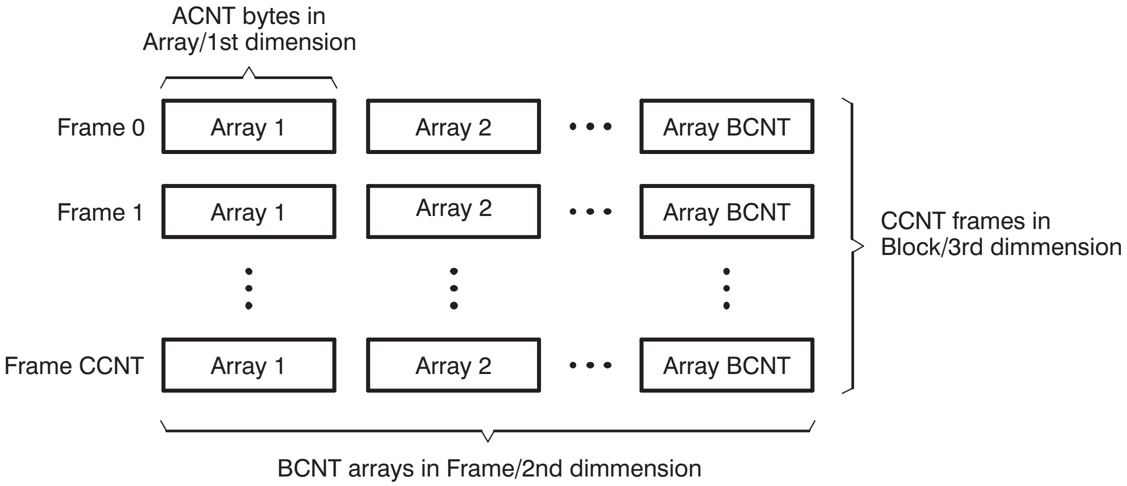
10.3.2 Types of EDMA3 Transfers

An EDMA3 transfer is always defined in terms of three dimensions. Figure 10-6 shows the three dimensions used by EDMA3 transfers. These three dimensions are defined as:

- 1st Dimension or Array (A): The 1st dimension in a transfer consists of ACNT contiguous bytes.
- 2nd Dimension or Frame (B): The 2nd dimension in a transfer consists of BCNT arrays of ACNT bytes. Each array transfer in the 2nd dimension is separated from each other by an index programmed using SRCBIDX or DSTBIDX.
- 3rd Dimension or Block (C): The 3rd dimension in a transfer consists of CCNT frames of BCNT arrays of ACNT bytes. Each transfer in the 3rd dimension is separated from the previous by an index programmed using SRCCIDX or DSTCIDX.

Note that the reference point for the index depends on the synchronization type. The amount of data transferred upon receipt of a trigger/synchronization event is controlled by the synchronization types (SYNCDIM bit in OPT). Of the three dimensions, only two synchronization types are supported: A-synchronized transfers and AB-synchronized transfers.

Figure 10-6. Definition of ACNT, BCNT, and CCNT



10.3.2.1 A-Synchronized Transfers

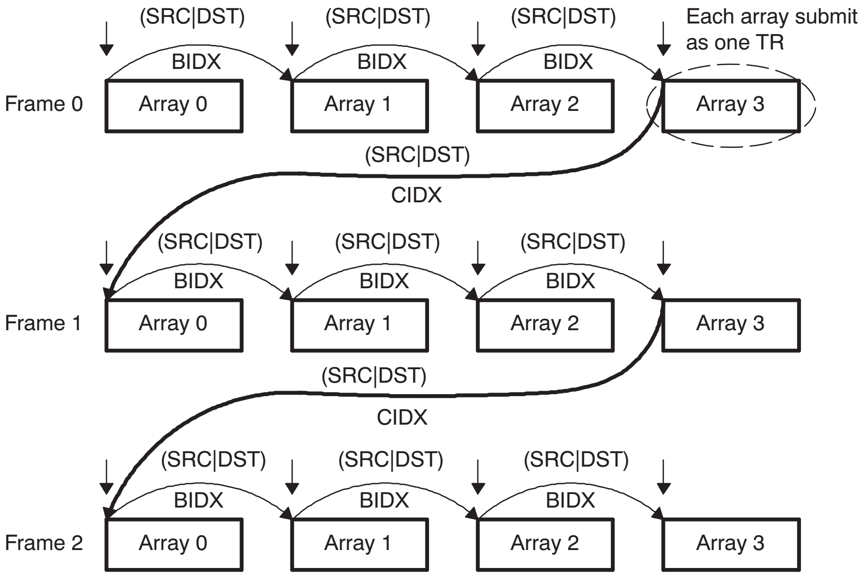
In an A-synchronized transfer, each EDMA3 sync event initiates the transfer of the 1st dimension of ACNT bytes, or one array of ACNT bytes. In other words, each event/TR packet conveys the transfer information for one array only. Thus, BCNT × CCNT events are needed to completely service a PaRAM set.

Arrays are always separated by SRCBIDX and DSTBIDX, as shown in Figure 10-7, where the start address of Array N is equal to the start address of Array N – 1 plus source (SRC) or destination (DST) BIDX.

Frames are always separated by SRCCIDX and DSTCIDX. For A-synchronized transfers, after the frame is exhausted, the address is updated by adding SRCCIDX/DSTCIDX to the beginning address of the last array in the frame. As in Figure 10-7, SRCCIDX/DSTCIDX is the difference between the start of Frame 0 Array 3 to the start of Frame 1 Array 0.

Figure 10-7 shows an A-synchronized transfer of 3 (CCNT) frames of 4 (BCNT) arrays of n (ACNT) bytes. In this example, a total of 12 sync events (BCNT × CCNT) exhaust a PaRAM set. See Section 10.3.3.6 for details on parameter set updates.

Figure 10-7. A-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3)



10.3.2.2 AB-Synchronized Transfers

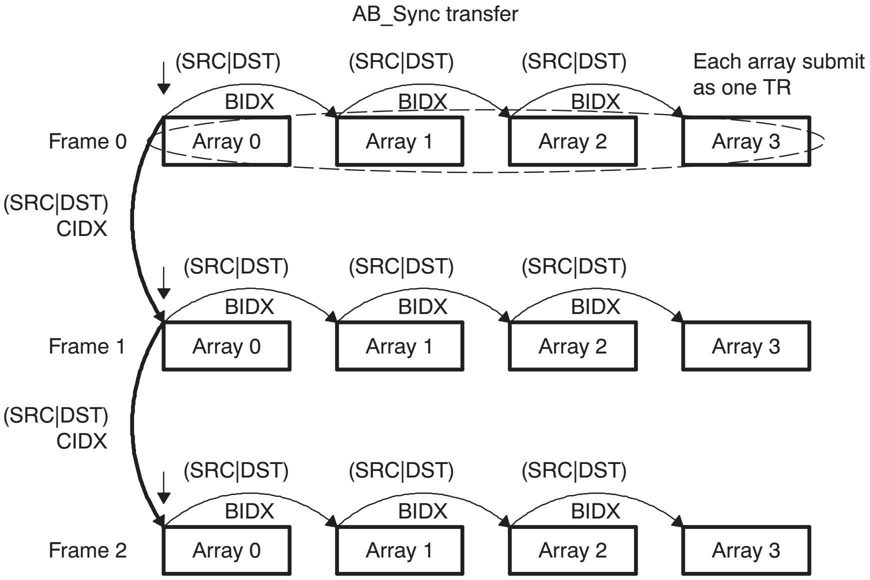
In a AB-synchronized transfer, each EDMA3 sync event initiates the transfer of 2 dimensions or one frame. In other words, each event/TR packet conveys information for one entire frame of BCNT arrays of ACNT bytes. Thus, CCNT events are needed to completely service a PaRAM set.

Arrays are always separated by SRCBIDX and DSTBIDX as shown in Figure 10-8. Frames are always separated by SRCCIDX and DSTCIDX.

Note that for AB-synchronized transfers, after a TR for the frame is submitted, the address update is to add SRCCIDX/DSTCIDX to the beginning address of the beginning array in the frame. This is different from A-synchronized transfers where the address is updated by adding SRCCIDX/DSTCIDX to the start address of the last array in the frame. See Section 10.3.3.6 for details on parameter set updates.

Figure 10-8 shows an AB-synchronized transfer of 3 (CCNT) frames of 4 (BCNT) arrays of *n* (ACNT) bytes. In this example, a total of 3 sync events (CCNT) exhaust a PaRAM set; that is, a total of 3 transfers of 4 arrays each completes the transfer.

Figure 10-8. AB-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3)



NOTE: ABC-synchronized transfers are not directly supported. But can be logically achieved by chaining between multiple AB-synchronized transfers.

10.3.3 Parameter RAM (PaRAM)

The EDMA3 controller is a RAM-based architecture. The transfer context (source/destination addresses, count, indexes, etc.) for DMA or QDMA channels is programmed in a parameter RAM table within EDMA3CC, referred to as PaRAM. The PaRAM table is segmented into multiple PaRAM sets. Each PaRAM set includes eight four-byte PaRAM set entries (32-bytes total per PaRAM set), which includes typical DMA transfer parameters such as source address, destination address, transfer counts, indexes, options, etc.

The PaRAM structure supports flexible ping-pong, circular buffering, channel chaining, and auto-reloading (linking).

The contents of the PaRAM include the following:

- 256 PaRAM sets
- 64 channels that are direct mapped and can be used as link or QDMA sets if not used for DMA channels
- 64 channels remain for link or QDMA sets

By default, all channels map to PaRAM set to 0. These should be remapped before use. For more information, see DCHMAP registers and QCHMAP registers.

Table 10-5. EDMA3 Parameter RAM Contents

PaRAM Set Number	Address	Parameters
0	EDMA Base Address + 4000h to EDMA Base Address + 401Fh	PaRAM set 0
1	EDMA Base Address + 4020h to EDMA Base Address + 403Fh	PaRAM set 1
2	EDMA Base Address + 4040h to EDMA Base Address + 405Fh	PaRAM set 2
3	EDMA Base Address + 4060h to EDMA Base Address + 407Fh	PaRAM set 3
4	EDMA Base Address + 4080h to EDMA Base Address + 409Fh	PaRAM set 4
5	EDMA Base Address + 40A0h to EDMA Base Address + 40BFh	PaRAM set 5
6	EDMA Base Address + 40C0h to EDMA Base Address + 40DFh	PaRAM set 6
7	EDMA Base Address + 40E0h to EDMA Base Address + 40FFh	PaRAM set 7
8	EDMA Base Address + 4100h to EDMA Base Address + 411Fh	PaRAM set 8
9	EDMA Base Address + 4120h to EDMA Base Address + 413Fh	PaRAM set 9
...	...	...
63	EDMA Base Address + 47E0h to EDMA Base Address + 47FFh	PaRAM set 63
64	EDMA Base Address + 4800h to EDMA Base Address + 481Fh	PaRAM set 64
65	EDMA Base Address + 4820h to EDMA Base Address + 483Fh	PaRAM set 65
...	...	...
254	EDMA Base Address + 5FC0h to EDMA Base Address + 5FDFh	PaRAM set 254
255	EDMA Base Address + 5FE0h to EDMA Base Address + 5FFFh	PaRAM set 255

10.3.3.1 PaRAM

Each parameter set of PaRAM is organized into eight 32-bit words or 32 bytes, as shown in Figure 10-9 and described in Table 10-6. Each PaRAM set consists of 16-bit and 32-bit parameters.

Figure 10-9. PaRAM Set

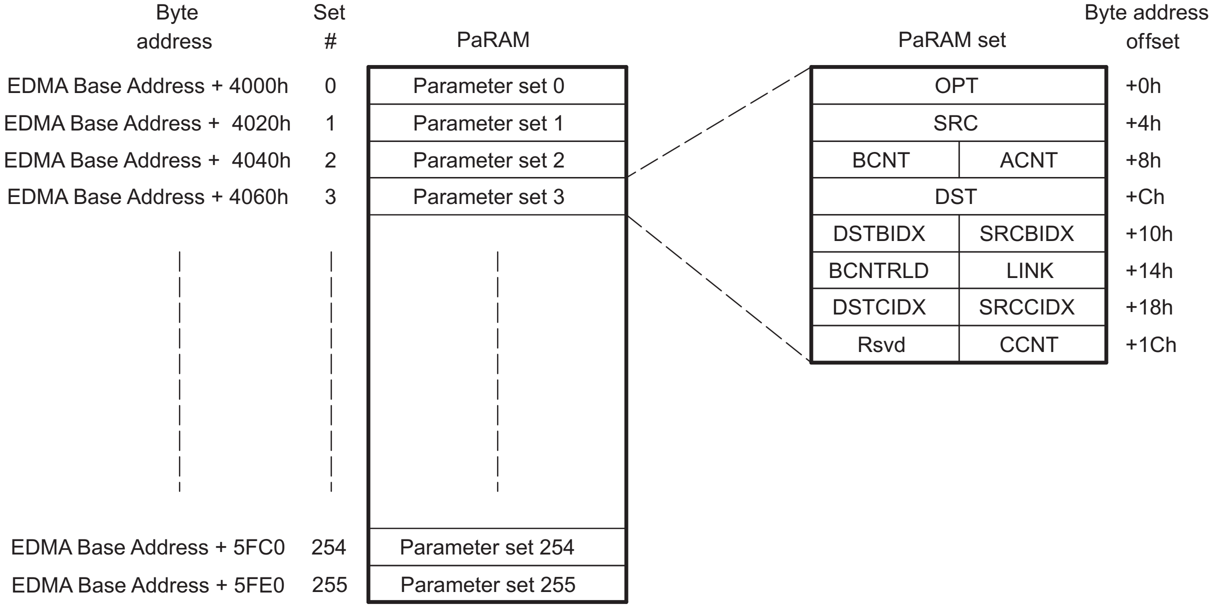


Table 10-6. EDMA3 Channel Parameter Description

Offset Address (bytes)	Acronym	Parameter	Description
0h	OPT	Channel Options	Transfer configuration options
4h	SRC	Channel Source Address	The byte address from which data is transferred
8h(1)	ACNT	Count for 1st Dimension	Unsigned value specifying the number of contiguous bytes within an array (first dimension of the transfer). Valid values range from 1 to 65 535.
	BCNT	Count for 2nd Dimension	Unsigned value specifying the number of arrays in a frame, where an array is ACNT bytes. Valid values range from 1 to 65 535.
Ch	DST	Channel Destination Address	The byte address to which data is transferred
10h(1)	SRCBIDX	Source BCNT Index	Signed value specifying the byte address offset between source arrays within a frame (2nd dimension). Valid values range from –32 768 and 32 767.
	DSTBIDX	Destination BCNT Index	Signed value specifying the byte address offset between destination arrays within a frame (2nd dimension). Valid values range from –32 768 and 32 767.
14h(1)	LINK	Link Address	The PaRAM address containing the PaRAM set to be linked (copied from) when the current PaRAM set is exhausted. A value of FFFFh specifies a null link.
	BCNTRLD	BCNT Reload	The count value used to reload BCNT when BCNT decrements to 0 (TR is submitted for the last array in 2nd dimension). Only relevant in A-synchronized transfers.
18h(1)	SRCCIDX	Source CCNT Index	Signed value specifying the byte address offset between frames within a block (3rd dimension). Valid values range from –32 768 and 32 767. A-synchronized transfers: The byte address offset from the beginning of the last source array in a frame to the beginning of the first source array in the next frame. AB-synchronized transfers: The byte address offset from the beginning of the first source array in a frame to the beginning of the first source array in the next frame.
	DSTCIDX	Destination CCNT index	Signed value specifying the byte address offset between frames within a block (3rd dimension). Valid values range from –32 768 and 32 767. A-synchronized transfers: The byte address offset from the beginning of the last destination array in a frame to the beginning of the first destination array in the next frame. AB-synchronized transfers: The byte address offset from the beginning of the first destination array in a frame to the beginning of the first destination array in the next frame.
1Ch	CCNT	Count for 3rd Dimension	Unsigned value specifying the number of frames in a block, where a frame is BCNT arrays of ACNT bytes. Valid values range from 1 to 65 535.
	RSVD	Reserved	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.

(1) If OPT, SRC, or DST is the trigger word for a QDMA transfer, then doing a 32-bit access is required for that field. TI also recommends performing only 32-bit accesses on the parameter RAM for best code compatibility. For example, switching the endianness of the processor swaps addresses of the 16-bit fields, but 32-bit accesses avoid the issue entirely.

10.3.3.2 EDMA3 Channel PaRAM Set Entry Fields

10.3.3.2.1 Channel Options Parameter (OPT)

The channel options parameter (OPT) is shown in Figure 10-10 and described in Table 10-7.

Figure 10-10. Channel Options Parameter (OPT)

31	30 28	27 24	23	22	21	20	19 18	17 16
PRIV	Reserved	PRIVID	ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	Reserved	TCC
R-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15 12	11	10 8	7 4	3	2	1	0
TCC	TCCMODE	FWID	Reserved	STATIC	SYNCDIM	DAM	SAM
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Table 10-7. Channel Options Parameters (OPT) Field Descriptions

Bit	Field	Value	Description
31	PRIV		Privilege level (supervisor versus user) for the host/CPU/DMA that programmed this PaRAM set. This value is set with the EDMA3 master's privilege value when any part of the PaRAM set is written.
		0	User level privilege.
		1	Supervisor level privilege.
30-28	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
27-24	PRIVID	0-Fh	Privilege identification for the external host/CPU/DMA that programmed this PaRAM set. This value is set with the EDMA3 master's privilege identification value when any part of the PaRAM set is written.
23	ITCCHEN		Intermediate transfer completion chaining enable.
		0	Intermediate transfer complete chaining is disabled.
		1	Intermediate transfer complete chaining is enabled.
			When enabled, the chained event register (CER/CERH) bit is set on every intermediate chained transfer completion (upon completion of every intermediate TR in the PaRAM set, except the final TR in the PaRAM set). The bit (position) set in CER or CERH is the TCC value specified.
22	TCCHEN		Transfer complete chaining enable.
		0	Transfer complete chaining is disabled.
		1	Transfer complete chaining is enabled.
			When enabled, the chained event register (CER/CERH) bit is set on final chained transfer completion (upon completion of the final TR in the PaRAM set). The bit (position) set in CER or CERH is the TCC value specified.
21	ITCINTEN		Intermediate transfer completion interrupt enable.
		0	Intermediate transfer complete interrupt is disabled.
		1	Intermediate transfer complete interrupt is enabled.
			When enabled, the interrupt pending register (IPR / IPRH) bit is set on every intermediate transfer completion (upon completion of every intermediate TR in the PaRAM set, except the final TR in the PaRAM set). The bit (position) set in IPR or IPRH is the TCC value specified. To generate a completion interrupt to the CPU, the corresponding IER [TCC] / IERH [TCC] bit must be set.
20	TCINTEN		Transfer complete interrupt enable.
		0	Transfer complete interrupt is disabled.
		1	Transfer complete interrupt is enabled.
			When enabled, the interrupt pending register (IPR / IPRH) bit is set on transfer completion (upon completion of the final TR in the PaRAM set). The bit (position) set in IPR or IPRH is the TCC value specified. To generate a completion interrupt to the CPU, the corresponding IER[TCC] / IERH [TCC] bit must be set.
19-18	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.

Table 10-7. Channel Options Parameters (OPT) Field Descriptions (continued)

Bit	Field	Value	Description
17-12	TCC	0-3Fh	Transfer complete code. This 6-bit code sets the relevant bit in the chaining enable register (CER [TCC] /CERH [TCC]) for chaining or in the interrupt pending register (IPR [TCC] / IPRH [TCC]) for interrupts.
11	TCCMODE		Transfer complete code mode. Indicates the point at which a transfer is considered completed for chaining and interrupt generation.
		0	Normal completion: A transfer is considered completed after the data has been transferred.
		1	Early completion: A transfer is considered completed after the EDMA3CC submits a TR to the EDMA3TC. TC may still be transferring data when the interrupt/chain is triggered.
10-8	FWID	0-7h	FIFO Width. Applies if either SAM or DAM is set to constant addressing mode.
		0	FIFO width is 8-bit.
		1h	FIFO width is 16-bit.
		2h	FIFO width is 32-bit.
		3h	FIFO width is 64-bit.
		4h	FIFO width is 128-bit.
		5h	FIFO width is 256-bit.
		6h-7h	Reserved.
7-4	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
3	STATIC		Static set.
		0	Set is not static. The PaRAM set is updated or linked after a TR is submitted. A value of 0 should be used for DMA channels and for non-final transfers in a linked list of QDMA transfers.
		1	Set is static. The PaRAM set is not updated or linked after a TR is submitted. A value of 1 should be used for isolated QDMA transfers or for the final transfer in a linked list of QDMA transfers.
2	SYNCDIM		Transfer synchronization dimension.
		0	A-synchronized. Each event triggers the transfer of a single array of ACNT bytes.
		1	AB-synchronized. Each event triggers the transfer of BCNT arrays of ACNT bytes.
1	DAM		Destination address mode.
		0	Increment (INCR) mode. Destination addressing within an array increments. Destination is not a FIFO.
		1	Constant addressing (CONST) mode. Destination addressing within an array wraps around upon reaching FIFO width.
0	SAM		Source address mode.
		0	Increment (INCR) mode. Source addressing within an array increments. Source is not a FIFO.
		1	Constant addressing (CONST) mode. Source addressing within an array wraps around upon reaching FIFO width.

10.3.3.2.2 Channel Source Address (SRC)

The 32-bit source address parameter specifies the starting byte address of the source. For SAM in increment mode, there are no alignment restrictions imposed by EDMA3. For SAM in constant addressing mode, you must program the source address to be aligned to a 256-bit aligned address (5 LSBs of address must be 0). The EDMA3TC will signal an error, if this rule is violated. See Section 10.3.12.3 for additional details.

10.3.3.2.3 Channel Destination Address (DST)

The 32-bit destination address parameter specifies the starting byte address of the destination. For DAM in increment mode, there are no alignment restrictions imposed by EDMA3. For DAM in constant addressing mode, you must program the destination address to be aligned to a 256-bit aligned address (5 LSBs of address must be 0). The EDMA3TC will signal an error, if this rule is violated. See Section 10.3.12.3 for additional details.

10.3.3.2.4 Count for 1st Dimension (ACNT)

ACNT represents the number of bytes within the 1st dimension of a transfer. ACNT is a 16-bit unsigned value with valid values between 0 and 65 535. Therefore, the maximum number of bytes in an array is 65 535 bytes (64K – 1 bytes). ACNT must be greater than or equal to 1 for a TR to be submitted to EDMA3TC. A transfer with ACNT equal to 0 is considered either a null or dummy transfer. A dummy or null transfer generates a completion code depending on the settings of the completion bit fields in OPT.

See Section 10.3.3.5 and Section 10.3.5.3 for details on dummy/null completion conditions.

10.3.3.2.5 Count for 2nd Dimension (BCNT)

BCNT is a 16-bit unsigned value that specifies the number of arrays of length ACNT. For normal operation, valid values for BCNT are between 1 and 65 535. Therefore, the maximum number of arrays in a frame is 65 535 (64K – 1 arrays). A transfer with BCNT equal to 0 is considered either a null or dummy transfer. A dummy or null transfer generates a completion code depending on the settings of the completion bit fields in OPT.

See Section 10.3.3.5 and Section 10.3.5.3 for details on dummy/null completion conditions.

10.3.3.2.6 Count for 3rd Dimension (CCNT)

CCNT is a 16-bit unsigned value that specifies the number of frames in a block. Valid values for CCNT are between 1 and 65 535. Therefore, the maximum number of frames in a block is 65 535 (64K – 1 frames). A transfer with CCNT equal to 0 is considered either a null or dummy transfer. A dummy or null transfer generates a completion code depending on the settings of the completion bit fields in OPT.

A CCNT value of 0 is considered either a null or dummy transfer. See Section 10.3.3.5 and Section 10.3.5.3 for details on dummy/null completion conditions.

10.3.3.2.7 BCNT Reload (BCNTRLD)

BCNTRLD is a 16-bit unsigned value used to reload the BCNT field once the last array in the 2nd dimension is transferred. This field is only used for A-synchronized transfers. In this case, the EDMA3CC decrements the BCNT value by 1 on each TR submission. When BCNT reaches 0, the EDMA3CC decrements CCNT and uses the BCNTRLD value to reinitialize the BCNT value.

For AB-synchronized transfers, the EDMA3CC submits the BCNT in the TR and the EDMA3TC decrements BCNT appropriately. For AB-synchronized transfers, BCNTRLD is not used.

10.3.3.2.8 Source B Index (SRCBIDX)

SRCBIDX is a 16-bit signed value (2s complement) used for source address modification between each array in the 2nd dimension. Valid values for SRCBIDX are between –32 768 and 32 767. It provides a byte address offset from the beginning of the source array to the beginning of the next source array. It applies to both A-synchronized and AB-synchronized transfers. Some examples:

- SRCBIDX = 0000h (0): no address offset from the beginning of an array to the beginning of the next array. All arrays are fixed to the same beginning address.
- SRCBIDX = 0003h (+3): the address offset from the beginning of an array to the beginning of the next array in a frame is 3 bytes. For example, if the current array begins at address 1000h, the next array begins at 1003h.
- SRCBIDX = FFFFh (–1): the address offset from the beginning of an array to the beginning of the next array in a frame is –1 byte. For example, if the current array begins at address 5054h, the next array begins at 5053h.

10.3.3.2.9 Destination B Index (DSTBIDX)

DSTBIDX is a 16-bit signed value (2s complement) used for destination address modification between each array in the 2nd dimension. Valid values for DSTBIDX are between –32 768 and 32 767. It provides a byte address offset from the beginning of the destination array to the beginning of the next destination array within the current frame. It applies to both A-synchronized and AB-synchronized transfers. See SRCBIDX for examples.

10.3.3.2.10 Source C Index (SRCCIDX)

SRCCIDX is a 16-bit signed value (2s complement) used for source address modification in the 3rd dimension. Valid values for SRCCIDX are between –32 768 and 32 767. It provides a byte address offset from the beginning of the current array (pointed to by SRC address) to the beginning of the first source array in the next frame. It applies to both A-synchronized and AB-synchronized transfers. Note that when SRCCIDX is applied, the current array in an A-synchronized transfer is the last array in the frame (Figure 10-7), while the current array in an AB-synchronized transfer is the first array in the frame (Figure 10-8).

10.3.3.2.11 Destination C Index (DSTCIDX)

DSTCIDX is a 16-bit signed value (2s complement) used for destination address modification in the 3rd dimension. Valid values are between –32 768 and 32 767. It provides a byte address offset from the beginning of the current array (pointed to by DST address) to the beginning of the first destination array TR in the next frame. It applies to both A-synchronized and AB-synchronized transfers. Note that when DSTCIDX is applied, the current array in an A-synchronized transfer is the last array in the frame (Figure 10-7), while the current array in a AB-synchronized transfer is the first array in the frame (Figure 10-8).

10.3.3.2.12 Link Address (LINK)

The EDMA3CC provides a mechanism, called linking, to reload the current PaRAM set upon its natural termination (that is, after the count fields are decremented to 0) with a new PaRAM set. The 16-bit parameter LINK specifies the byte address offset in the PaRAM from which the EDMA3CC loads/reloads the next PaRAM set during linking.

You must program the link address to point to a valid aligned 32-byte PaRAM set. The 5 LSBs of the LINK field should be cleared to 0.

The EDMA3CC ignores the upper 2 bits of the LINK entry, allowing the programmer the flexibility of programming the link address as either an absolute/literal byte address or use the PaRAM-base-relative offset address. Therefore, if you make use of the literal address with a range from 4000h to 7FFFh, it will be treated as a PaRAM-base-relative value of 0000h to 3FFFh.

You should make sure to program the LINK field correctly, so that link update is requested from a PaRAM address that falls in the range of the available PaRAM addresses on the device.

A LINK value of FFFFh is referred to as a NULL link that should cause the EDMA3CC to perform an internal write of 0 to all entries of the current PaRAM set, except for the LINK field that is set to FFFFh. Also, see Section 10.3.5 for details on terminating a transfer.

10.3.3.3 Null PaRAM Set

A null PaRAM set is defined as a PaRAM set where all count fields (ACNT, BCNT, and CCNT) are cleared to 0. If a PaRAM set associated with a channel is a NULL set, then when serviced by the EDMA3CC, the bit corresponding to the channel is set in the associated event missed register (EMR, EMRH, or QEMR). This bit remains set in the associated secondary event register (SER, SERH, or QSER). *This implies that any future events on the same channel are ignored by the EDMA3CC and you are required to clear the bit in SER, SERH, or QSER for the channel.* This is considered an error condition, since events are not expected on a channel that is configured as a null transfer. For more information, see the SER and EMR registers.

10.3.3.4 Dummy PaRAM Set

A dummy PaRAM set is defined as a PaRAM set where at least one of the count fields (ACNT, BCNT, or CCNT) is cleared to 0 and at least one of the count fields is nonzero.

If a PaRAM set associated with a channel is a dummy set, then when serviced by the EDMA3CC, it will not set the bit corresponding to the channel (DMA/QDMA) in the event missed register (EMR, EMRH, or QEMR) and the secondary event register (SER, SERH, or QSER) bit gets cleared similar to a normal transfer. Future events on that channel are serviced. A dummy transfer is a legal transfer of 0 bytes. For more information, see the SER and EMR registers.

10.3.3.5 Dummy Versus Null Transfer Comparison

There are some differences in the way the EDMA3CC logic treats a dummy versus a null transfer request. A null transfer request is an error condition, but a dummy transfer is a legal transfer of 0 bytes. A null transfer causes an error bit (E*n*) in EMR to get set and the E*n* bit in SER remains set, essentially preventing any further transfers on that channel without clearing the associated error registers.

Table 10-8 summarizes the conditions and effects of null and dummy transfer requests.

Table 10-8. Dummy and Null Transfer Request

Feature	Null TR	Dummy TR
EMR/EMRH/QEMR is set	Yes	No
SER/SERH/QSER remains set	Yes	No
Link update (STATIC = 0 in OPT)	Yes	Yes
QER is set	Yes	Yes
IPR/IPRH CER/CERH is set using early completion	Yes	Yes

10.3.3.6 Parameter Set Updates

When a TR is submitted for a given DMA/QDMA channel and its corresponding PaRAM set, the EDMA3CC is responsible for updating the PaRAM set in anticipation of the next trigger event. For events that are not final, this includes address and count updates; for final events, this includes the link update.

The specific PaRAM set entries that are updated depend on the channel’s synchronization type (A-synchronized or B-synchronized) and the current state of the PaRAM set. A B-update refers to the decrementing of BCNT in the case of A-synchronized transfers after the submission of successive TRs. A C-update refers to the decrementing of CCNT in the case of A-synchronized transfers after BCNT TRs for ACNT byte transfers have submitted. For AB-synchronized transfers, a C-update refers to the decrementing of CCNT after submission of every transfer request.

See Table 10-9 for details and conditions on the parameter updates. A link update occurs when the PaRAM set is exhausted, as described in Section 10.3.3.7.

After the TR is read from the PaRAM (and is in process of being submitted to EDMA3TC), the following fields are updated if needed:

- A-synchronized: BCNT, CCNT, SRC, DST.
- AB-synchronized: CCNT, SRC, DST.

The following fields are not updated (except for during linking, where all fields are overwritten by the link PaRAM set):

- A-synchronized: ACNT, BCNTRLD, SRCBIDX, DSTBIDX, SRCCIDX, DSTCIDX, OPT, LINK.
- AB-synchronized: ACNT, BCNT, BCNTRLD, SRCBIDX, DSTBIDX, SRCCIDX, DSTCIDX, OPT, LINK.

Note that PaRAM updates only pertain to the information that is needed to properly submit the next transfer request to the EDMA3TC. Updates that occur while data is moved within a transfer request are tracked within the transfer controller, and is detailed in Section 10.3.12. For A-synchronized transfers, the EDMA3CC always submits a TRP for ACNT bytes (BCNT = 1 and CCNT = 1). For AB-synchronized transfers, the EDMA3CC always submits a TRP for ACNT bytes of BCNT arrays (CCNT = 1). The EDMA3TC is responsible for updating source and destination addresses within the array based on ACNT and FWID (in OPT). For AB-synchronized transfers, the EDMA3TC is also responsible to update source and destination addresses between arrays based on SRCBIDX and DSTBIDX.

Table 10-9 shows the details of parameter updates that occur within EDMA3CC for A-synchronized and AB-synchronized transfers.

Table 10-9. Parameter Updates in EDMA3CC (for Non-Null, Non-Dummy PaRAM Set)

	A-Synchronized Transfer			AB-Synchronized Transfer		
	B-Update	C-Update	Link Update	B-Update	C-Update	Link Update
Condition:	**BCNT > 1**	**BCNT == 1 && CCNT > 1**	**BCNT == 1 && CCNT == 1**	**N/A**	**CCNT > 1**	**CCNT == 1**
SRC	+= SRCBIDX	+= SRCCIDX	= Link.SRC	in EDMA3TC	+= SRCCIDX	= Link.SRC
DST	+= DSTBIDX	+= DSTCIDX	= Link.DST	in EDMA3TC	+= DSTCIDX	= Link.DST
ACNT	None	None	= Link.ACNT	None	None	= Link.ACNT
BCNT	–= 1	= BCNTRLD	= Link.BCNT	in EDMA3TC	N/A	= Link.BCNT
CCNT	None	–= 1	= Link.CCNT	in EDMA3TC	–=1	= Link.CCNT
SRCBIDX	None	None	= Link.SRCBIDX	in EDMA3TC	None	= Link.SRCBIDX
DSTBIDX	None	None	= Link.DSTBIDX	None	None	= Link.DSTBIDX
SRCCIDX	None	None	= Link.SRCBIDX	in EDMA3TC	None	= Link.SRCBIDX
DSTCIDX	None	None	= Link.DSTBIDX	None	None	= Link.DSTBIDX
LINK	None	None	= Link.LINK	None	None	= Link.LINK
BCNTRLD	None	None	= Link.BCNTRLD	None	None	= Link.BCNTRLD
OPT(1)	None	None	= LINK.OPT	None	None	= LINK.OPT

(1) In all cases, no updates occur if OPT.STATIC == 1 for the current PaRAM set.

NOTE: The EDMA3CC includes no special hardware to detect when an indexed address update calculation overflows/underflows. The address update will wrap across boundaries as programmed by the user. You should ensure that no transfer is allowed to cross internal port boundaries between peripherals. A single TR must target a single source/destination slave endpoint.

10.3.3.7 Linking Transfers

The EDMA3CC provides a mechanism known as linking, which allows the entire PaRAM set to be reloaded from a location within the PaRAM memory map (for both DMA and QDMA channels). Linking is especially useful for maintaining ping-pong buffers, circular buffering, and repetitive/continuous transfers with no CPU intervention. Upon completion of a transfer, the current transfer parameters are reloaded with the parameter set pointed that the 16-bit link address field of the current parameter set points to. Linking only occurs when the STATIC bit in OPT is cleared.

NOTE: You should always link a transfer (EDMA3 or QDMA) to another useful transfer. If you must terminate a transfer, then you should link the transfer to a NULL parameter set. See Section 10.3.3.3.

The link update occurs after the current PaRAM set event parameters have been exhausted. An event's parameters are exhausted when the EDMA3 channel controller has submitted all of the transfers that are associated with the PaRAM set.

A link update occurs for null and dummy transfers depending on the state of the STATIC bit in OPT and the LINK field. In both cases (null or dummy), if the value of LINK is FFFFh, then a null PaRAM set (with all 0s and LINK set to FFFFh) is written to the current PaRAM set. Similarly, if LINK is set to a value other than FFFFh, then the appropriate PaRAM location that LINK points to is copied to the current PaRAM set.

Once the channel completion conditions are met for an event, the transfer parameters that are located at the link address are loaded into the current DMA or QDMA channel’s associated parameter set. This indicates that the EDMA3CC reads the entire set (eight words) from the PaRAM set specified by LINK and writes all eight words to the PaRAM set that is associated with the current channel. Figure 10-11 shows an example of a linked transfer.

Any PaRAM set in the PaRAM can be used as a link/reload parameter set. The PaRAM sets associated with peripheral synchronization events (see Section 10.3.6) should only be used for linking if the corresponding events are disabled.

If a PaRAM set location is defined as a QDMA channel PaRAM set (by QCHMAP*n*), then copying the link PaRAM set into the current QDMA channel PaRAM set is recognized as a trigger event. It is latched in QER because a write to the trigger word was performed. You can use this feature to create a linked list of transfers using a single QDMA channel and multiple PaRAM sets. See Section 10.3.4.2.

Linking to itself replicates the behavior of auto-initialization, thus facilitating the use of circular buffering and repetitive transfers. After an EDMA3 channel exhausts its current PaRAM set, it reloads all of the parameter set entries from another PaRAM set, which is initialized with values that are identical to the original PaRAM set. Figure 10-11 shows an example of a linked to self transfer. Here, the PaRAM set 255 has the link field pointing to the address of parameter set 255 (linked to self).

NOTE: If the STATIC bit in OPT is set for a PaRAM set, then link updates are not performed.

10.3.3.8 Constant Addressing Mode Transfers/Alignment Issues

If either SAM or DAM is set (constant addressing mode), then the source or destination address must be aligned to a 256-bit aligned address, respectively, and the corresponding BIDX should be an even multiple of 32 bytes (256 bits). The EDMA3CC does not recognize errors here, but the EDMA3TC asserts an error if this is not true. See Section 10.3.12.3.

NOTE: The constant addressing (CONST) mode has limited applicability. The EDMA3 should be configured for the constant addressing mode (SAM/DAM = 1) only if the transfer source or destination (on-chip memory, off-chip memory controllers, slave peripherals) support the constant addressing mode. See the device-specific data manual or chapter in this TRM to verify if the constant addressing mode is supported. If the constant addressing mode is not supported, the similar logical transfer can be achieved using the increment (INCR) mode (SAM/DAM =0) by appropriately programming the count and indices values.

10.3.3.9 Element Size

The EDMA3 controller does not use element-size and element-indexing. Instead, all transfers are defined in terms of all three dimensions: ACNT, BCNT, and CCNT. An element-indexed transfer is logically achieved by programming ACNT to the size of the element and BCNT to the number of elements that need to be transferred. For example, if you have 16-bit audio data and 256 audio samples that must be transferred to a serial port, you can only do this by programming the ACNT = 2 (2 bytes) and BCNT = 256.

Figure 10-11. Linked Transfer

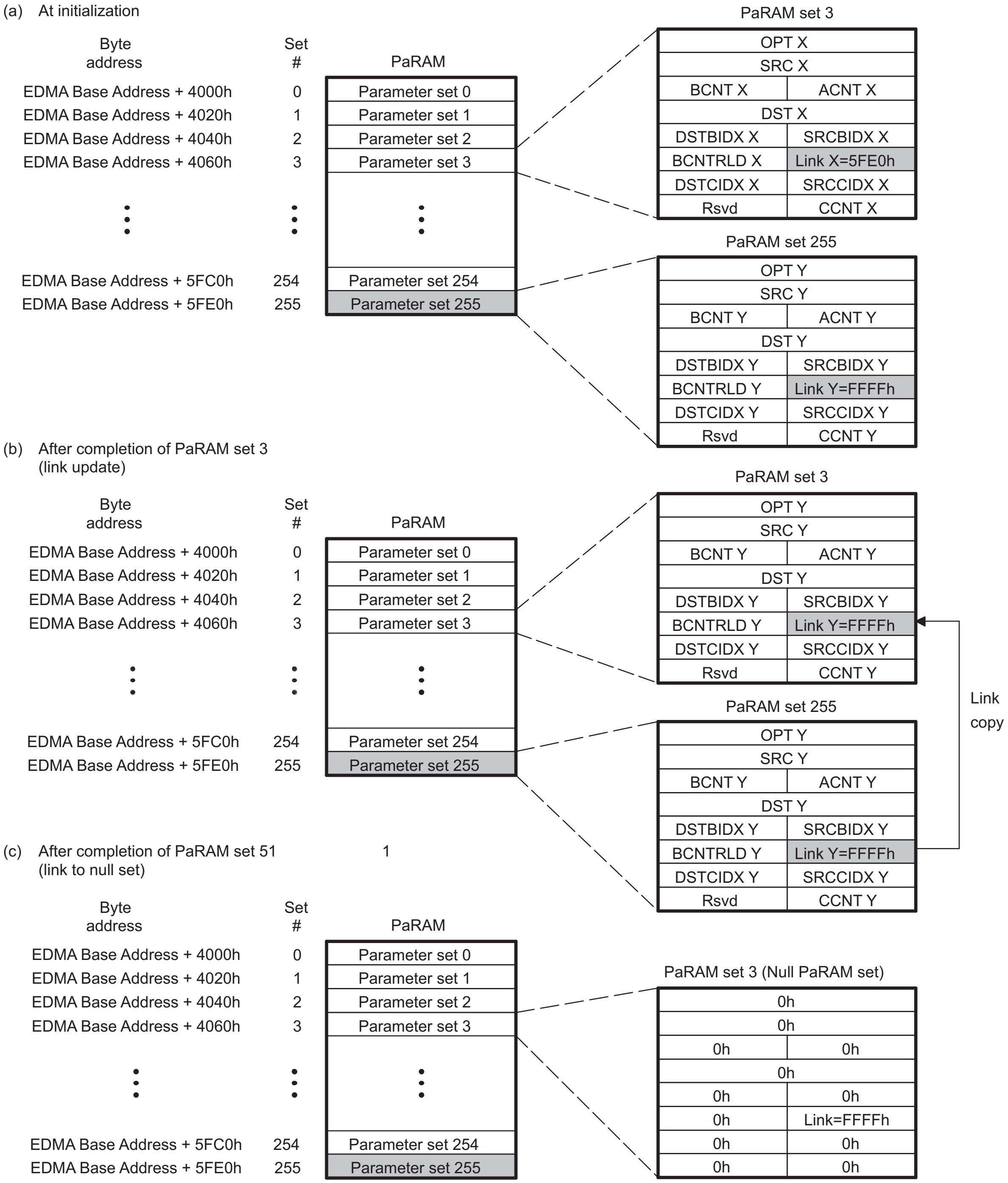
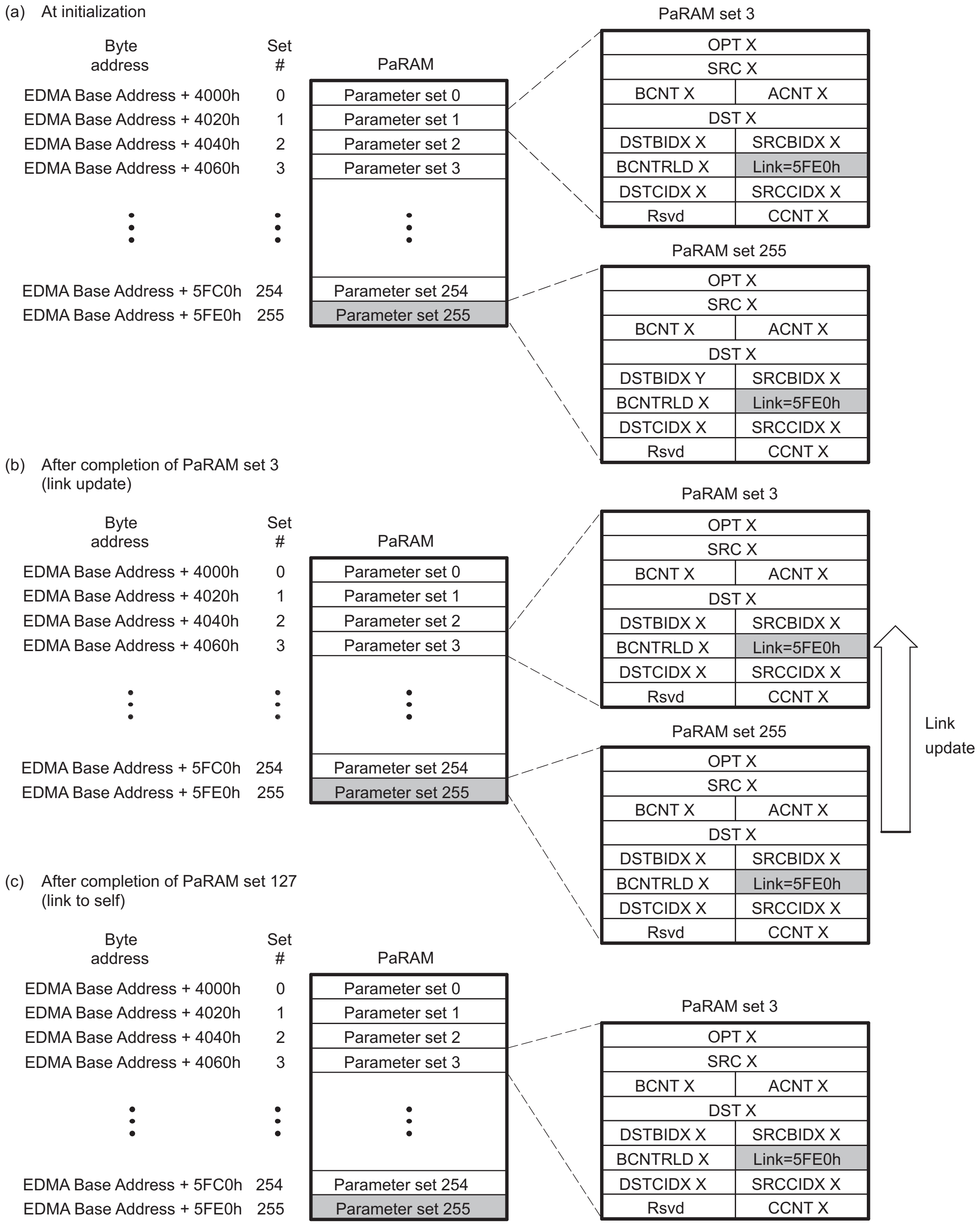


Figure 10-12. Link-to-Self Transfer



10.3.4 Initiating a DMA Transfer

There are multiple ways to initiate a programmed data transfer using the EDMA3 channel controller. Transfers on DMA channels are initiated by three sources.

They are listed as follows:

- **Event-triggered transfer request** (this is the more typical usage of EDMA3): A peripheral, system, or externally-generated event triggers a transfer request.
- **Manually-triggered transfer request:**The CPU to manually triggers a transfer by writing a 1 to the

corresponding bit in the event set register (ESR/ESRH).

- **Chain-triggered transfer request:** A transfer is triggered on the completion of another transfer or sub-transfer.

Transfers on QDMA channels are initiated by two sources. They are as follows:

- **Auto-triggered transfer request:** Writing to the programmed trigger word triggers a transfer.
- **Link-triggered transfer requests:** Writing to the trigger word triggers the transfer when linking occurs.

10.3.4.1 DMA Channel

10.3.4.1.1 Event-Triggered Transfer Request

When an event is asserted from a peripheral or device pins, it gets latched in the corresponding bit of the event register (ER.E*n* = 1). If the corresponding event in the event enable register (EER) is enabled (EER.E*n* = 1), then the EDMA3CC prioritizes and queues the event in the appropriate event queue. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

If the PaRAM set is valid (not a NULL set), then a transfer request packet (TRP) is submitted to the EDMA3TC and the E*n* bit in ER is cleared. At this point, a new event can be safely received by the EDMA3CC.

If the PaRAM set associated with the channel is a NULL set (see Section 10.3.3.3), then no transfer request (TR) is submitted and the corresponding E*n* bit in ER is cleared and simultaneously the corresponding channel bit is set in the event miss register (EMR.E*n* = 1) to indicate that the event was discarded due to a null TR being serviced. Good programming practices should include cleaning the event missed error before re-triggering the DMA channel.

When an event is received, the corresponding event bit in the event register is set (ER.E*n* = 1), regardless of the state of EER.E*n*. If the event is disabled when an external event is received (ER.E*n* = 1 and EER.E*n* = 0), the ER.E*n* bit remains set. If the event is subsequently enabled (EER.E*n* = 1), then the pending event is processed by the EDMA3CC and the TR is processed/submitted, after which the ER.E*n* bit is cleared.

If an event is being processed (prioritized or is in the event queue) and another sync event is received for the same channel prior to the original being cleared (ER.E*n* != 0), then the second event is registered as a missed event in the corresponding bit of the event missed register (EMR.E*n* = 1).

See Section 7.2.2, *EDMA Event Multiplexing*, for a description of how DMA events map to the EDMA event crossbar. See Section 10.3.20, *EDMA Events*, for a table of direct and crossbar mapped EDMA events.

10.3.4.1.2 Manually Triggered Transfer Request

The CPU or any EDMA programmer initiates a DMA transfer by writing to the event set register (ESR). Writing a 1 to an event bit in the ESR results in the event being prioritized/queued in the appropriate event queue, regardless of the state of the EER.E*n* bit. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

As in the event-triggered transfers, if the PaRAM set associated with the channel is valid (it is not a null set) then the TR is submitted to the associated EDMA3TC and the channel can be triggered again.

If the PaRAM set associated with the channel is a NULL set (see Section 10.3.3.3), then no transfer request (TR) is submitted and the corresponding E*n* bit in ER is cleared and simultaneously the corresponding channel bit is set in the event miss register (EMR.E*n* = 1) to indicate that the event was discarded due to a null TR being serviced. Good programming practices should include clearing the event missed error before re-triggering the DMA channel.

If an event is being processed (prioritized or is in the event queue) and the same channel is manually set by a write to the corresponding channel bit of the event set register (ESR.E*n* = 1) prior to the original being cleared (ESR.E*n* = 0), then the second event is registered as a missed event in the corresponding bit of the event missed register (EMR.E*n* = 1).

10.3.4.1.3 Chain-Triggered Transfer Request

Chaining is a mechanism by which the completion of one transfer automatically sets the event for another channel. When a chained completion code is detected, the value of which is dictated by the transfer completion code (TCC[5:0] in OPT of the PaRAM set associated with the channel), it results in the corresponding bit in the chained event register (CER) to be set (CER.E[TCC] = 1).

Once a bit is set in CER, the EDMA3CC prioritizes and queues the event in the appropriate event queue. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

As in the event-triggered transfers, if the PaRAM set associated with the channel is valid (it is not a null set) then the TR is submitted to the associated EDMA3TC and the channel can be triggered again.

If the PaRAM set associated with the channel is a NULL set (see Section 10.3.3.3), then no transfer request (TR) is submitted and the corresponding E*n* bit in CER is cleared and simultaneously the corresponding channel bit is set in the event miss register (EMR.E*n* = 1) to indicate that the event was discarded due to a null TR being serviced. In this case, the error condition must be cleared by you before the DMA channel can be re-triggered. Good programming practices might include clearing the event missed error before re-triggering the DMA channel.

If a chaining event is being processed (prioritized or queued) and another chained event is received for the same channel prior to the original being cleared (CER.E*n* != 0), then the second chained event is registered as a missed event in the corresponding channel bit of the event missed register (EMR.E*n* = 1).

NOTE: Chained event registers, event registers, and event set registers operate independently. An event (E*n*) can be triggered by any of the trigger sources (event-triggered, manually-triggered, or chain-triggered).

10.3.4.2 QDMA Channels

10.3.4.2.1 Auto-triggered and Link-Triggered Transfer Request

QDMA-based transfer requests are issued when a QDMA event gets latched in the QDMA event register (QER.E*n* = 1). A bit corresponding to a QDMA channel is set in the QDMA event register (QER) when the following occurs:

- A CPU (or any EDMA3 programmer) write occurs to a PaRAM address that is defined as a QDMA channel trigger word (programmed in the QDMA channel mapping register (QCHMAP*n*)) for the particular QDMA channel and the QDMA channel is enabled via the QDMA event enable register (QEER.E*n* = 1).
- EDMA3CC performs a link update on a PaRAM set address that is configured as a QDMA channel (matches QCHMAP*n* settings) and the corresponding channel is enabled via the QDMA event enable register (QEER.E*n* = 1).

Once a bit is set in QER, the EDMA3CC prioritizes and queues the event in the appropriate event queue. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

As in the event-triggered transfers, if the PaRAM set associated with the channel is valid (it is not a null set) then the TR is submitted to the associated EDMA3TC and the channel can be triggered again.

If a bit is already set in QER (QER.E*n* = 1) and a second QDMA event for the same QDMA channel occurs prior to the original being cleared, the second QDMA event gets captured in the QDMA event miss register (QEMR.E*n* = 1).

10.3.4.3 Comparison Between DMA and QDMA Channels

The primary difference between DMA and QDMA channels is the event/channel synchronization. QDMA events are either auto-triggered or link triggered. auto-triggering allows QDMA channels to be triggered by CPUs with a minimum number of linear writes to PaRAM. Link triggering allows a linked list of transfers to be executed, using a single QDMA PaRAM set and multiple link PaRAM sets.

A QDMA transfer is triggered when a CPU (or other EDMA3 programmer) writes to the trigger word of the QDMA channel parameter set (auto-triggered) or when the EDMA3CC performs a link update on a PaRAM set that has been mapped to a QDMA channel (link triggered). Note that for CPU triggered (manually triggered) DMA channels, in addition to writing to the PaRAM set, it is required to write to the event set register (ESR) to kick-off the transfer.

QDMA channels are typically for cases where a single event will accomplish a complete transfer since the CPU (or EDMA3 programmer) must reprogram some portion of the QDMA PaRAM set in order to re-trigger the channel. In other words, QDMA transfers are programmed with BCNT = CCNT = 1 for A-synchronized transfers, and CCNT = 1 for AB-synchronized transfers.

Additionally, since linking is also supported (if STATIC = 0 in OPT) for QDMA transfers, it allows you to initiate a linked list of QDMAs, so when EDMA3CC copies over a link PaRAM set (including the write to the trigger word), the current PaRAM set mapped to the QDMA channel will automatically be recognized as a valid QDMA event and initiate another set of transfers as specified by the linked set.

10.3.5 Completion of a DMA Transfer

A parameter set for a given channel is complete when the required number of transfer requests is submitted (based on receiving the number of synchronization events). The expected number of TRs for a non-null/non-dummy transfer is shown in Table 10-10 for both synchronization types along with state of the PaRAM set prior to the final TR being submitted. When the counts (BCNT and/or CCNT) are this value, the next TR results in a:

- Final chaining or interrupt codes to be sent by the transfer controllers (instead of intermediate).
- Link updates (linking to either null or another valid link set).

Table 10-10. Expected Number of Transfers for Non-Null Transfer

Sync Mode	Counts at time 0	Total # Transfers	Counts prior to final TR
A-synchronized	ACNT BCNT CCNT	(BCNT × CCNT) TRs of ACNT bytes each	BCNT == 1 && CCNT == 1
AB-synchronized	ACNT BCNT CCNT	CCNT TRs for ACNT × BCNT bytes each	CCNT == 1

You must program the PaRAM OPT field with a specific transfer completion code (TCC) along with the other OPT fields (TCCHEN, TCINTEN, ITCCHEN, and ITCINTEN bits) to indicate whether the completion code is to be used for generating a chained event or/and for generating an interrupt upon completion of a transfer.

The specific TCC value (6-bit binary value) programmed dictates which of the 64-bits in the chain event register (CER[TCC]) and/or interrupt pending register (IPR[TCC]) is set.

You can also selectively program whether the transfer controller sends back completion codes on completion of the final transfer request (TR) of a parameter set (TCCHEN or TCINTEN), for all but the final transfer request (TR) of a parameter set (ITCCHEN or ITCINTEN), or for all TRs of a parameter set (both). See Section 10.3.8 for details on chaining (intermediate/final chaining) and Section 10.3.9 for details on intermediate/final interrupt completion.

A completion detection interface exists between the EDMA3 channel controller and transfer controllers. This interface sends back information from the transfer controller to the channel controller to indicate that a specific transfer is completed. Completion of a transfer is used for generating chained events and/or generating interrupts to the CPUs.

All DMA/QDMA PaRAM sets must also specify a link address value. For repetitive transfers such as ping-pong buffers, the link address value should point to another predefined PaRAM set. Alternatively, a non-repetitive transfer should set the link address value to the null link value. The null link value is defined as FFFFh. See Section 10.3.3.7 for more details.

NOTE: Any incoming events that are mapped to a null PaRAM set results in an error condition. The error condition should be cleared before the corresponding channel is used again. See Section 10.3.3.5.

There are three ways the EDMA3CC gets updated/informed about a transfer completion: normal completion, early completion, and dummy/null completion. This applies to both chained events and completion interrupt generation.

10.3.5.1 Normal Completion

In normal completion mode (TCCMODE = 0 in OPT), the transfer or sub-transfer is considered to be complete when the EDMA3 channel controller receives the completion codes from the EDMA3 transfer controller. In this mode, the completion code to the channel controller is posted by the transfer controller after it receives a signal from the destination peripheral. Normal completion is typically used to generate an interrupt to inform the CPU that a set of data is ready for processing.

10.3.5.2 Early Completion

In early completion mode (TCCMODE = 1 in OPT), the transfer is considered to be complete when the EDMA3 channel controller submits the transfer request (TR) to the EDMA3 transfer controller. In this mode, the channel controller generates the completion code internally. Early completion is typically useful for chaining, as it allows subsequent transfers to be chained-triggered while the previous transfer is still in progress within the transfer controller, maximizing the overall throughput of the set of the transfers.

10.3.5.3 Dummy or Null Completion

This is a variation of early completion. Dummy or null completion is associated with a dummy set (Section 10.3.3.4) or null set (Section 10.3.3.3). In both cases, the EDMA3 channel controller does not submit the associated transfer request to the EDMA3 transfer controllers. However, if the set (dummy/null) has the OPT field programmed to return completion code (intermediate/final interrupt/chaining completion), then it will set the appropriate bits in the interrupt pending registers (IPR/IPRH) or chained event register (CER/CERH). The internal early completion path is used by the channel controller to return the completion codes internally (that is, EDMA3CC generates the completion code).

10.3.6 Event, Channel, and PaRAM Mapping

Several of the 64 DMA channels are tied to a specific hardware event, thus allowing events from device peripherals or external hardware to trigger transfers. A DMA channel typically requests a data transfer when it receives its event (apart from manually-triggered, chain-triggered, and other transfers). The amount of data transferred per synchronization event depends on the channel's configuration (ACNT, BCNT, CCNT, etc.) and the synchronization type (A-synchronized or AB-synchronized).

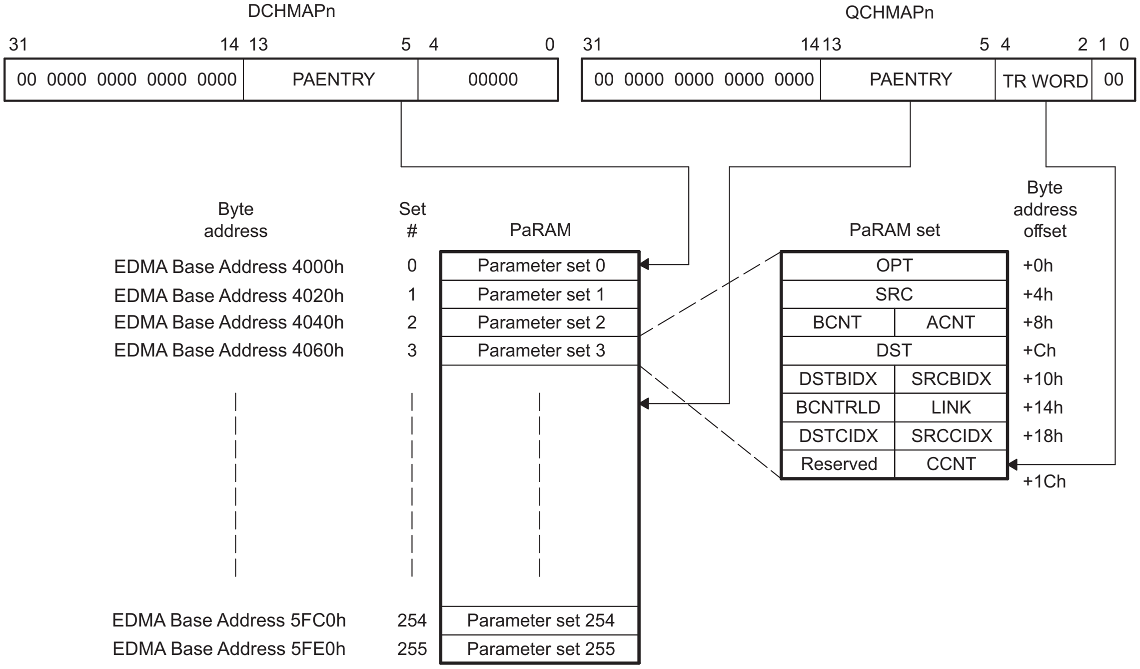
The association of an event to a channel is fixed, each DMA channel has one specific event associated with it. See Section 7.2.2, *EDMA Event Multiplexing*, for a description of how DMA events map to the EDMA event crossbar. See Section 10.3.20, *EDMA Events*, for a table of direct and crossbar mapped EDMA events.

In an application, if a channel does not use the associated synchronization event or if it does not have an associated synchronization event (unused), that channel can be used for manually-triggered or chained-triggered transfers, for linking/reloading, or as a QDMA channel.

10.3.6.1 DMA Channel to PaRAM Mapping

The mapping between the DMA channel numbers and the PaRAM sets is programmable (see Table 10-5). The DMA channel mapping registers (DCHMAPn) in the EDMA3CC provide programmability that allows the DMA channels to be mapped to any of the PaRAM sets in the PaRAM memory map. Figure 10-13 illustrates the use of DCHMAP. There is one DCHMAP register per channel.

Figure 10-13. DMA Channel and QDMA Channel to PaRAM Mapping

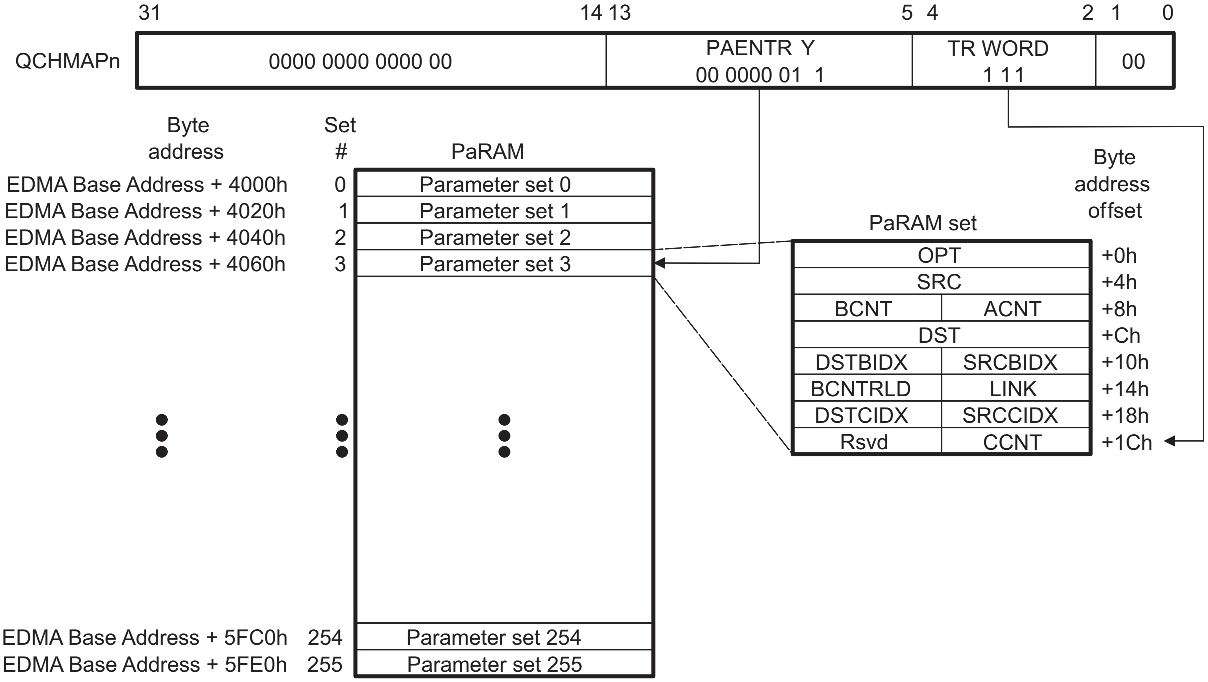


10.3.6.2 QDMA Channel to PaRAM Mapping

The mapping between the QDMA channels and the PaRAM sets is programmable. The QDMA channel mapping register (QCHMAP) in the EDMA3CC allows you to map the QDMA channels to any of the PaRAM sets in the PaRAM memory map. Figure 10-14 illustrates the use of QCHMAP.

Additionally, QCHMAP allows you to program the trigger word in the PaRAM set for the QDMA channel. A trigger word is one of the eight words in the PaRAM set. For a QDMA transfer to occur, a valid TR synchronization event for EDMA3CC is a write to the trigger word in the PaRAM set pointed to by QCHMAP for a particular QDMA channel. By default, QDMA channels are mapped to PaRAM set 0. You must appropriately re-map PaRAM set 0 before you use it.

Figure 10-14. QDMA Channel to PaRAM Mapping



10.3.7 EDMA3 Channel Controller Regions

The EDMA3 channel controller divides its address space into eight regions. Individual channel resources are assigned to a specific region, where each region is typically assigned to a specific EDMA programmer.

You can design the application software to use regions or to ignore them altogether. You can use active memory protection in conjunction with regions so that only a specific EDMA programmer (for example, privilege identification) or privilege level (for example, user vs. supervisor) is allowed access to a given region, and thus to a given DMA or QDMA channel. This allows robust system-level DMA code where each EDMA programmer only modifies the state of the assigned resources. Memory protection is described in Section 10.3.10.

10.3.7.1 Region Overview

The EDMA3 channel controller memory-mapped registers are divided in three main categories:

1. Global registers
2. Global region channel registers
3. Shadow region channel registers

The global registers are located at a single/fixed location in the EDMA3CC memory map. These registers control EDMA3 resource mapping and provide debug visibility and error tracking information.

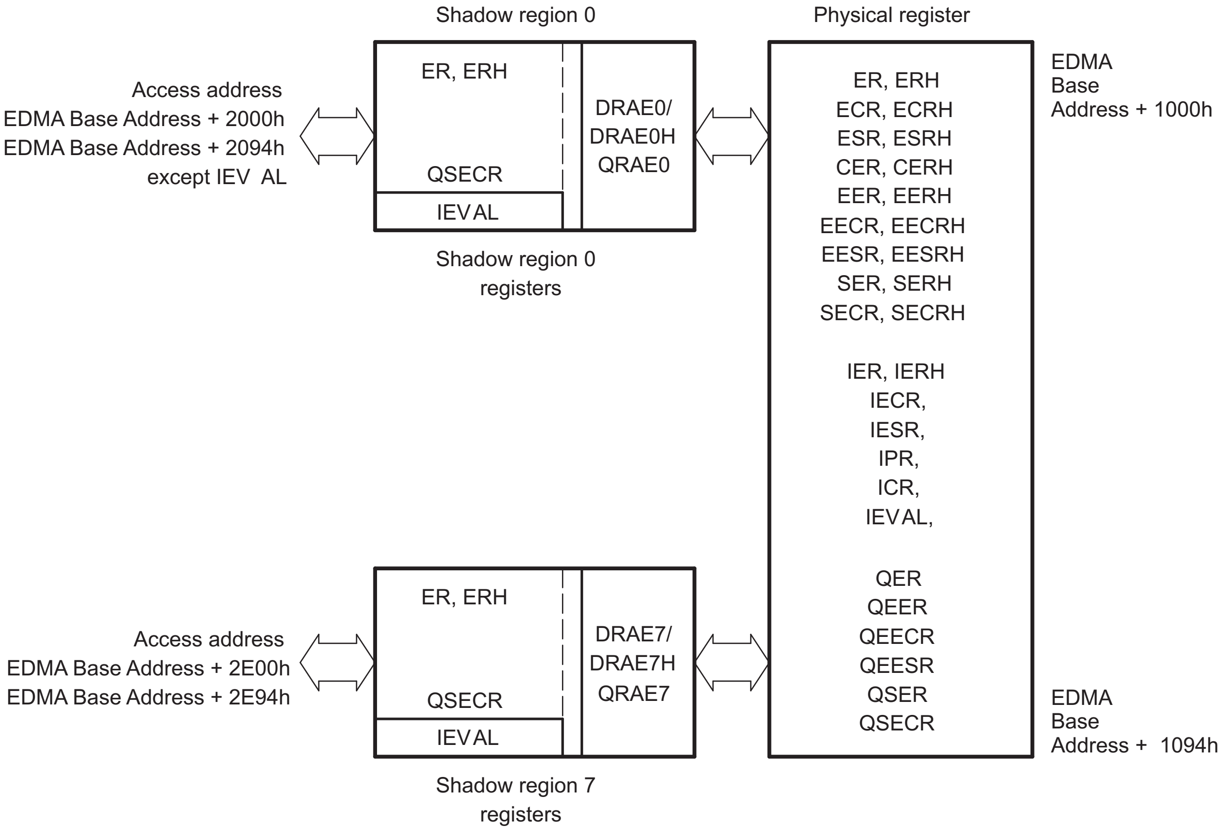
The channel registers (including DMA, QDMA, and interrupt registers) are accessible via the global channel region address range, or in the shadow *n* channel region address ranges. For example, the event enable register (EER) is visible at the global address of EDMA Base Address + 1020h or region addresses of EDMA Base Address + 2020h for region 0, EDMA Base Address + 2220h for region 1, … EDMA Base Address + 2E20h for region 7.

The DMA region access enable registers (DRAE*m*) and the QDMA region access enable registers (QRAE*n*) control the underlying control register bits that are accessible via the shadow region address space (except for IEVAL*n*). Table 10-11 lists the registers in the shadow region memory map. See the EDMA3CC memory map (Section 10.4) for the complete global memory maps. Figure 10-15 illustrates the conceptual view of the regions.

Table 10-11. Shadow Region Registers

DRAE*m*	**DRAEH*m***	**QRAE*n***
ER	ERH	QER
ECR	ECRH	QEER
ESR	ESRH	QEECR
CER	CERH	QEESR
EER	EERH	
EECR	EECRH	
EESR	EESRH	
SER	SERH	
SECR	SECRH	
IER	IERH	
IECR	IECRH	
IESR	IESRH	
IPR	IPRH	
ICR	ICRH	
Register not affected by DRAE\DRAEH		
IEVAL		

Figure 10-15. Shadow Region Registers



10.3.7.2 Channel Controller Regions

There are eight EDMA3 shadow regions (and associated memory maps). Associated with each shadow region are a set of registers defining which channels and interrupt completion codes belong to that region. These registers are user-programmed per region to assign ownership of the DMA/QDMA channels to a region.

- DRAE*m* and DRAEH*m*: One register pair exists for each of the shadow regions. The number of bits in each register pair matches the number of DMA channels (64 DMA channels). These registers need to be programmed to assign ownership of DMA channels and interrupt (or TCC codes) to the respective region. Accesses to DMA and interrupt registers via the shadow region address view are filtered through the DRAE/DRAEH pair. A value of 1 in the corresponding DRAE(H) bit implies that the corresponding DMA/interrupt channel is accessible; a value of 0 in the corresponding DRAE(H) bit forces writes to be discarded and returns a value of 0 for reads.
- QRAE*n*: One register exists for every region. The number of bits in each register matches the number of QDMA channels (4 QDMA channels). These registers must be programmed to assign ownership of QDMA channels to the respective region. To enable a channel in a shadow region using shadow region 0 QEER, the respective bit in QRAE must be set or writing into QEESR will not have the desired effect.
- MPPA*n* and MPPAG: One register exists for every region. This register defines the privilege level, requestor, and types of accesses allowed to a region's memory-mapped registers.

It is typical for an application to have a unique assignment of QDMA/DMA channels (and, therefore, a given bit position) to a given region.

The use of shadow regions allows for restricted access to EDMA3 resources (DMA channels, QDMA channels, TCC, interrupts) by tasks in a system by setting or clearing bits in the DRAE/ORAE registers. If exclusive access to any given channel / TCC code is required for a region, then only that region's DRAE/ORAE should have the associated bit set.

Example 10-1. Resource Pool Division Across Two Regions

This example illustrates a judicious resource pool division across two regions, assuming region 0 must be allocated 16 DMA channels (0-15) and 1 QDMA channel (0) and 32 TCC codes (0-15 and 48-63). Region 1 needs to be allocated 16 DMA channels (16-32) and the remaining 7 QDMA channels (1-7) and TCC codes (16-47). DRAE should be equal to the OR of the bits that are required for the DMA channels and the TCC codes:

```
Region 0: DRAEH, DRAE = 0xFFFF0000, 0x0000FFFF QRAE = 0x0000001
Region 1: DRAEH, DRAE = 0x0000FFFF, 0xFFFF0000 QRAE = 0x00000FE
```

10.3.7.3 Region Interrupts

In addition to the EDMA3CC global completion interrupt, there is an additional completion interrupt line that is associated with every shadow region. Along with the interrupt enable register (IER), DRAE acts as a secondary interrupt enable for the respective shadow region interrupts. See Section 10.3.9 for more information.

10.3.8 Chaining EDMA3 Channels

The channel chaining capability for the EDMA3 allows the completion of an EDMA3 channel transfer to trigger another EDMA3 channel transfer. The purpose is to allow you the ability to chain several events through one event occurrence.

Chaining is different from linking (Section 10.3.3.7). The EDMA3 link feature reloads the current channel parameter set with the linked parameter set. The EDMA3 chaining feature does not modify or update any channel parameter set; it provides a synchronization event to the chained channel (see Section 10.3.4.1.3 for chain-triggered transfer requests).

Chaining is achieved at either final transfer completion or intermediate transfer completion, or both, of the current channel. Consider a channel *m* (DMA/QDMA) required to chain to channel *n*. Channel number *n* (0-63) needs to be programmed into the TCC bit of channel *m* channel options parameter (OPT) set.

- If final transfer completion chaining (TCCHEN = 1 in OPT) is enabled, the chain-triggered event occurs after the submission of the last transfer request of channel *m* is either submitted or completed (depending on early or normal completion).
- If intermediate transfer completion chaining (ITCCHEN = 1 in OPT) is enabled, the chain-triggered event occurs after every transfer request, except the last of channel *m* is either submitted or completed

(depending on early or normal completion).

- If both final and intermediate transfer completion chaining (TCCHEN = 1 and ITCCHEN = 1 in OPT) are enabled, then the chain-trigger event occurs after every transfer request is submitted or completed (depending on early or normal completion).

Table 10-12 illustrates the number of chain event triggers occurring in different synchronized scenarios. Consider channel 31 programmed with ACNT = 3, BCNT = 4, CCNT = 5, and TCC = 30.

Table 10-12. Chain Event Triggers

	(Number of chained event triggers on channel 30)	
Options	**A-Synchronized**	**AB-Synchronized**
TCCHEN = 1, ITCCHEN = 0	1 (Owing to the last TR)	1 (Owing to the last TR)
TCCHEN = 0, ITCCHEN = 1	19 (Owing to all but the last TR)	4 (Owing to all but the last TR)
TCCHEN = 1, ITCCHEN = 1	20 (Owing to a total of 20 TRs)	5 (Owing to a total of 5 TRs)

10.3.9 EDMA3 Interrupts

The EDMA3 interrupts are divided into 2 categories: transfer completion interrupts and error interrupts.

There are nine region interrupts, eight shadow regions and one global region. The transfer completion interrupts are listed in Table 10-13. The transfer completion interrupts and the error interrupts from the transfer controllers are all routed to the ARM interrupt controllers.

Table 10-13. EDMA3 Transfer Completion Interrupts

Name	**Description**
EDMA3CC_INT0	EDMA3CC Transfer Completion Interrupt Shadow Region 0
EDMA3CC_INT1	EDMA3CC Transfer Completion Interrupt Shadow Region 1
EDMA3CC_INT2	EDMA3CC Transfer Completion Interrupt Shadow Region 2
EDMA3CC_INT3	EDMA3CC Transfer Completion Interrupt Shadow Region 3
EDMA3CC_INT4	EDMA3CC Transfer Completion Interrupt Shadow Region 4
EDMA3CC_INT5	EDMA3CC Transfer Completion Interrupt Shadow Region 5
EDMA3CC_INT6	EDMA3CC Transfer Completion Interrupt Shadow Region 6
EDMA3CC_INT7	EDMA3CC Transfer Completion Interrupt Shadow Region 7

Table 10-14. EDMA3 Error Interrupts

Name	**Description**
EDMA3CC_ERRINT	EDMA3CC Error Interrupt
EDMA3CC_MPINT	EDMA3CC Memory Protection Interrupt
EDMA3TC0_ERRINT	TC0 Error Interrupt
EDMA3TC1_ERRINT	TC1 Error Interrupt
EDMA3TC2_ERRINT	TC2 Error Interrupt
EDMA3TC3_ERRINT	TC3 Error Interrupt

10.3.9.1 Transfer Completion Interrupts

The EDMA3CC is responsible for generating transfer completion interrupts to the CPUs (and other EDMA3 masters). The EDMA3 generates a single completion interrupt per shadow region, as well as one for the global region on behalf of all 64 channels. The various control registers and bit fields facilitate EDMA3 interrupt generation.

The software architecture should either use the global interrupt or the shadow interrupts, but not both.

The transfer completion code (TCC) value is directly mapped to the bits of the interrupt pending register (IPR/IPRH). For example, if TCC = 10 0001b, IPRH[1] is set after transfer completion, and results in interrupt generation to the CPUs if the completion interrupt is enabled for the CPU. See Section 10.3.9.1.1 for details on enabling EDMA3 transfer completion interrupts.

When a completion code is returned (as a result of early or normal completions), the corresponding bit in IPR/IPRH is set if transfer completion interrupt (final/intermediate) is enabled in the channel options parameter (OPT) for a PaRAM set associated with the transfer.

Table 10-15. Transfer Complete Code (TCC) to EDMA3CC Interrupt Mapping

TCC Bits in OPT (TCINTEN/ITCINTEN = 1)	IPR Bit Set	TCC Bits in OPT (TCINTEN/ITCINTEN = 1)	IPRH Bit Set[1]
0	IPR0	20h	IPR32/IPRH0
1	IPR1	21h	IPR33/IPRH1
2h	IPR2	22h	IPR34/IPRH2
3h	IPR3	23h	IPR35/IPRH3
4h	IPR4	24h	IPR36/IPRH4
...	...	...	...
1Eh	IPR30	3Eh	IPR62/IPRH30
1Fh	IPR31	3Fh	IPR63/IPRH31

[1] Bit fields IPR[32-63] correspond to bits 0 to 31 in IPRH, respectively.

You can program the transfer completion code (TCC) to any value for a DMA/QDMA channel. A direct relation between the channel number and the transfer completion code value does not need to exist. This allows multiple channels having the same transfer completion code value to cause a CPU to execute the same interrupt service routine (ISR) for different channels.

If the channel is used in the context of a shadow region and you intend for the shadow region interrupt to be asserted, then ensure that the bit corresponding to the TCC code is enabled in IER/IERH and in the corresponding shadow region's DMA region access registers (DRAE/DRAEH).

You can enable Interrupt generation at either final transfer completion or intermediate transfer completion, or both. Consider channel *m* as an example.

- If the final transfer interrupt (TCCINT = 1 in OPT) is enabled, the interrupt occurs after the last transfer request of channel *m* is either submitted or completed (depending on early or normal completion).
- If the intermediate transfer interrupt (ITCCINT = 1 in OPT) is enabled, the interrupt occurs after every transfer request, except the last TR of channel *m* is either submitted or completed (depending on early or normal completion).
- If both final and intermediate transfer completion interrupts (TCCINT = 1, and ITCCINT = 1 in OPT) are enabled, then the interrupt occurs after every transfer request is submitted or completed (depending on early or normal completion).

Table 10-16 shows the number of interrupts that occur in different synchronized scenarios. Consider channel 31, programmed with ACNT = 3, BCNT = 4, CCNT = 5, and TCC = 30.

Table 10-16. Number of Interrupts

Options	A-Synchronized	AB-Synchronized
TCINTEN = 1, ITCINTEN = 0	1 (Last TR)	1 (Last TR)
TCINTEN = 0, ITCINTEN = 1	19 (All but the last TR)	4 (All but the last TR)
TCINTEN = 1, ITCINTEN = 1	20 (All TRs)	5 (All TRs)

10.3.9.1.1 Enabling Transfer Completion Interrupts

For the EDMA3 channel controller to assert a transfer completion to the external environment, the interrupts must be enabled in the EDMA3CC. This is in addition to setting up the TCINTEN and ITCINTEN bits in OPT of the associated PaRAM set.

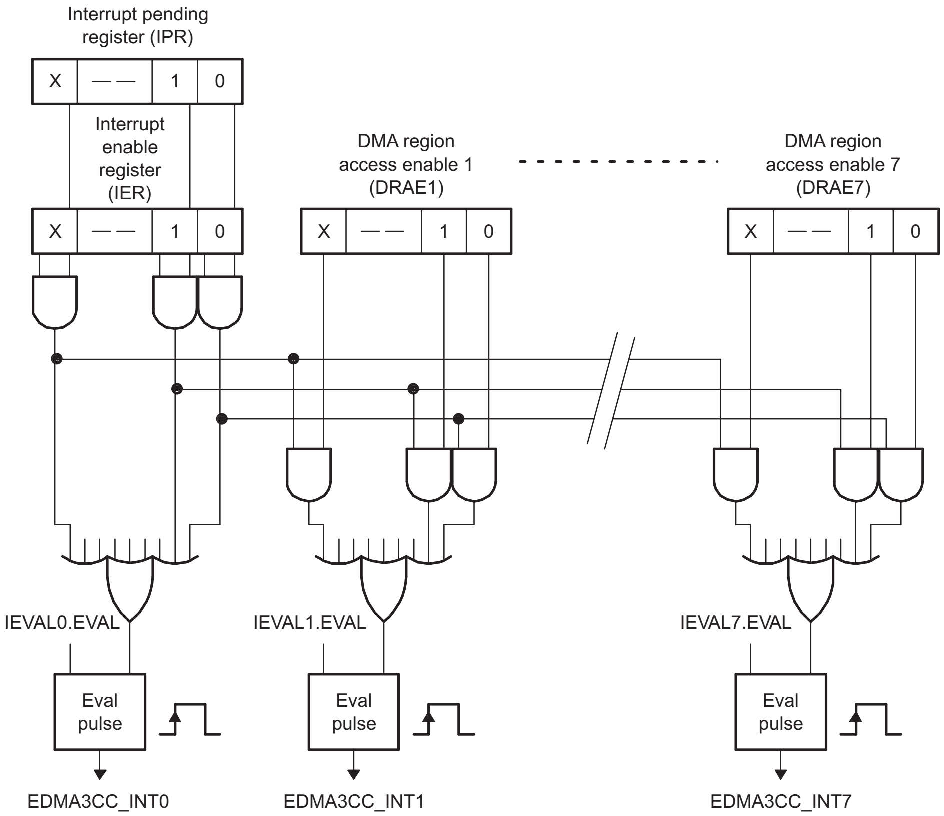
The EDMA3 channel controller has interrupt enable registers (IER/IERH) and each bit location in IER/IERH serves as a primary enable for the corresponding interrupt pending registers (IPR/IPRH).

All of the interrupt registers (IER, IESR, IECR, and IPR) are either manipulated from the global DMA channel region, or by the DMA channel shadow regions. The shadow regions provide a view to the same set of physical registers that are in the global region.

The EDMA3 channel controller has a hierarchical completion interrupt scheme that uses a single set of interrupt pending registers (IPR/IPRH) and single set of interrupt enable registers (IER/IERH). The programmable DMA region access enable registers (DRAE/DRAEH) provides a second level of interrupt masking. The global region interrupt output is gated based on the enable mask that is provided by IER/IERH. see Figure 10-16

The region interrupt outputs are gated by IER and the specific DRAE/DRAEH associated with the region. See Figure 10-16.

Figure 10-16. Interrupt Diagram



For the EDMA3CC to generate the transfer completion interrupts that are associated with each shadow region, the following conditions must be true:

- EDMA3CC_INT0: (IPR.E0 & IER.E0 & DRAE0.E0) | (IPR.E1 & IER.E1 & DRAE0.E1) | …|(IPRH.E63 & IERH.E63 & DRAHE0.E63)
- EDMA3CC_INT1: (IPR.E0 & IER.E0 & DRAE1.E0) | (IPR.E1 & IER.E1 & DRAE1.E1) | …| (IPRH.E63 & IERH.E63 & DRAHE1.E63)
- EDMA3CC_INT2 : (IPR.E0 & IER.E0 & DRAE2.E0) | (IPR.E1 & IER.E1 & DRAE2.E1) | …|(IPRH.E63 & IERH.E63 & DRAHE2.E63)....

- Up to EDMA3CC_INT7 : (IPR.E0 & IER.E0 & DRAE7.E0) | (IPR.E1 & IER.E1 & DRAE7.E1) | …|(IPRH.E63 & IERH.E63 & DRAEH7.E63)

NOTE: The DRAE/DRAEH for all regions are expected to be set up at system initialization and to remain static for an extended period of time. The interrupt enable registers should be used for dynamic enable/disable of individual interrupts.

Because there is no relation between the TCC value and the DMA/QDMA channel, it is possible, for example, for DMA channel 0 to have the OPT.TCC = 63 in its associated PaRAM set. This would mean that if a transfer completion interrupt is enabled (OPT.TCINTEN or OPT.ITCINTEN is set), then based on the TCC value, IPRH.E63 is set up on completion. For proper channel operations and interrupt generation using the shadow region map, you must program the DRAE/DRAEH that is associated with the shadow region to have read/write access to both bit 0 (corresponding to channel 0) and bit 63 (corresponding to IPRH bit that is set upon completion).

10.3.9.1.2 Clearing Transfer Completion Interrupts

Transfer completion interrupts that are latched to the interrupt pending registers (IPR/IPRH) are cleared by writing a 1 to the corresponding bit in the interrupt pending clear register (ICR/ICRH). For example, a write of 1 to ICR.E0 clears a pending interrupt in IPR.E0.

If an incoming transfer completion code (TCC) gets latched to a bit in IPR/IPRH, then additional bits that get set due to a subsequent transfer completion will not result in asserting the EDMA3CC completion interrupt. In order for the completion interrupt to be pulsed, the required transition is from a state where no enabled interrupts are set to a state where at least one enabled interrupt is set.

10.3.9.2 EDMA3 Interrupt Servicing

Upon completion of a transfer (early or normal completion), the EDMA3 channel controller sets the appropriate bit in the interrupt pending registers (IPR/IPRH), as the transfer completion codes specify. If the completion interrupts are appropriately enabled, then the CPU enters the interrupt service routine (ISR) when the completion interrupt is asserted.

After servicing the interrupt, the ISR should clear the corresponding bit in IPR/IPRH, thereby enabling recognition of future interrupts. The EDMA3CC will only assert additional completion interrupts when all IPR/IPRH bits clear.

When one interrupt is serviced many other transfer completions may result in additional bits being set in IPR/IPRH, thereby resulting in additional interrupts. Each of the bits in IPR/IPRH may need different types of service; therefore, the ISR may check all pending interrupts and continue until all of the posted interrupts are serviced appropriately.

Examples of pseudo code for a CPU interrupt service routine for an EDMA3CC completion interrupt are shown in Example 10-2 and Example 10-3.

The ISR routine in Example 10-2 is more exhaustive and incurs a higher latency.

Example 10-2. Interrupt Servicing

The pseudo code:

1. Reads the interrupt pending register (IPR/IPRH).
2. Performs the operations needed.
3. Writes to the interrupt pending clear register (ICR/ICRH) to clear the corresponding IPR/IPRH bits.
4. Reads IPR/IPRH again:
   a. If IPR/IPRH is not equal to 0, repeat from step 2 (implies occurrence of new event between step 2 to step 4).
   b. If IPR/IPRH is equal to 0, this should assure you that all of the enabled interrupts are inactive.

Example 10-2. Interrupt Servicing (continued)

NOTE: An event may occur during step 4 while the IPR/IPRH bits are read as 0 and the application is still in the interrupt service routine. If this happens, a new interrupt is recorded in the device interrupt controller and a new interrupt generates as soon as the application exits in the interrupt service routine.

Example 10-3 is less rigorous, with less burden on the software in polling for set interrupt bits, but can occasionally cause a race condition as mentioned above.

Example 10-3. Interrupt Servicing

If you want to leave any enabled and pending (possibly lower priority) interrupts; you must force the interrupt logic to reassert the interrupt pulse by setting the EVAL bit in the interrupt evaluation register (IEVAL).

The pseudo code is as follows:

1. Enters ISR.
2. Reads IPR/IPRH.
3. For the condition that is set in IPR/IPRH that you want to service, do the following:
   a. Service interrupt as the application requires.
   b. Clear the bit for serviced conditions (others may still be set, and other transfers may have resulted in returning the TCC to EDMA3CC after step 2).
4. Reads IPR/IPRH prior to exiting the ISR:
   a. If IPR/IPRH is equal to 0, then exit the ISR.
   b. If IPR/IPRH is not equal to 0, then set IEVAL so that upon exit of ISR, a new interrupt triggers if any enabled interrupts are still pending.

10.3.9.3 Interrupt Evaluation Operations

The EDMA3CC has interrupt evaluate registers (IEVAL) that exist in the global region and in each shadow region. The registers in the shadow region are the only registers in the DMA channel shadow region memory map that are not affected by the settings for the DMA region access enable registers (DRAE/DRAEH). Writing a 1 to the EVAL bit in the registers that are associated with a particular shadow region results in pulsing the associated region interrupt (global or shadow), if any enabled interrupt (via IER/IERH) is still pending (IPR/IPRH). This register assures that the CPU does not miss the interrupts (or the EDMA3 master associated with the shadow region) if the software architecture chooses not to use all interrupts. See Example 10-3 for the use of IEVAL in the EDMA3 interrupt service routine (ISR).

Similarly, an error evaluation register (EEVAL) exists in the global region. Writing a 1 to the EVAL bit in EEVAL causes the pulsing of the error interrupt if any pending errors are in EMR/EMRH, QEMR, or CCERR. The EVAL bit must be written with 1 to clear interrupts to the INTC, even when all error interrupt registers are cleared. See Section 10.3.9.4, *Error Interrupts*, for additional information regarding error interrupts.

NOTE: While using IEVAL for shadow region completion interrupts, you should make sure that the IEVAL operated upon is from that particular shadow region memory map.

10.3.9.4 Error Interrupts

The EDMA3CC error registers provide the capability to differentiate error conditions (event missed, threshold exceed, etc.). Additionally, setting the error bits in these registers results in asserting the EDMA3CC error interrupt. If the EDMA3CC error interrupt is enabled in the device interrupt controllers, then it allows the CPUs to handle the error conditions.

The EDMA3CC has a single error interrupt (EDMA3CC_ERRINT) that is asserted for all EDMA3CC error conditions. There are four conditions that cause the error interrupt to pulse:

- DMA missed events: for all 64 DMA channels. DMA missed events are latched in the event missed registers (EMR/EMRH).
- QDMA missed events: for all 8 QDMA channels. QDMA missed events are latched in the QDMA event missed register (QEMR).
- Threshold exceed: for all event queues. These are latched in EDMA3CC error register (CCERR).
- TCC error: for outstanding transfer requests that are expected to return completion code (TCCHEN or TCINTEN bit in OPT is set to 1) exceeding the maximum limit of 63. This is also latched in the EDMA3CC error register (CCERR).

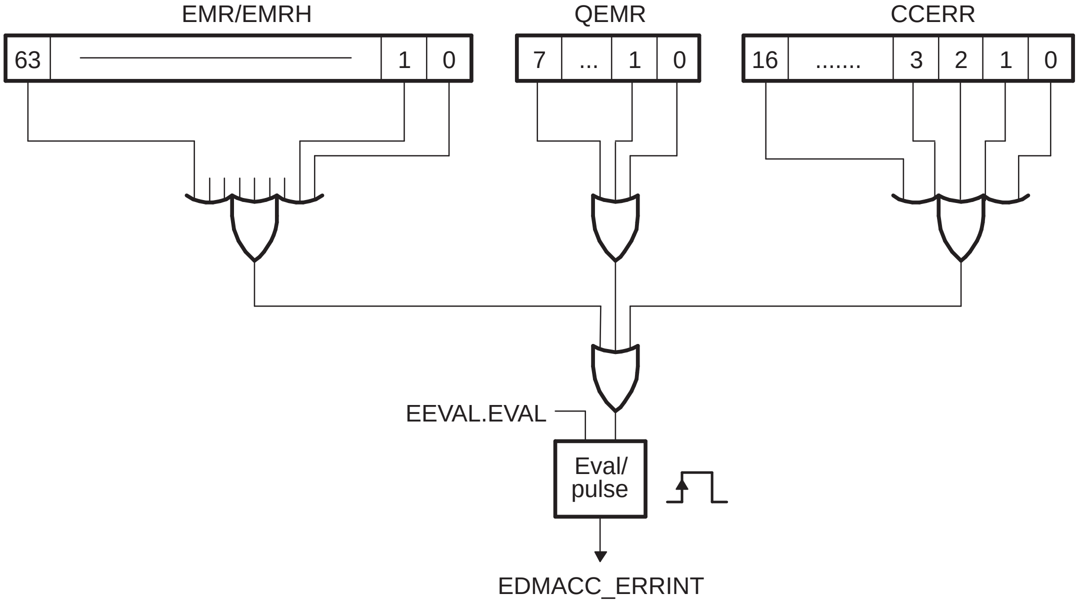
Figure 10-17 illustrates the EDMA3CC error interrupt generation operation.

If any of the bits are set in the error registers due to any error condition, the EDMA3CC_ERRINT is always asserted, as there are no enables for masking these error events. Similar to transfer completion interrupts (EDMA3CC_INT), the error interrupt also only pulses when the error interrupt condition transitions from no errors being set to at least one error being set. If additional error events are latched prior to the original error bits clearing, the EDMA3CC does not generate additional interrupt pulses.

To reduce the burden on the software, there is an error evaluate register (EEVAL) that allows re-evaluation of pending set error events/bits, similar to the interrupt evaluate register (IEVAL). You can use this so that the CPU does not miss any error events. You must write a 1 to the EEVAL.EVAL bit to clear interrupts to the INTC after all error registers have been cleared.

NOTE: It is good practice to enable the error interrupt in the device interrupt controller and to associate an interrupt service routine with it to address the various error conditions appropriately. Doing so puts less burden on the software (polling for error status); additionally, it provides a good debug mechanism for unexpected error conditions.

Figure 10-17. Error Interrupt Operation



10.3.10 Memory Protection

The EDMA3 channel controller supports two kinds of memory protection: active and proxy.

10.3.10.1 Active Memory Protection

Active memory protection is a feature that allows or prevents read and write accesses (by any EDMA3 programmer) to the EDMA3CC registers (based on permission characteristics that you program). Active memory protection is achieved by a set of memory protection permissions attribute (MPPA) registers.

The EDMA3CC register map is divided into three categories:

- a global region.
- a global channel region.
- eight shadow regions.

Each shadow region consists of the respective shadow region registers and the associated PaRAM. For more detailed information regarding the contents of a shadow region, refer to section Table 10-11.

Each of the eight shadow regions has an associated MPPA register (MPPA*n*) that defines the specific requestors and types of requests that are allowed to the regions resources.

The global channel region is also protected with a memory-mapped register (MPPAG). The MPPAG applies to the global region and to the global channel region, except the other MPPA registers themselves. For more detailed information on the list of the registers in each region, refer to the register memory-map in Table 10-18.

The MPPA*n* have a certain set of access rules. For more information, see the bit field descriptions of MPPA*n*.

Table 10-17 shows the accesses that are allowed or not allowed to the MPPAG and MPPA*n*. The active memory protection uses the PRIV and PRIVID attributes of the EDMA programmer. The PRIV is the privilege level (i.e., user vs. supervisor). The PRIVID refers to a privilege ID with a number that is associated with an EDMA3 programmer. See the device-specific data manual for the PRIVIDs that are associated with potential EDMA3 programmers.

Table 10-17. Allowed Accesses

Access	Supervisor	User
Read	Yes	Yes
Write	Yes	No

Table 10-18 describes the MPPA register mapping for the shadow regions (which includes shadow region registers and PaRAM addresses).

The region-based MPPA registers are used to protect accesses to the DMA shadow regions and the associated region PaRAM. Because there are eight regions, there are eight MPPA region registers (MPPA[0-7]).

Table 10-18. MPPA Registers to Region Assignment

Register	Registers Protect	Address Range	PaRAM Protect[1]	Address Range
MPPAG	Global Range	0000h-1FFCh	N/A	N/A
MPPA0	DMA Shadow 0	2000h-21FCh	1st octant	4000h-47FCh
MPPA1	DMA Shadow 1	2200h-23FCh	2nd octant	4800h-4FFCh
MPPA2	DMA Shadow 2	2400h-25FCh	3rd octant	5000h-57FCh
MPPA3	DMA Shadow 3	2600h-27FCh	4th octant	5800h-5FFCh
MPPA4	DMA Shadow 4	2800h-29FCh	5th octant	6000h-67FCh
MPPA5	DMA Shadow 5	2A00h-2BFCh	6th octant	6800h-6FFCh
MPPA6	DMA Shadow 6	2C00h-2DFCh	7th octant	7000h-77FCh
MPPA7	DMA Shadow 7	2E00h-2FFCh	8th octant	7800h-7FFCh

[1] The PARAM region is divided into 8 regions referred to as an octant.

Example Access denied.

Write access to shadow region 7's event enable set register (EESR):

1. The original value of the event enable register (EER) at address offset 0x1020 is 0x0.
2. The MPPA[7] is set to prevent user level accesses (UW = 0, UR = 0), but it allows supervisor level accesses (SW = 1, SR = 1) with a privilege ID of 0. (AID0 = 1).
3. An EDMA3 programmer with a privilege ID of 0 attempts to perform a user-level write of a value of 0xFF00FF00 to shadow region 7's event enable set register (EESR) at address offset 0x2E30. Note that the EER is a read-only register and the only way that you can write to it is by writing to the EESR. Also remember that there is only one physical register for EER, EESR, etc. and that the shadow regions only provide to the same physical set.
4. Since the MPPA[7] has UW = 0, though the privilege ID of the write access is set to 0, the access is not allowed and the EER is not written to.

Table 10-19. Example Access Denied

Register	Value	Description
EER (offset 0x1020)	0x0000 0000	Value in EER to begin with.
EESR (offset 0x2E30)	0xFF00 FF00 ↓	Value attempted to be written to shadow region 7's EESR. This is done by an EDMA3 programmer with a privilege level of User and Privilege ID of 0.
MPPA[7] (offset 0x082C)	0x0000 04B0	Memory Protection Filter AID0 = 1, UW = 0, UR = 0, SW = 1, SR = 1.
	X	Access Denied
EER (offset 0x1020)	0x0000 0000	Final value of EER

Example Access Allowed

Write access to shadow region 7's event enable set register (EESR):

1. The original value of the event enable register (EER) at address offset 0x1020 is 0x0.
2. The MPPA[7] is set to allow user-level accesses (UW = 1, UR = 1) and supervisor-level accesses (SW = 1, SR = 1) with a privilege ID of 0. (AID0 = 1).
3. An EDMA3 programmer with a privilege ID of 0, attempts to perform a user-level write of a value of 0xABCD0123 to shadow region 7's event enable set register (EESR) at address offset 0x2E30. Note that the EER is a read-only register and the only way that you can write to it is by writing to the EESR. Also remember that there is only one physical register for EER, EESR, etc. and that the shadow regions only provide to the same physical set.
4. Since the MPPA[7] has UW = 1 and AID0 = 1, the user-level write access is allowed.
5. Remember that accesses to shadow region registers are masked by their respective DRAE register. In this example, the DRAE[7] is set of 0x9FF00FC2.
6. The value finally written to EER is 0x8BC00102.

Table 10-20. Example Access Allowed

Register	Value	Description
EER (offset 0x1020)	0x0000 0000	Value in EER to begin with.
EESR (offset 0x2E30)	0xFF00 FF00	Value attempted to be written to shadow region 7's EESR. This is done by an EDMA3 programmer with a privilege level of User and Privilege ID of 0.
MPPA[7] (offset 0x082C)	0x0000 04B3	Memory Protection Filter AID = 1, UW = 1, UR = 1, SW = 1, SR = 1.
	√ ↓	Access allowed.
DRAE[7] (offset 0x0378)	0x9FF0 0FC2 ↓	DMA Region Access Enable Filter
EESR (offset 0x2E30)	0x8BC0 0102 ↓	Value written to shadow region 7's EESR. This is done by an EDMA3 programmer with a privilege level of User and a Privilege ID of 0.
EER (offset 0x1020)	↓ 0xBC0 0102	Final value of EER.

10.3.10.2 Proxy Memory Protection

Proxy memory protection allows an EDMA3 transfer programmed by a given EDMA3 programmer to have its permissions travel with the transfer through the EDMA3TC. The permissions travel along with the read transactions to the source and the write transactions to the destination endpoints. The PRIV bit and PRIVID bit in the channel options parameter (OPT) is set with the EDMA3 programmer's PRIV value and PRIVID values, respectively, when any part of the PaRAM set is written.

The PRIV is the privilege level (i.e., user vs. supervisor). The PRIVID refers to a privilege ID with a number that is associated with an EDMA3 programmer.

See the data manual for the PRIVIDs that are associated with potential EDMA3 programmers.

These options are part of the TR that are submitted to the transfer controller. The transfer controller uses the above values on their respective read and write command bus so that the target endpoints can perform memory protection checks based on these values.

Consider a parameter set that is programmed by a CPU in user privilege level for a simple transfer with the source buffer on an L2 page and the destination buffer on an L1D page. The PRIV is 0 for user-level and the CPU has a PRIVID of 0.

The PaRAM set is shown in Figure 10-18.

Figure 10-18. PaRAM Set Content for Proxy Memory Protection Example

(a) EDMA3 Parameters

Parameter Contents		Parameter	
0010 0007h		Channel Options Parameter (OPT)	
009F 0000h		Channel Source Address (SRC)	
0001h	0004h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
00F0 7800h		Channel Destination Address (DST)	
0001h	0001h	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0000h	FFFFh	BCNT Reload (BCNTRLD)	Link Address (LINK)
0001h	1000h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

Figure 10-19. Channel Options Parameter (OPT) Example

(b) Channel Options Parameter (OPT) Content

31	30	29 28	27 24	23	22	21	20	19 18	17 16
0	0	00	0000	0	0	0	1	00	00
PRIV	Rsvd	Rsvd	PRIVID	ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	Reserved	TCC

15 12	11	10 8	7 4	3	2	1	0
0000	0	000	0000	0	1	1	1
TCC	TCCMOD	FWID	Reserved	STATIC	SYNCDIM	DAM	SAM

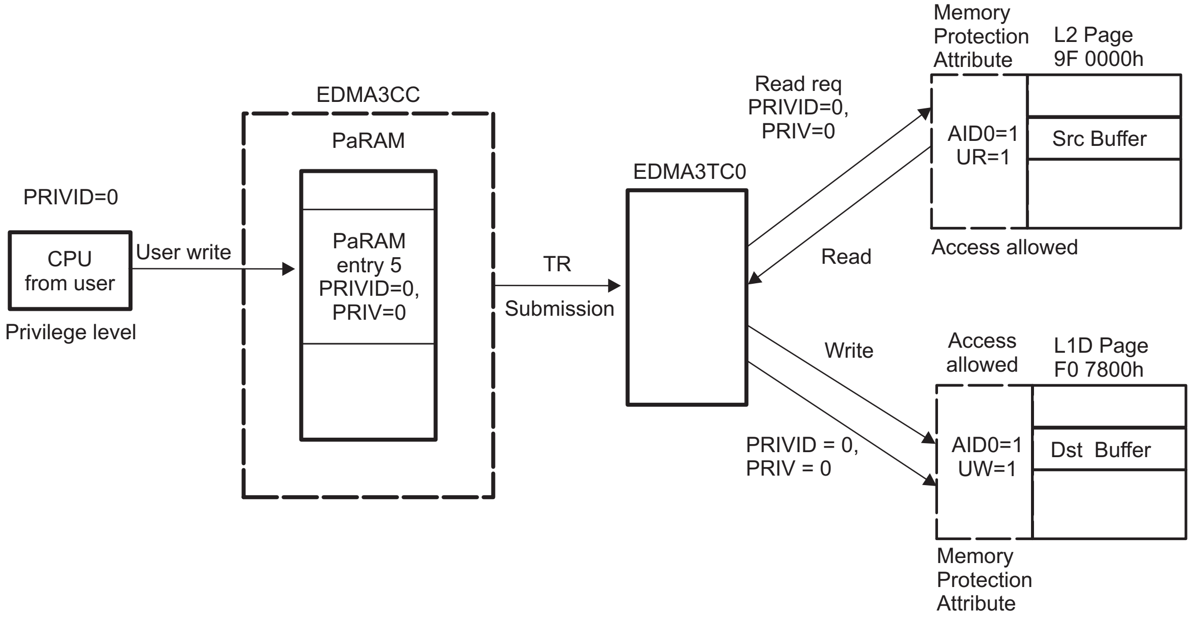
The PRIV and PRIVID information travels along with the read and write requests that are issued to the source and destination memories.

For example, if the access attributes that are associated with the L2 page with the source buffer only allow supervisor read, write accesses (SR,SW), the user-level read request above is refused. Similarly, if the access attributes that are associated with the L1D page with the destination buffer only allow supervisor read and write accesses (SR, SW), the user-level write request above is refused. For the transfer to succeed, the source and destination pages should have user-read and user-write permissions, respectively, along with allowing accesses from a PRIVID 0.

Because the programmer's privilege level and privilege identification travel with the read and write requests, EDMA3 acts as a proxy.

Figure 10-20 illustrates the propagation of PRIV and PRIVID at the boundaries of all the interacting entities (CPU, EDMA3CC, EDMA3TC, and slave memories).

Figure 10-20. Proxy Memory Protection Example



10.3.11 Event Queues

Event queues are a part of the EDMA3 channel controller. Event queues form the interface between the event detection logic in the EDMA3CC and the transfer request (TR) submission logic of the EDMA3CC. Each queue is 16 entries deep; thus, each event queue can queue a maximum of 16 events. If there are more than 16 events, then the events that cannot find a place in the event queue remain set in the associated event register and the CPU does not stall.

There are four event queues for the device: Queue0, Queue1, Queue2, and Queue3. Events in Queue0 result in submission of its associated transfer requests (TRs) to TC0. Similarly, transfer requests that are associated with events in Queue3 are submitted to TC3.

An event that wins prioritization against other DMA and/or QDMA pending events is placed at the tail of the appropriate event queue. Each event queue is serviced in FIFO order. Once the event reaches the head of its queue and the corresponding transfer controller is ready to receive another TR, the event is de-queued and the PaRAM set corresponding to the de-queued event is processed and submitted as a transfer request packet (TRP) to the associated EDMA3 transfer controller.

Queue0 has highest priority and Queue3 has the lowest priority, if Queue0 and Queue1 both have at least one event entry and if both TC0 and TC1 can accept transfer requests, then the event in Queue0 is de-queued first and its associated PaRAM set is processed and submitted as a transfer request (TR) to TC0.

See Section 10.3.11.4 for system-level performance considerations. All of the event entries in all of the event queues are software readable (not writeable) by accessing the event entry registers (Q0E0, Q0E1,…Q1E15, and so on). Each event entry register characterizes the queued event in terms of the type of event (manual, event, chained or auto-triggered) and the event number. For more information, see the bit fields in the queue event entry (QxEy) registers.

10.3.11.1 DMA/QDMA Channel to Event Queue Mapping

Each of the 64 DMA channels and eight QDMA channels are programmed independently to map to a specific queue, using the DMA queue number register (DMAQNUM) and the QDMA queue number register (QDMANUM). The mapping of DMA/QDMA channels is critical to achieving the desired performance level for the EDMA and most importantly, in meeting real-time deadlines. See Section 10.3.11.4.

NOTE: If an event is ready to be queued and both the event queue and the EDMA3 transfer controller that is associated to the event queue are empty, then the event bypasses the event queue, and moves the PaRAM processing logic, and eventually to the transfer request submission logic for submission to the EDMA3TC. In this case, the event is not logged in the event queue status registers.

10.3.11.2 Queue RAM Debug Visibility

There are four event queues and each queue has 16 entries. These 16 entries are managed in a circular FIFO manner. There is a queue status register (QSTAT) associated with each queue. These along with all of the 16 entries per queue can be read via registers QSTAT*n* and QxEy, respectively.

These registers provide user visibility and may be helpful while debugging real-time issues (typically post-mortem), involving multiple events and event sources. The event queue entry register (Q*x*E*y*) uniquely identifies the specific event type (event-triggered, manually-triggered, chain-triggered, and QDMA events) along with the event number (for all DMA/QDMA event channels) that are in the queue or have been de-queued (passed through the queue).

Each of the 16 entries in the event queue are read using the EDMA3CC memory-mapped register. By reading the event queue, you see the history of the last 16 TRs that have been processed by the EDMA3 on a given queue. This provides user/software visibility and is helpful for debugging real-time issues (typically post-mortem), involving multiple events and event sources.

The queue status register (QSTAT*n*) includes fields for the start pointer (STRTPTR) which provides the offset to the head entry of an event. It also includes a field called NUMVAL that provides the total number of valid entries residing in the event queue at a given instance of time. The STRTPTR may be used to index appropriately into the 16 event entries. NUMVAL number of entries starting from STRTPTR are indicative of events still queued in the respective queue. The remaining entry may be read to determine what's already de-queued and submitted to the associated transfer controller.

10.3.11.3 Queue Resource Tracking

The EDMA3CC event queue includes watermarking/threshold logic that allows you to keep track of maximum usage of all event queues. This is useful for debugging real-time deadline violations that may result from head-of-line blocking on a given EDMA3 event queue.

You can program the maximum number of events that can queue up in an event queue by programming the threshold value (between 0 to 15) in the queue watermark threshold A register (QWMTHRA). The maximum queue usage is recorded actively in the watermark (WM) field of the queue status register (QSTAT*n*) that keeps getting updated based on a comparison of number of valid entries, which is also visible in the NUMVAL bit in QSTAT*n* and the maximum number of entries (WM bit in QSTAT*n*).

If the queue usage is exceeded, this status is visible in the EDMA3CC registers: the QTHRXCD*n* bit in the channel controller error register (CCERR) and the THRXCD bit in QSTAT*n*, where *n* stands for the event queue number. Any bits that are set in CCERR also generate an EDMA3CC error interrupt.

10.3.11.4 Performance Considerations

The main system bus infrastructure (L3) arbitrates bus requests from all of the masters (TCs, CPU(S), and other bus masters) to the shared slave resources (peripherals and memories).

The priorities of transfer requests (read and write commands) from the EDMA3 transfer controllers with respect to other masters within the system crossbar are programmed using the queue priority register (QUEPRI). QUEPRI programs the priority of the event queues (or indirectly, TC0-TC3, because Queue*N* transfer requests are submitted to TC*N*).

Therefore, the priority of unloading queues has a secondary affect compared to the priority of the transfers as they are executed by the EDMA3TC (dictated by the priority set using QUEPRI).

10.3.12 EDMA3 Transfer Controller (EDMA3TC)

The EDMA3 channel controller is the user-interface of the EDMA3 and the EDMA3 transfer controller (EDMA3TC) is the data movement engine of the EDMA3. The EDMA3CC submits transfer requests (TR) to the EDMA3TC and the EDMA3TC performs the data transfers dictated by the TR; thus, the EDMA3TC is a slave to the EDMA3CC.

10.3.12.1 Architecture Details

10.3.12.1.1 Command Fragmentation

The TC read and write controllers in conjunction with the source and destination register sets are responsible for issuing optimally-sized reads and writes to the slave endpoints. An optimally-sized command is defined by the transfer controller default burst size (DBS), which is defined in Section 10.3.12.5.

The EDMA3TC attempts to issue the largest possible command size as limited by the DBS value or the ACNT/BCNT value of the TR. EDMA3TC obeys the following rules:

- The read/write controllers always issue commands less than or equal to the DBS value.
- The first command of a 1D transfer command always aligns the address of subsequent commands to the DBS value.

Table 10-21 lists the TR segmentation rules that are followed by the EDMA3TC. In summary, if the ACNT value is larger than the DBS value, then the EDMA3TC breaks the ACNT array into DBS-sized commands to the source/destination addresses. Each BCNT number of arrays are then serviced in succession.

For BCNT arrays of ACNT bytes (that is, a 2D transfer), if the ACNT value is less than or equal to the DBS value, then the TR may be optimized into a 1D-transfer in order to maximize efficiency. The optimization takes place if the EDMA3TC recognizes that the 2D-transfer is organized as a single dimension (ACNT == BIDX) and the ACNT value is a power of 2.

Table 10-21 lists conditions in which the optimizations are performed.

Table 10-21. Read/Write Command Optimization Rules

ACNT ≤ DBS	ACNT is power of 2	BIDX = ACNT	BCNT ≤ 1023	SAM/DAM = Increment	Description
Yes	Yes	Yes	Yes	Yes	Optimized
No	x	x	x	x	Not Optimized
x	No	x	x	x	Not Optimized
x	x	No	x	x	Not Optimized
x	x	x	No	x	Not Optimized
x	x	x	x	No	Not Optimized

10.3.12.1.2 TR Pipelining

TR pipelining refers to the ability of the source active set to proceed ahead of the destination active set. Essentially, the reads for a given TR may already be in progress while the writes of a previous TR may not have completed.

The number of outstanding TRs is limited by the number of destination FIFO register entries.

TR pipelining is useful for maintaining throughput on back-to-back small TRs. It minimizes the startup overhead because reads start in the background of a previous TR writes.

Example 10-4. Command Fragmentation (DBS = 64)

The pseudo code:

1. ACNT = 8, BCNT = 8, SRCBIDX = 8, DSTBIDX = 10, SRCADDR = 64, DSTADDR = 191

   Read Controller: This is optimized from a 2D-transfer to a 1D-transfer such that the read side is equivalent to ACNT = 64, BCNT = 1.

   Cmd0 = 64 byte

   Write Controller: Because DSTBIDX != ACNT, it is not optimized.

   Cmd0 = 8 byte, Cmd1 = 8 byte, Cmd2 = 8 byte, Cmd3 = 8 byte, Cmd4 = 8 byte, Cmd5 = 8 byte, Cmd6 = 8 byte, Cmd7 = 8 byte.
2. ACNT=128, BCNT = 1,SRCADDR = 63, DSTADDR = 513

   Read Controller: Read address is not aligned.

   Cmd0 = 1 byte, (now the SRCADDR is aligned to 64 for the next command)

   Cmd1 = 64 bytes

   Cmd2 = 63 bytes

   Write Controller: The write address is also not aligned.

   Cmd0 = 63 bytes, (now the DSTADDR is aligned to 64 for the next command)

   Cmd1 = 64 bytes

   Cmd2 = 1 byte

10.3.12.1.3 Performance Tuning

By default, reads are as issued as fast as possible. In some cases, the reads issued by the EDMA3TC could fill the available command buffering for a slave, delaying other (potentially higher priority) masters from successfully submitting commands to that slave. The rate at which read commands are issued by the EDMA3TC is controlled by the RDRATE register. The RDRATE register defines the number of cycles that the EDMA3TC read controller waits before issuing subsequent commands for a given TR, thus minimizing the chance of the EDMA3TC consuming all available slave resources. The RDRATE value should be set to a relatively small value if the transfer controller is targeted for high priority transfers and to a higher value if the transfer controller is targeted for low priority transfers.

In contrast, the Write Interface does not have any performance turning knobs because writes always have an interval between commands as write commands are submitted along with the associated write data.

10.3.12.2 Memory Protection

The transfer controller plays an important role in handling proxy memory protection. There are two access properties associated with a transfer: for instance, the privilege id (system-wide identification assigned to a master) of the master initiating the transfer, and the privilege level (user versus supervisor) used to program the transfer. This information is maintained in the PaRAM set when it is programmed in the channel controller. When a TR is submitted to the transfer controller, this information is made available to the EDMA3TC and used by the EDMA3TC while issuing read and write commands. The read or write commands have the same privilege identification, and privilege level as that programmed in the EDMA3 transfer in the channel controller.

10.3.12.3 Error Generation

Errors are generated if enabled under three conditions:

- EDMA3TC detection of an error signaled by the source or destination address.
- Attempt to read or write to an invalid address in the configuration memory map.
- Detection of a constant addressing mode TR violating the constant addressing mode transfer rules (the source/destination addresses and source/destination indexes must be aligned to 32 bytes).

Either or all error types may be disabled. If an error bit is set and enabled, the error interrupt for the concerned transfer controller is pulsed.

10.3.12.4 Debug Features

The DMA program register set, DMA source active register set, and the destination FIFO register set are used to derive a brief history of TRs serviced through the transfer controller.

Additionally, the EDMA3TC status register (TCSTAT) has dedicated bit fields to indicate the ongoing activity within different parts of the transfer controller:

- The SRCACTV bit indicates whether the source active set is active.
- The DSTACTV bit indicates the number of TRs resident in the destination register active set at a given instance.
- The PROGBUSY bit indicates whether a valid TR is present in the DMA program set.

If the TRs are in progression, caution must be used and you must realize that there is a chance that the values read from the EDMA3TC status registers will be inconsistent since the EDMA3TC may change the values of these registers due to ongoing activities.

It is recommended that you ensure no additional submission of TRs to the EDMA3TC in order to facilitate ease of debug.

10.3.12.4.1 Destination FIFO Register Pointer

The destination FIFO register pointer is implemented as a circular buffer with the start pointer being DFSTRTPTR and a buffer depth of usually 2 or 4. The EDMA3TC maintains two important status details in TCSTAT that may be used during advanced debugging, if necessary. The DFSTRTPTR is a start pointer, that is, the index to the head of the destination FIFO register. The DSTACTV is a counter for the number of valid (occupied) entries. These registers may be used to get a brief history of transfers.

Examples of some register field values and their interpretation:

- DFSTRTPTR = 0 and DSTACTV = 0 implies that no TRs are stored in the destination FIFO register.
- DFSTRTPTR = 1 and DSTACTV = 2h implies that two TRs are present. The first pending TR is read from the destination FIFO register entry 1 and the second pending TR is read from the destination FIFO register entry 2.
- DFSTRTPTR = 3h and DSTACTV = 2h implies that two TRs are present. The first pending TR is read from the destination FIFO register entry 3 and the second pending TR is read from the destination FIFO register entry 0.

10.3.12.5 EDMA3TC Configuration

Table 10-22 provides the configuration of the individual EDMA3 transfer controllers present on the device. The default burst size (DBS) for each transfer controller is configurable using the TPTC_CFG register in the control module.

Table 10-22. EDMA3 Transfer Controller Configurations

Name	TC0	TC1	TC2	TC3
FIFOSIZE	512 bytes	512 bytes	512 bytes	512 bytes
BUSWIDTH	16 bytes	16 bytes	16 bytes	16 bytes
DSTREGDEPTH	4 entries	4 entries	4 entries	4 entries
DBS	Configurable	Configurable	Configurable	Configurable

10.3.13 Event Dataflow

This section summarizes the data flow of a single event, from the time the event is latched to the channel controller to the time the transfer completion code is returned. The following steps list the sequence of EDMA3CC activity:

1. Event is asserted from an external source (peripheral or external interrupt). This also is similar for a manually-triggered, chained-triggered, or QDMA-triggered event. The event is latched into the ER.E*n*/ERH.E*n* (or CER.E*n*/CERH.E*n*, ESR.E*n* /ESRH.E*n*, QER.E*n*) bit.
2. Once an event is prioritized and queued into the appropriate event queue, the SER.E*n*\SERH.E*n* (or QSER.E*n*) bit is set to inform the event prioritization/processing logic to disregard this event since it is already in the queue. Alternatively, if the transfer controller and the event queue are empty, then the event bypasses the queue.
3. The EDMA3CC processing and the submission logic evaluates the appropriate PaRAM set and determines whether it is a non-null and non-dummy transfer request (TR).
4. The EDMA3CC clears the ER.E*n*/ERH.E*n* (or CER.E*n*/CERH.E*n*, ESR.E*n*/ESRH.E*n*, QER.E*n*) bit and the SER.E*n*/SERH.E*n* bit as soon as it determines the TR is non-null. In the case of a null set, the SER.E*n*/SERH.E*n* bit remains set. It submits the non-null/non-dummy TR to the associated transfer controller. If the TR was programmed for early completion, the EDMA3CC immediately sets the interrupt pending register (IPR.I[TCC]/IPRH.I[TCC]-32).
5. If the TR was programmed for normal completion, the EDMA3CC sets the interrupt pending register (IPR.I[TCC]/IPRH.I[TCC]) when the EDMA3TC informs the EDMA3CC about completion of the transfer (returns transfer completion codes).
6. The EDMA3CC programs the associated EDMA3TC*n*'s Program Register Set with the TR.
7. The TR is then passed to the Source Active set and the DST FIFO Register Set, if both the register sets are available.
8. The Read Controller processes the TR by issuing read commands to the source slave endpoint. The Read Data lands in the Data FIFO of the EDMA3TC*n*.
9. As soon as sufficient data is available, the Write Controller begins processing the TR by issuing write commands to the destination slave endpoint.
10. This continues until the TR completes and the EDMA3TC*n* then signals completion status to the EDMA3CC.

10.3.14 EDMA3 Prioritization

The EDMA3 controller has many implementation rules to deal with concurrent events/channels, transfers, etc. The following subsections detail various arbitration details whenever there might be occurrence of concurrent activity. Figure 10-21 shows the different places EDMA3 priorities come into play.

10.3.14.1 Channel Priority

The DMA event registers (ER and ERH) capture up to 64 events; likewise, the QDMA event register (QER) captures QDMA events for all QDMA channels; therefore, it is possible for events to occur simultaneously on the DMA/QDMA event inputs. For events arriving simultaneously, the event associated with the lowest channel number is prioritized for submission to the event queues (for DMA events, channel 0 has the highest priority and channel 63 has the lowest priority; similarly, for QDMA events, channel 0 has the highest priority and channel 7 has the lowest priority). This mechanism only sorts simultaneous events for submission to the event queues.

If a DMA and QDMA event occurs simultaneously, the DMA event always has prioritization against the QDMA event for submission to the event queues.

10.3.14.2 Trigger Source Priority

If a DMA channel is associated with more than one trigger source (event trigger, manual trigger, and chain trigger), and if multiple events are set simultaneously for the same channel (ER.E*n* = 1, ESR.E*n* = 1 , CER.E*n* = 1) , then the EDMA3CC always services these events in the following priority order: event trigger (via ER) is higher priority than chain trigger (via CER) and chain trigger is higher priority than manual trigger (via ESR).

This implies that if for channel 0, both ER.E0 = 1 and CER.E0 = 1 at the same time, then the ER.E0 event is always queued before the CER.E0 event.

10.3.14.3 Dequeue Priority

The priority of the associated transfer request (TR) is further mitigated by which event queue is being used for event submission (dictated by DMAQNUM and QDMAQNUM). For submission of a TR to the transfer request, events need to be de-queued from the event queues. Queue 0 has the highest dequeue priority and queue 3 the lowest.

10.3.14.4 System (Transfer Controller) Priority

INIT_PRIORITY_0 and INIT_PRIORITY_1 registers in the chip configuration module are used to configure the EDMA TC's priority through the system bus infrastructure. Additionally, the priority settings for DDR memory accesses are defined in the dynamic memory manager (DMM).

NOTE: The default priority for all TCs is the same, 0 or highest priority relative to other masters. It is recommended that this priority be changed based on system level considerations, such as real-time deadlines for all masters including the priority of the transfer controllers with respect to each other.

10.3.15 EDMA3 Operating Frequency (Clock Control)

The EDMA3 channel controller and transfer controller are clocked from PLL_L3 SYSCLK4. The EDMA3 system runs at the L3 clock frequency.

10.3.16 Reset Considerations

A hardware reset resets the EDMA3 (EDMA3CC and EDMA3TC) and the EDMA3 configuration registers. The PaRAM memory contents are undefined after device reset and you should not rely on parameters to be reset to a known state. The PaRAM entry must be initialized to a desired value before it is used.

10.3.17 Power Management

The EDMA3 (EDMA3CC and EDMA3TC) can be placed in reduced-power modes to conserve power during periods of low activity. The power management of the peripheral is controlled by the power reset clock management (PRCM). The PRCM acts as a master controller for power management for all peripherals on the device.

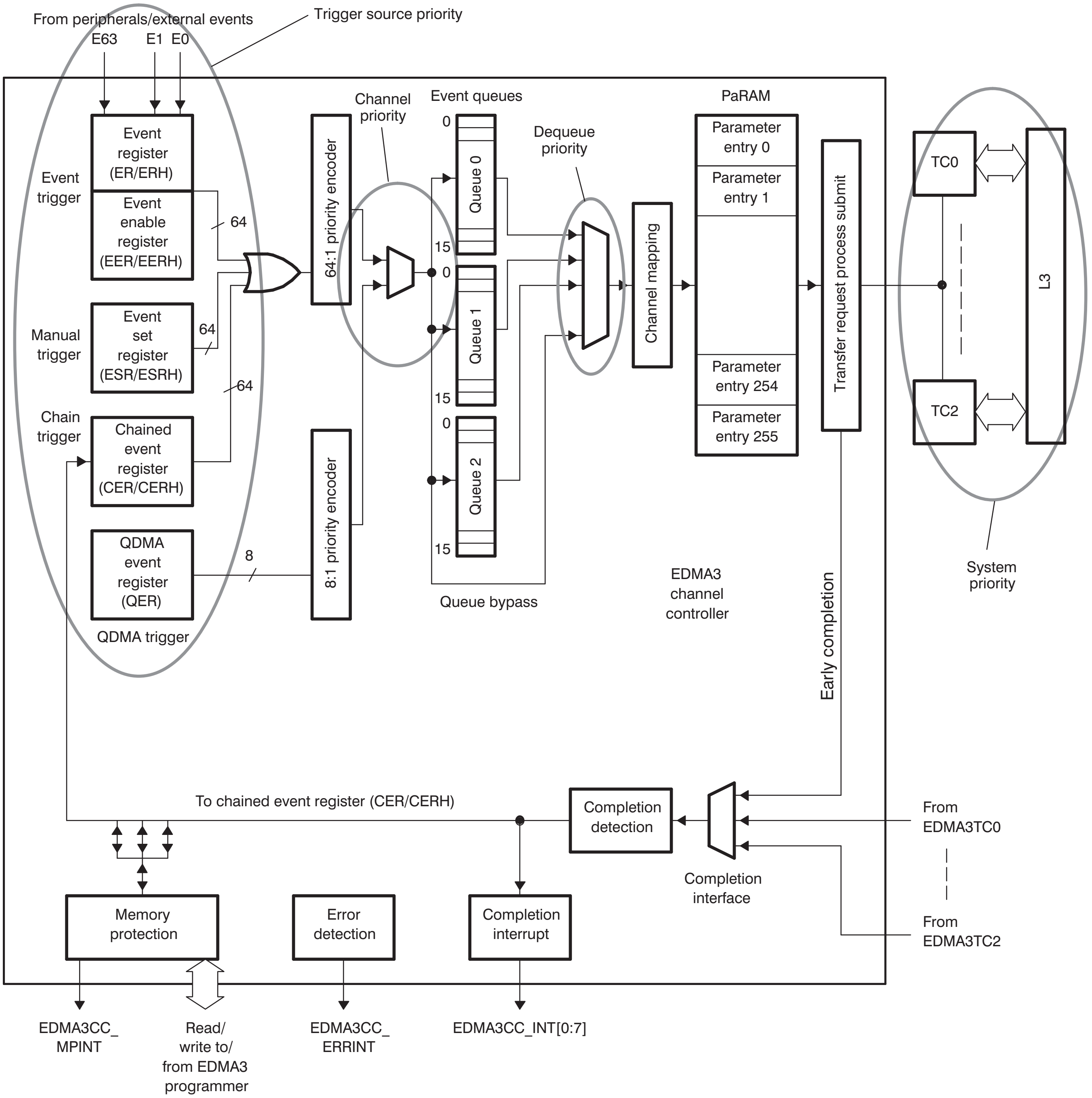
The EDMA3 controller can be idled on receiving a clock stop request from the PRCM. The requests to EDMA3CC and EDMA3TC are separate. In general, it should be verified that there are no pending activities in the EDMA3 controller

10.3.18 Emulation Considerations

During debug when using the emulator, the CPU(s) may be halted on an execute packet boundary for single-stepping, benchmarking, profiling, or other debug purposes. During an emulation halt, the EDMA3 channel controller and transfer controller operations continue. Events continue to be latched and processed and transfer requests continue to be submitted and serviced.

Since EDMA3 is involved in servicing multiple master and slave peripherals, it is not feasible to have an independent behavior of the EDMA3 for emulation halts. EDMA3 functionality would be coupled with the peripherals it is servicing, which might have different behavior during emulation halts. For example, if a McASP is halted during an emulation access (FREE = 0 and SOFT = 0 or 1 in McASP registers), the McASP stops generating the McASP receive or transmit events (REVT or XEVT) to the EDMA. From the point of view of the McASP, the EDMA3 is suspended, but other peripherals (for example, a timer) still assert events and will be serviced by the EDMA.

Figure 10-21. EDMA3 Prioritization



10.3.19 EDMA Transfer Examples

The EDMA3 channel controller performs a variety of transfers depending on the parameter configuration. The following sections provide a description and PaRAM configuration for some typical use case scenarios.

10.3.19.1 Block Move Example

The most basic transfer performed by the EDMA3 is a block move. During device operation it is often necessary to transfer a block of data from one location to another, usually between on-chip and off-chip memory.

In this example, a section of data is to be copied from external memory to internal L2 SRAM as shown in Figure 10-22. Figure 10-23 shows the parameters for this transfer.

The source address for the transfer is set to the start of the data block in external memory, and the destination address is set to the start of the data block in L2. If the data block is less than 64K bytes, the PaRAM configuration shown in Figure 10-23 holds true with the synchronization type set to A-synchronized and indexes cleared to 0. If the amount of data is greater than 64K bytes, BCNT and the B-indexes need to be set appropriately with the synchronization type set to AB-synchronized. The STATIC bit in OPT is set to prevent linking.

This transfer example may also be set up using QDMA. For successive transfer submissions, of a similar nature, the number of cycles used to submit the transfer are fewer depending on the number of changing transfer parameters. You may program the QDMA trigger word to be the highest numbered offset in the PaRAM set that undergoes change.

Figure 10-22. Block Move Example

Figure 10-23. Block Move Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents		Parameter	
0010 0008h		Channel Options Parameter (OPT)	
Channel Source Address (SRC)		Channel Source Address (SRC)	
0001h	0100h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0000h	0000h	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0000h	FFFFh	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content

31	30 28	27 24	23	22	21	20	19 18	17 16
0	000	0000	0	0	0	1	00	00
PRIV	Reserved	PRIVID	ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	Reserved	TCC

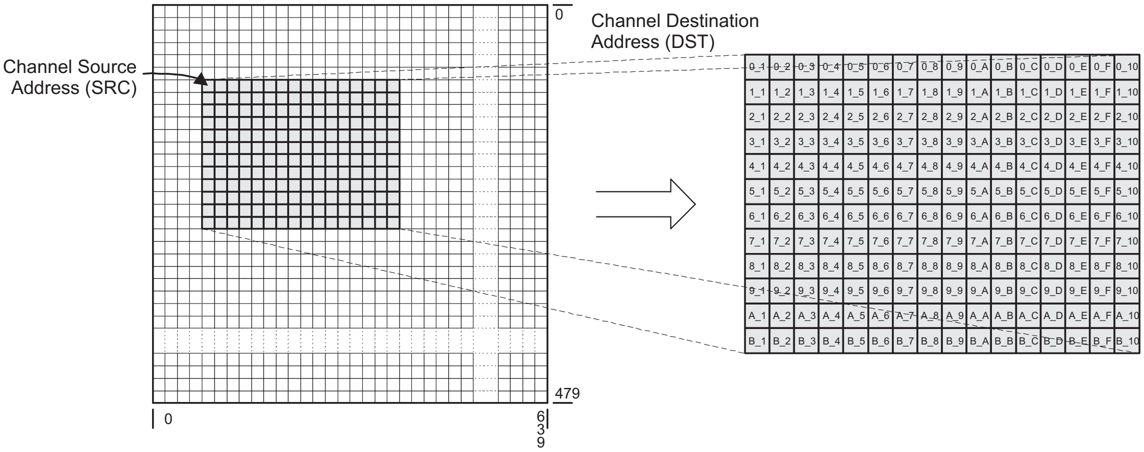
15 12	11	10 8	7 4	3	2	1	0
0000	0	000	0000	1	0	0	0
TCC	TCCMOD	FWID	Reserved	STATIC	SYNCDIM	DAM	SAM

10.3.19.2 Subframe Extraction Example

The EDMA3 can efficiently extract a small frame of data from a larger frame of data. By performing a 2D-to-1D transfer, the EDMA3 retrieves a portion of data for the CPU to process. In this example, a 640 × 480-pixel frame of video data is stored in external memory. Each pixel is represented by a 16-bit halfword. The CPU extracts a 16 × 12-pixel subframe of the image for processing. To facilitate more efficient processing time by the CPU, the EDMA3 places the subframe in internal L2 SRAM. Figure 10-24 shows the transfer of a subframe from external memory to L2. Figure 10-25 shows the parameters for this transfer.

The same PaRAM entry options are used for QDMA channels, as well as DMA channels. The STATIC bit in OPT is set to prevent linking. For successive transfers, only changed parameters need to be programmed before triggering the channel.

Figure 10-24. Subframe Extraction Example



0_1	0_2	0_3	0_4	0_5	0_6	0_7	0_8	0_9	0_A	0_B	0_C	0_D	0_E	0_F	0_10
1_1	1_2	1_3	1_4	1_5	1_6	1_7	1_8	1_9	1_A	1_B	1_C	1_D	1_E	1_F	1_10
2_1	2_2	2_3	2_4	2_5	2_6	2_7	2_8	2_9	2_A	2_B	2_C	2_D	2_E	2_F	2_10
3_1	3_2	3_3	3_4	3_5	3_6	3_7	3_8	3_9	3_A	3_B	3_C	3_D	3_E	3_F	3_10
4_1	4_2	4_3	4_4	4_5	4_6	4_7	4_8	4_9	4_A	4_B	4_C	4_D	4_E	4_F	4_10
5_1	5_2	5_3	5_4	5_5	5_6	5_7	5_8	5_9	5_A	5_B	5_C	5_D	5_E	5_F	5_10
6_1	6_2	6_3	6_4	6_5	6_6	6_7	6_8	6_9	6_A	6_B	6_C	6_D	6_E	6_F	6_10
7_1	7_2	7_3	7_4	7_5	7_6	7_7	7_8	7_9	7_A	7_B	7_C	7_D	7_E	7_F	7_10
8_1	8_2	8_3	8_4	8_5	8_6	8_7	8_8	8_9	8_A	8_B	8_C	8_D	8_E	8_F	8_10
9_1	9_2	9_3	9_4	9_5	9_6	9_7	9_8	9_9	9_A	9_B	9_C	9_D	9_E	9_F	9_10
A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_A	A_B	A_C	A_D	A_E	A_F	A_10
B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_A	B_B	B_C	B_D	B_E	B_F	B_10

Figure 10-25. Subframe Extraction Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents		Parameter	
0010 000Ch		Channel Options Parameter (OPT)	
Channel Source Address (SRC)		Channel Source Address (SRC)	
000Ch	0020h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0020h	0500h	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0000h	FFFFh	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content

31	30–28	27–24	23	22	21	20	19–18	17–16
0	000	0000	0	0	0	1	00	00
PRIV	Reserved	PRIVID	ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	Reserved	TCC

15–12	11	10–8	7–4	3	2	1	0
0000	0	000	0000	1	1	0	0
TCC	TCCMOD	FWID	Reserved	STATIC	SYNCDIM	DAM	SAM

10.3.19.3 Data Sorting Example

Many applications require the use of multiple data arrays; it is often desirable to have the arrays arranged such that the first elements of each array are adjacent, the second elements are adjacent, and so on. Often this is not how the data is presented to the device. Either data is transferred via a peripheral with the data arrays arriving one after the other or the arrays are located in memory with each array occupying a portion of contiguous memory spaces. For these instances, the EDMA3 can reorganize the data into the desired format. Figure 10-26 shows the data sorting.

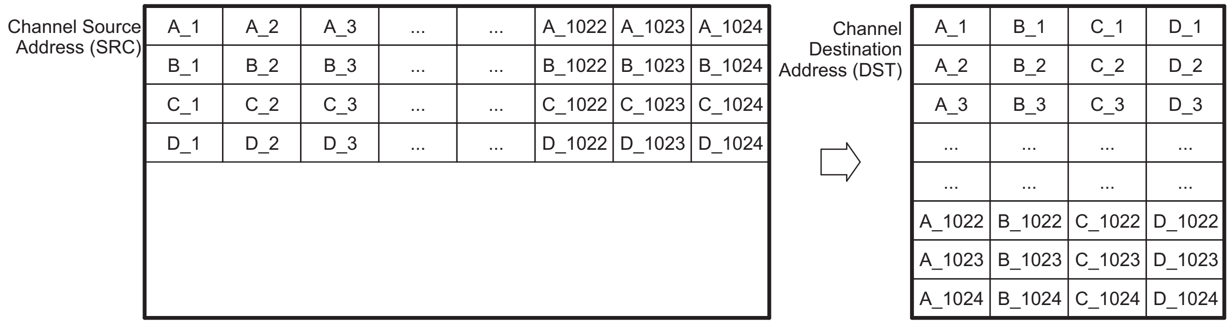
To determine the parameter set values, the following need to be considered:

- ACNT - Program this to be the size in bytes of an element.
- BCNT - Program this to be the number of elements in a frame.
- CCNT - Program this to be the number of frames.
- SRCBIDX - Program this to be the size of the element or ACNT.
- DSTBIDX - CCNT × ACNT
- SRCCDX - ACNT × BCNT
- DSTCIDX - ACNT

The synchronization type needs to be AB-synchronized and the STATIC bit is 0 to allow updates to the parameter set. It is advised to use normal EDMA3 channels for sorting.

It is not possible to sort this with a single trigger event. Instead, the channel can be programmed to be chained to itself. After BCNT elements get sorted, intermediate chaining could be used to trigger the channel again causing the transfer of the next BCNT elements and so on. Figure 10-27 shows the parameter set programming for this transfer, assuming channel 0 and an element size of 4 bytes.

Figure 10-26. Data Sorting Example



Channel Source Address (SRC)

A_1	A_2	A_3	...	...	A_1022	A_1023	A_1024
B_1	B_2	B_3	...	...	B_1022	B_1023	B_1024
C_1	C_2	C_3	...	...	C_1022	C_1023	C_1024
D_1	D_2	D_3	...	...	D_1022	D_1023	D_1024

Channel Destination Address (DST)

A_1	B_1	C_1	D_1
A_2	B_2	C_2	D_2
A_3	B_3	C_3	D_3
...	...	...	...
...	...	...	...
A_1022	B_1022	C_1022	D_1022
A_1023	B_1023	C_1023	D_1023
A_1024	B_1024	C_1024	D_1024

Figure 10-27. Data Sorting Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents		Parameter	
0090 0004h		Channel Options Parameter (OPT)	
Channel Source Address (SRC)		Channel Source Address (SRC)	
0400h	0004h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0010h	0001h	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0000h	FFFFh	BCNT Reload (BCNTRLD)	Link Address (LINK)
0001h	1000h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	0004h	Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content

31	30 28	27 24	23	22	21	20	19 18	17 16
0	000	0000	1	0	0	1	00	00
PRIV	Reserved	PRIVID	ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	Reserved	TCC

15 12	11	10 8	7 4	3	2	1	0
0000	0	000	0000	0	1	0	0
TCC	TCCMOD	FWID	Reserved	STATIC	SYNCDIM	DAM	SAM

10.3.19.4 Peripheral Servicing Example

The EDMA3 channel controller also services peripherals in the background of CPU operation, without requiring any CPU intervention. Through proper initialization of the EDMA3 channels, they can be configured to continuously service on-chip and off-chip peripherals throughout the device operation. Each event available to the EDMA3 has its own dedicated channel, and all channels operate simultaneously. The only requirements are to use the proper channel for a particular transfer and to enable the channel event in the event enable register (EER). When programming an EDMA3 channel to service a peripheral, it is necessary to know how data is to be presented to the processor. Data is always provided with some kind of synchronization event as either one element per event (non-bursting) or multiple elements per event (bursting).

10.3.19.4.1 Non-bursting Peripherals

Non-bursting peripherals include the on-chip multichannel audio serial port (McASP) and many external devices, such as codecs. Regardless of the peripheral, the EDMA3 channel configuration is the same.

The McASP transmit and receive data streams are treated independently by the EDMA3. The transmit and receive data streams can have completely different counts, data sizes, and formats. Figure 10-28 shows servicing incoming McASP data.

To transfer the incoming data stream to its proper location in DDR memory, the EDMA3 channel must be set up for a 1D-to-1D transfer with A-synchronization. Because an event (AREVT) is generated for every word as it arrives, it is necessary to have the EDMA3 issue the transfer request for each element individually. Figure 10-29 shows the parameters for this transfer. The source address of the EDMA3 channel is set to the data port address(DAT) for McASP, and the destination address is set to the start of the data block in DDR. Because the address of serializer buffer is fixed, the source B index is cleared to 0 (no modification) and the destination B index is set to 01b (increment).

Based on the premise that serial data is typically a high priority, the EDMA3 channel should be programmed to be on queue 0.

Figure 10-28. Servicing Incoming McASP Data Example

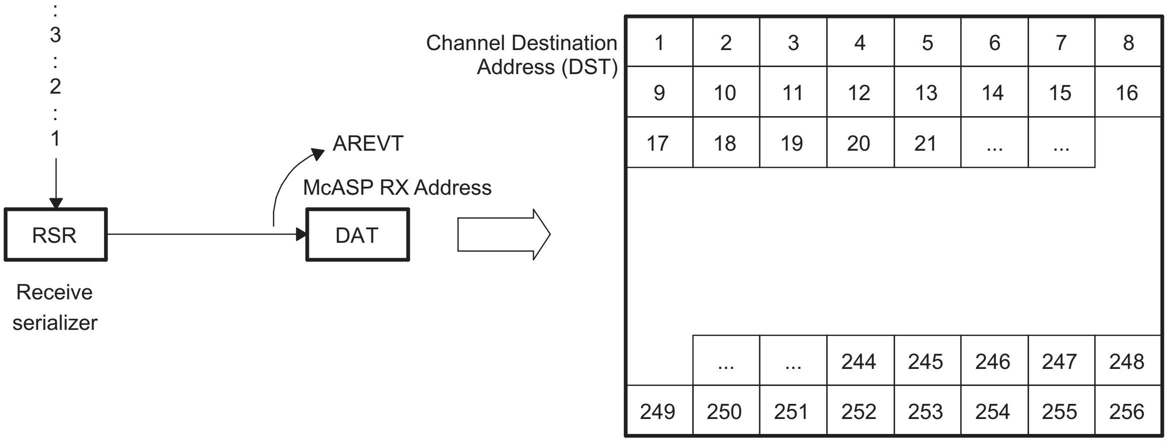


Figure 10-29. Servicing Incoming McASP Data Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents		**Parameter**	
0010 0000h		Channel Options Parameter (OPT)	
McASP RX Address		Channel Source Address (SRC)	
0100h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0001h	0000h	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0000h	FFFFh	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	0004h	Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content

31	30 28	27 24	23	22	21	20	19 18	17 16
0	000	0000	0	0	0	1	00	00
PRIV	Reserved	PRIVID	ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	Reserved	TCC

15 12	11	10 8	7 4	3	2	1	0
0000	0	000	0000	0	0	0	0
TCC	TCCMOD	FWID	Reserved	STATIC	SYNCDIM	DAM	SAM

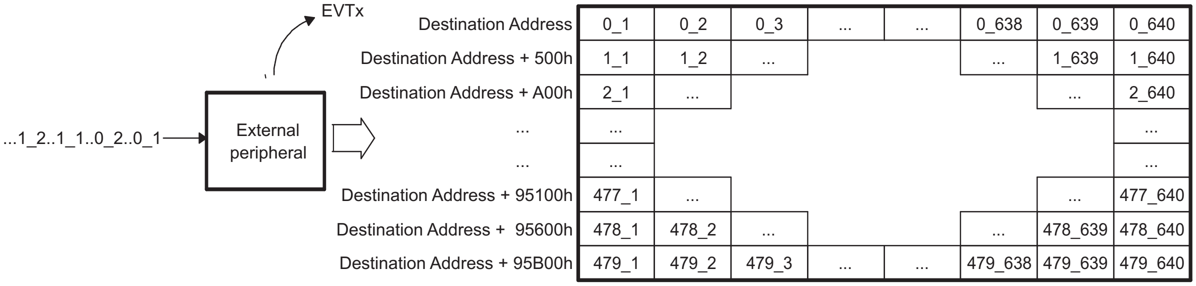
10.3.19.4.2 Bursting Peripherals

Higher bandwidth applications require that multiple data elements be presented to the processor core for every synchronization event. This frame of data can either be from multiple sources that are working simultaneously or from a single high-throughput peripheral that streams data to/from the processor.

In this example, a port is receiving a video frame from a camera and presenting it to the processor one array at a time. The video image is 640 × 480 pixels, with each pixel represented by a 16-bit element. The image is to be stored in external memory. Figure 10-30 shows this example.

To transfer data from an external peripheral to an external buffer one array at a time based on EVT*n* , channel *n* must be configured. Due to the nature of the data (a video frame made up of arrays of pixels) the destination is essentially a 2D entity. Figure 10-31 shows the parameters to service the incoming data with a 1D-to-2D transfer using AB-synchronization. The source address is set to the location of the video framer peripheral, and the destination address is set to the start of the data buffer. Because the input address is static, the SRCBIDX is 0 (no modification to the source address). The destination is made up of arrays of contiguous, linear elements; therefore, the DSTBIDX is set to pixel size, 2 bytes. ANCT is equal to the pixel size, 2 bytes. BCNT is set to the number of pixels in an array, 640. CCNT is equal to the total number of arrays in the block, 480. SRCCIDX is 0 because the source address undergoes no increment. The DSTCIDX is equal to the difference between the starting addresses of each array. Because a pixel is 16 bits (2 bytes), DSTCIDX is equal to 640 × 2.

Figure 10-30. Servicing Peripheral Burst Example



Destination Address	0_1	0_2	0_3	...	...	0_638	0_639	0_640
Destination Address + 500h	1_1	1_2	...			...	1_639	1_640
Destination Address + A00h	2_1	...					...	2_640
...	...							...
...	...							...
Destination Address + 95100h	477_1	...					...	477_640
Destination Address + 95600h	478_1	478_2	...			...	478_639	478_640
Destination Address + 95B00h	479_1	479_2	479_3	...	...	479_638	479_639	479_640

Figure 10-31. Servicing Peripheral Burst Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents		Parameter	
0010 0004h		Channel Options Parameter (OPT)	
Channel Source Address		Channel Source Address (SRC)	
0280h	0002h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address		Channel Destination Address (DST)	
0002h	0000h	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0000h	FFFFh	BCNT Reload (BCNTRLD)	Link Address (LINK)
0500h	0000h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	01E0h	Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content

31	30 28	27 24	23	22	21	20	19 18	17 16
0	000	0000	0	0	0	1	00	00
PRIV	Reserved	PRIVID	ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	Reserved	TCC

15 12	11	10 8	7 4	3	2	1	0
0000	0	000	0000	0	1	0	0
TCC	TCCMOD	FWID	Reserved	STATIC	SYNCDIM	DAM	SAM

10.3.19.4.3 Continuous Operation

Configuring an EDMA3 channel to receive a single frame of data is useful, and is applicable to some systems. A majority of the time, however, data is going to be continuously transmitted and received throughout the entire operation of the processor. In this case, it is necessary to implement some form of linking such that the EDMA3 channels continuously reload the necessary parameter sets. In this example, McASP is configured to transmit and receive data on a T1 array. To simplify the example, only two channels are active for both transmit and receive data streams. Each channel receives packets of 128 elements. The packets are transferred from the serial port to internal memory and from internal memory to the serial port, as shown Figure 10-32.

The McASP generates AREVT for every element received and generates AXEVT for every element transmitted. To service the data streams, the DMA channels associated with the McASP must be setup for 1D-to-1D transfers with A-synchronization.

Figure 10-33 shows the parameter entries for the channel for these transfers. To service the McASP continuously, the channels must be linked to a duplicate PaRAM set in the PaRAM. After all frames have been transferred, the EDMA3 channels reload and continue. Figure 10-34 shows the reload parameters for the channel.

10.3.19.4.3.1 Receive Channel

EDMA3 channel 15 services the incoming data stream of McASP. The source address is set to that of the receive serializer buffer, and the destination address is set to the first element of the data block. Because there are two data channels being serviced, A and B, they are to be located separately within the L2 SRAM.

To facilitate continuous operation, a copy of the PaRAM set for the channel is placed in PaRAM set 64. The LINK option is set and the link address is provided in the PaRAM set. Upon exhausting the channel 15 parameter set, the parameters located at the link address are loaded into the channel 15 parameter set and operation continues. This function continues throughout device operation until halted by the CPU.

10.3.19.4.3.2 Transmit Channel

EDMA3 channel 12 services the outgoing data stream of McASP. In this case the destination address needs no update, hence, the parameter set changes accordingly. Linking is also used to allow continuous operation by the EDMA3 channel, with duplicate PaRAM set entries at PaRAM set 65.

Figure 10-32. Servicing Continuous McASP Data Example

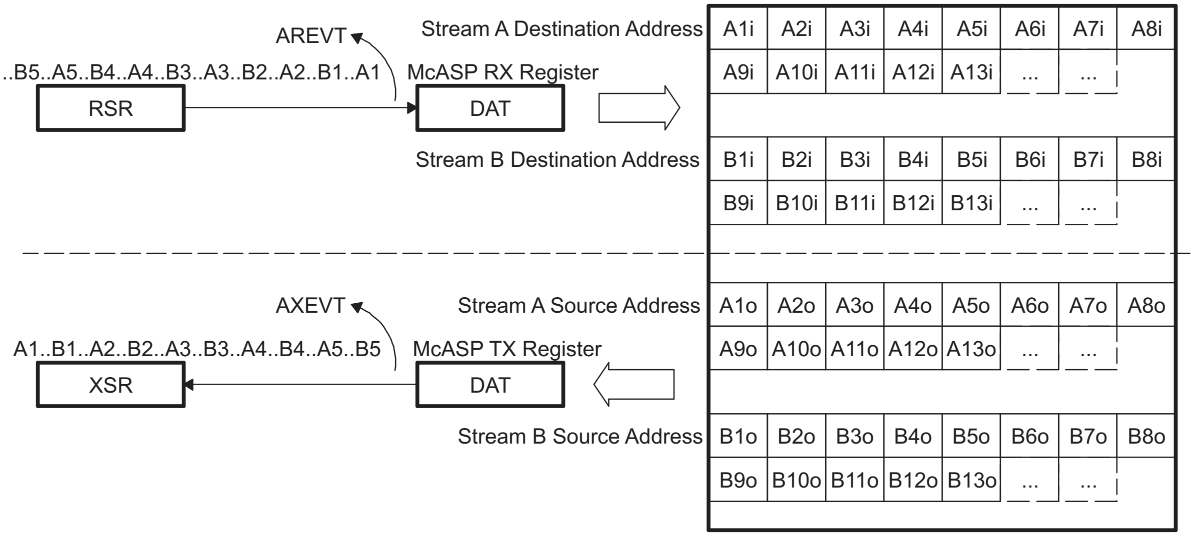


Figure 10-33. Servicing Continuous McASP Data Example PaRAM Configuration

(a) EDMA Parameters for Receive Channel (PaRAM Set 15) being Linked to PaRAM Set 64

Parameter Contents		**Parameter**	
0010 0000h		Channel Options Parameter (OPT)	
McASP RX Register		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0001h	0000h	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0080h	4800h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	FFFFh	Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content for Receive Channel (PaRAM Set 15)

31	30 28	27 24	23	22	21	20	19 18	17 16
0	000	0000	0	0	0	1	00	00
PRIV	Reserved	PRIVID	ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	Reserved	TCC

15 12	11	10 8	7 4	3	2	1	0
0000	0	000	0000	0	0	0	0
TCC	TCCMOD	FWID	Reserved	STATIC	SYNCDIM	DAM	SAM

(c) EDMA Parameters for Transmit Channel (PaRAM Set 12) being Linked to PaRAM Set 65

Parameter Contents		**Parameter**	
0010 1000h		Channel Options Parameter (OPT)	
Channel Source Address (SRC)		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
McASP TX Register		Channel Destination Address (DST)	
0000h	0001h	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0080h	4860h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	FFFFh	Reserved	Count for 3rd Dimension (CCNT)

(d) Channel Options Parameter (OPT) Content for Transmit Channel (PaRAM Set 12)

31	30 28	27 24	23	22	21	20	19 18	17 16
0	000	0000	0	0	0	1	00	00
PRIV	Reserved	PRIVID	ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	Reserved	TCC

15 12	11	10 8	7 4	3	2	1	0
0001	0	000	0000	0	0	0	0
TCC	TCCMOD	FWID	Reserved	STATIC	SYNCDIM	DAM	SAM

Figure 10-34. Servicing Continuous McASP Data Example Reload PaRAM Configuration

(a) EDMA Reload Parameters (PaRAM Set 64) for Receive Channel

Parameter Contents		Parameter	
0010 0000h		Channel Options Parameter (OPT)	
McASP RX Register		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0001h	0000h	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0080h	4800h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	FFFFh	Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content for Receive Channel (PaRAM Set 64)

31	30 28	27 24	23	22	21	20	19 18	17 16
0	000	0000	0	0	0	1	00	00
PRIV	Reserved	PRIVID	ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	Reserved	TCC

15 12	11	10 8	7 4	3	2	1	0
0000	0	000	0000	0	0	0	0
TCC	TCCMOD	FWID	Reserved	STATIC	SYNCDIM	DAM	SAM

(c) EDMA Reload Parameters (PaRAM Set 65) for Transmit Channel

Parameter Contents		Parameter	
0010 1000h		Channel Options Parameter (OPT)	
Channel Source Address (SRC)		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
McASP TX Register		Channel Destination Address (DST)	
0000h	0001h	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0080h	4860h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	FFFFh	Reserved	Count for 3rd Dimension (CCNT)

(d) Channel Options Parameter (OPT) Content for Transmit Channel (PaRAM Set 65)

31	30 28	27 24	23	22	21	20	19 18	17 16
0	000	0000	0	0	0	1	00	00
PRIV	Reserved	PRIVID	ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	Reserved	TCC

15 12	11	10 8	7 4	3	2	1	0
0001	0	000	0000	0	0	0	0
TCC	TCCMOD	FWID	Reserved	STATIC	SYNCDIM	DAM	SAM

10.3.19.4.4 Ping-Pong Buffering

Although the previous configuration allows the EDMA3 to service a peripheral continuously, it presents a number of restrictions to the CPU. Because the input and output buffers are continuously being filled/emptied, the CPU must match the pace of the EDMA3 very closely to process the data. The EDMA3 receive data must always be placed in memory before the CPU accesses it, and the CPU must provide the output data before the EDMA3 transfers it. Though not impossible, this is an unnecessary challenge. It is particularly difficult in a 2-level cache scheme.

Ping-pong buffering is a simple technique that allows the CPU activity to be distanced from the EDMA3 activity. This means that there are multiple (usually two) sets of data buffers for all incoming and outgoing data streams. While the EDMA3 transfers the data into and out of the ping buffers, the CPU manipulates the data in the pong buffers. When both CPU and EDMA3 activity completes, they switch. The EDMA3 then writes over the old input data and transfers the new output data. Figure 10-35 shows the ping-pong scheme for this example.

To change the continuous operation example, such that a ping-pong buffering scheme is used, the EDMA3 channels need only a moderate change. Instead of one parameter set, there are two; one for transferring data to/from the ping buffers and one for transferring data to/from the pong buffers. As soon as one transfer completes, the channel loads the PaRAM set for the other and the data transfers continue. Figure 10-36 shows the EDMA3 channel configuration required.

Each channel has two parameter sets, ping and pong. The EDMA3 channel is initially loaded with the ping parameters (Figure 10-36). The link address for the ping set is set to the PaRAM offset of the pong parameter set (Figure 10-37). The link address for the pong set is set to the PaRAM offset of the ping parameter set (Figure 10-38). The channel options, count values, and index values are all identical between the ping and pong parameters for each channel. The only differences are the link address provided and the address of the data buffer.

10.3.19.4.4.1 Synchronization with the CPU

To utilize the ping-pong buffering technique, the system must signal the CPU when to begin to access the new data set. After the CPU finishes processing an input buffer (ping), it waits for the EDMA3 to complete before switching to the alternate (pong) buffer. In this example, both channels provide their channel numbers as their report word and set the TCINTEN bit to generate an interrupt after completion. When channel 15 fills an input buffer, the E15 bit in the interrupt pending register (IPR) is set; when channel 12 empties an output buffer, the E12 bit in IPR is set. The CPU must manually clear these bits. With the channel parameters set, the CPU polls IPR to determine when to switch. The EDMA3 and CPU could alternatively be configured such that the channel completion interrupts the CPU. By doing this, the CPU could service a background task while waiting for the EDMA3 to complete.

Figure 10-35. Ping-Pong Buffering for McASP Data Example

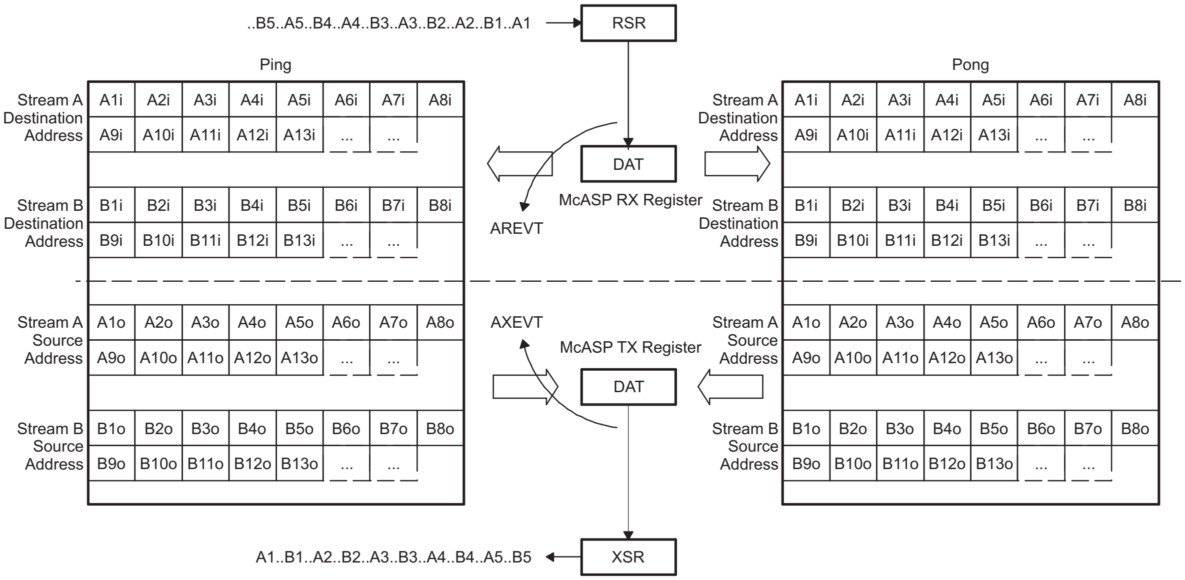


Figure 10-36. Ping-Pong Buffering for McASP Example PaRAM Configuration

(a) EDMA Parameters for Channel 15 (Using PaRAM Set 15 Linked to Pong Set 64)

Parameter Contents		**Parameter**	
0010 D000h		Channel Options Parameter (OPT)	
McASP RX Register		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0001h	0000h	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0080h	4800h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content for Channel 15

31	30 28	27 24	23	22	21	20	19 18	17 16
0	000	0000	0	0	0	1	00	00
PRIV	Reserved	PRIVID	ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	Reserved	TCC

15 12	11	10 8	7 4	3	2	1	0
1101	0	000	0000	0	0	0	0
TCC	TCCMOD	FWID	Reserved	STATIC	SYNCDIM	DAM	SAM

(c) EDMA Parameters for Channel 12 (Using PaRAM Set 12 Linked to Pong Set 66)

Parameter Contents		**Parameter**	
0010 C000h		Channel Options Parameter (OPT)	
Channel Source Address (SRC)		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
McASP TX Register		Channel Destination Address (DST)	
0000h	0001h	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0080h	4840h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

(d) Channel Options Parameter (OPT) Content for Channel 12

31	30 28	27 24	23	22	21	20	19 18	17 16
0	000	0000	0	0	0	1	00	00
PRIV	Reserved	PRIVID	ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	Reserved	TCC

15 12	11	10 8	7 4	3	2	1	0
1100	0	000	0000	0	0	0	0
TCC	TCCMOD	FWID	Reserved	STATIC	SYNCDIM	DAM	SAM

Figure 10-37. Ping-Pong Buffering for McASP Example Pong PaRAM Configuration

(a) EDMA Pong Parameters for Channel 15 at Set 64 Linked to Set 65

Parameter Contents		**Parameter**	
0010 D000h		Channel Options Parameter (OPT)	
McASP RX Register		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0001h	0000h	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0080h	4820h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

(b) EDMA Pong Parameters for Channel 12 at Set 66 Linked to Set 67

Parameter Contents		**Parameter**	
0010 C000h		Channel Options Parameter (OPT)	
Channel Source Address (SRC)		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
McASP TX Register		Channel Destination Address (DST)	
0000h	0001h	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0080h	4860h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

Figure 10-38. Ping-Pong Buffering for McASP Example Ping PaRAM Configuration

(a) EDMA Ping Parameters for Channel 15 at Set 65 Linked to Set 64

Parameter Contents		Parameter	
0010 D000h		Channel Options Parameter (OPT)	
McASP RX Register		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0001h	0000h	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0080h	4800h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

(b) EDMA Ping Parameters for Channel 12 at Set 67 Linked to Set 66

Parameter Contents		Parameter	
0010 C000h		Channel Options Parameter (OPT)	
Channel Source Address (SRC)		Channel Source Address (SRC)	
0080h	0001h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
McASP TX Register		Channel Destination Address (DST)	
0000h	0001h	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0080h	4840h	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

10.3.19.4.5 Transfer Chaining Examples

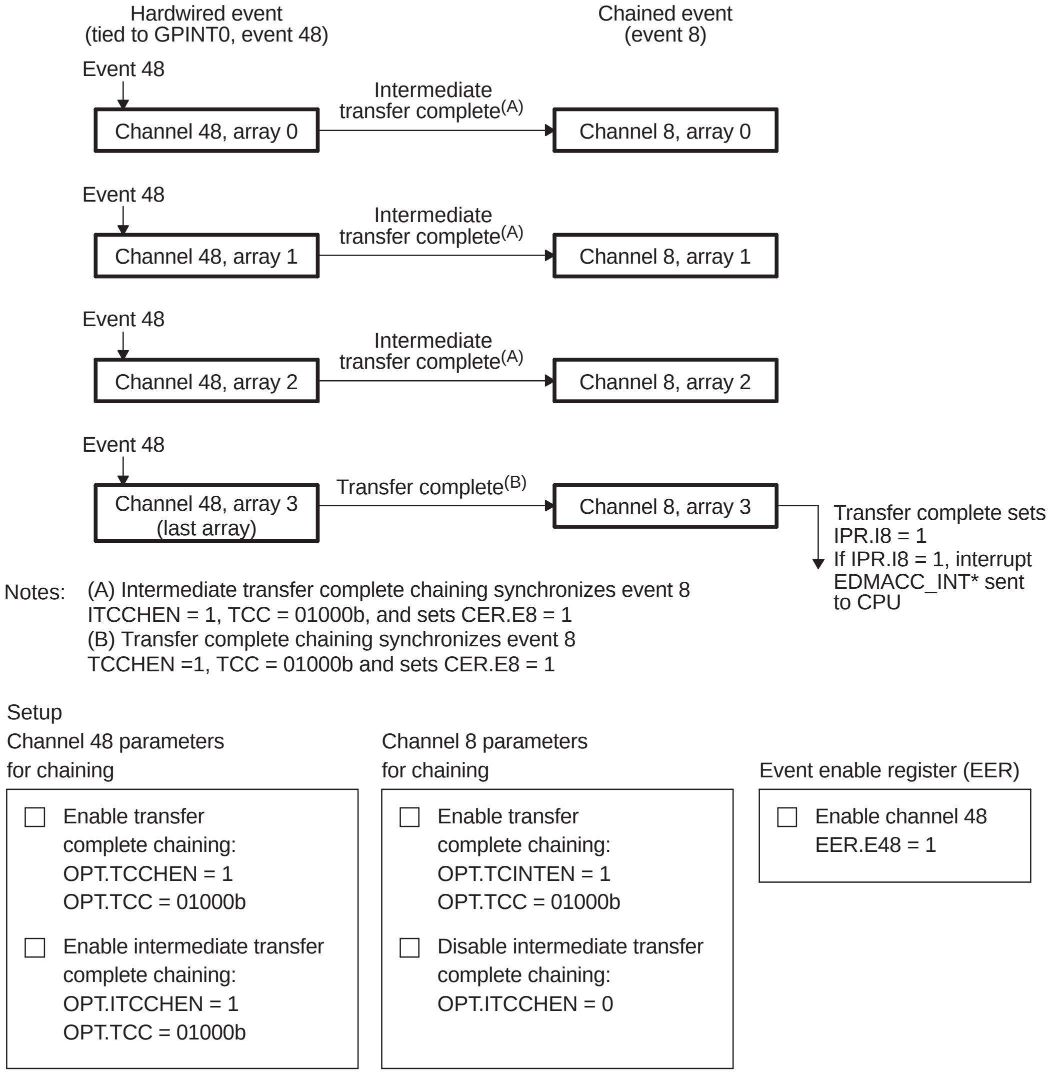
The following examples explain the intermediate transfer complete chaining function.

10.3.19.4.5.1 Servicing Input/Output FIFOs with a Single Event

Many systems require the use of a pair of external FIFOs that must be serviced at the same rate. One FIFO buffers data input, and the other buffers data output. The EDMA3 channels that service these FIFOs can be set up for AB-synchronized transfers. While each FIFO is serviced with a different set of parameters, both can be signaled from a single event. For example, an external interrupt pin can be tied to the status flags of one of the FIFOs. When this event arrives, the EDMA3 needs to perform servicing for both the input and output streams. Without the intermediate transfer complete chaining feature this would require two events, and thus two external interrupt pins. The intermediate transfer complete chaining feature allows the use of a single external event (for example, a GPIO event). Figure 10-39 shows the EDMA3 setup and illustration for this example.

A GPIO event (in this case, GPINT0) triggers an array transfer. Upon completion of each intermediate array transfer of channel 48, intermediate transfer complete chaining sets the E8 bit (specified by TCC of 8) in the chained event register (CER) and provides a synchronization event to channel 8. Upon completion of the last array transfer of channel 48, transfer complete chaining—not intermediate transfer complete chaining—sets the E8 bit in CER (specified by TCCMODE:TCC) and provides a synchronization event to channel 8. The completion of channel 8 sets the I8 bit (specified by TCCMODE:TCC) in the interrupt pending register (IPR), which can generate an interrupt to the CPU, if the I8 bit in the interrupt enable register (IER) is set.

Figure 10-39. Intermediate Transfer Completion Chaining Example



Notes: (A) Intermediate transfer complete chaining synchronizes event 8
ITCCHEN = 1, TCC = 01000b, and sets CER.E8 = 1
(B) Transfer complete chaining synchronizes event 8
TCCHEN =1, TCC = 01000b and sets CER.E8 = 1

Setup

Channel 48 parameters for chaining

☐ Enable transfer complete chaining:
OPT.TCCHEN = 1
OPT.TCC = 01000b

☐ Enable intermediate transfer complete chaining:
OPT.ITCCHEN = 1
OPT.TCC = 01000b

Channel 8 parameters for chaining

☐ Enable transfer complete chaining:
OPT.TCINTEN = 1
OPT.TCC = 01000b

☐ Disable intermediate transfer complete chaining:
OPT.ITCCHEN = 0

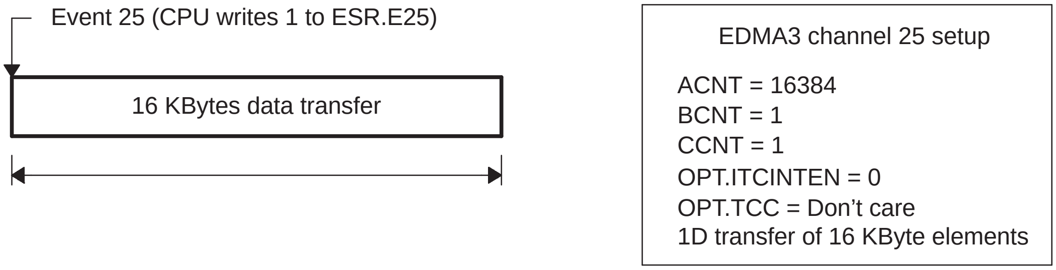
Event enable register (EER)

☐ Enable channel 48
EER.E48 = 1

10.3.19.4.5.2 Breaking Up Large Transfers with Intermediate Chaining

Another feature of intermediate transfer chaining (ITCCHEN) is for breaking up large transfers. A large transfer may lock out other transfers of the same priority level for the duration of the transfer. For example, a large transfer on queue 0 from the internal memory to the external memory using the EMIF may starve other EDMA3 transfers on the same queue. In addition, this large high-priority transfer may prevent the EMIF for a long duration to service other lower priority transfers. When a large transfer is considered to be high priority, it should be split into multiple smaller transfers. Figure 10-40 shows the EDMA3 setup and illustration of an example single large block transfer.

Figure 10-40. Single Large Block Transfer Example

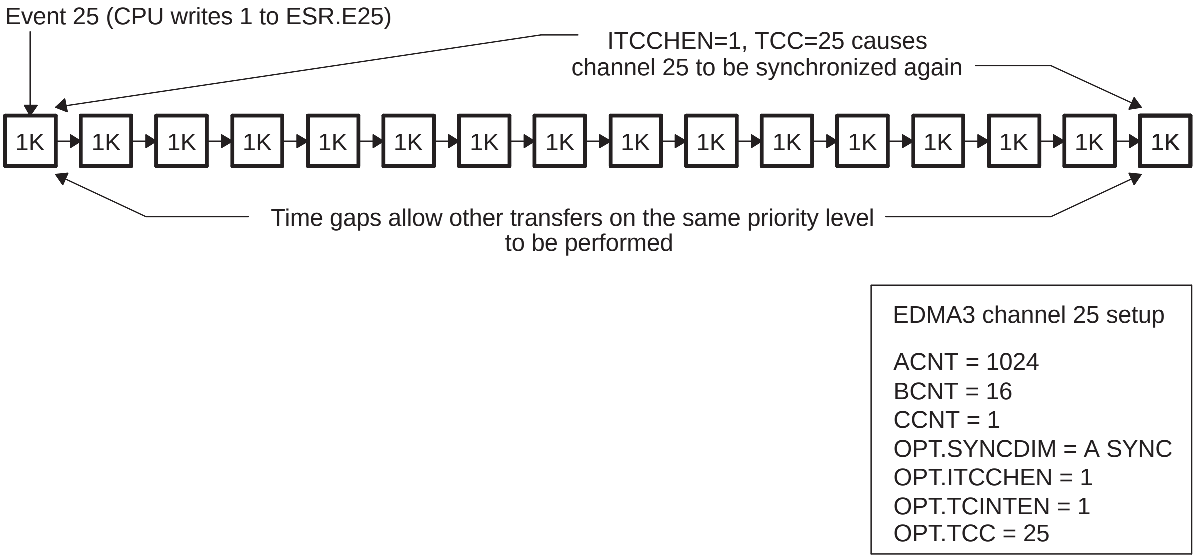


EDMA3 channel 25 setup

ACNT = 16384
BCNT = 1
CCNT = 1
OPT.ITCINTEN = 0
OPT.TCC = Don’t care
1D transfer of 16 KByte elements

The intermediate transfer chaining enable (ITCCHEN) provides a method to break up a large transfer into smaller transfers. For example, to move a single large block of memory (16K bytes), the EDMA3 performs an A-synchronized transfer. The element count is set to a reasonable value, where reasonable derives from the amount of time it would take to move this smaller amount of data. Assume 1 Kbyte is a reasonable small transfer in this example. The EDMA3 is set up to transfer 16 arrays of 1 Kbyte elements, for a total of 16K byte elements. The TCC field in the channel options parameter (OPT) is set to the same value as the channel number and ITCCHEN are set. In this example, EDMA3 channel 25 is used and TCC is also set to 25. The TCINTEN may also be set to trigger interrupt 25 when the last 1 Kbyte array is transferred. The CPU starts the EDMA3 transfer by writing to the appropriate bit of the event set register (ESR.E25). The EDMA3 transfers the first 1 Kbyte array. Upon completion of the first array, intermediate transfer complete code chaining generates a synchronization event to channel 25, a value specified by the TCC field. This intermediate transfer completion chaining event causes EDMA3 channel 25 to transfer the next 1 Kbyte array. This process continues until the transfer parameters are exhausted, at which point the EDMA3 has completed the 16K byte transfer. This method breaks up a large transfer into smaller packets, thus providing natural time slices in the transfer such that other events may be processed. Figure 10-41 shows the EDMA3 setup and illustration of the broken up smaller packet transfers.

Figure 10-41. Smaller Packet Data Transfers Example



EDMA3 channel 25 setup

ACNT = 1024
BCNT = 16
CCNT = 1
OPT.SYNCDIM = A SYNC
OPT.ITCCHEN = 1
OPT.TCINTEN = 1
OPT.TCC = 25

10.3.20 EDMA Events

10.3.20.1 Direct Mapped

Table 10-23. Direct Mapped

Event Number	Event Name	Source Module
0	pr1_host[7]	PRU_ICSS1
1	pr1_host[6]	PRU_ICSS1
2	SDTXEVT1	MMCSD1
3	SDRXEVT1	MMCSD1
4	Reserved	
5	Reserved	
6	Reserved	
7	Reserved	
8	AXEVT0	MCASP0
9	AREVT0	MCASP0
10	AXEVT1	MCASP1
11	AREVT1	MCASP1
12	pi_x_dma_event_intr4	DMA_INTR_PIN4
13	GPIO5EVT0	GPIO5
14	eHRPWMEVT0	PWMSS0
15	eHRPWMEVT1	PWMSS1
16	SPI0XEVT0	MCSPI0
17	SPI0REVT0	MCSPI0
18	SPI0XEVT1	MCSPI0
19	SPI0REVT1	MCSPI0
20	SPI0XEVT2	MCSPI0
21	SPI0REVT2	MCSPI0
22	GPIO0EVT0	GPIO0
23	GPIO1EVT0	GPIO1
24	SDTXEVT0	MMCSD0
25	SDRXEVT0	MMCSD0
26	UTXEVT0	UART0
27	URXEVT0	UART0
28	UTXEVT1	UART1
29	URXEVT1	UART1
30	UTXEVT2	UART2
31	URXEVT2	UART2
32	Reserved	
33	Reserved	
34	Reserved	
35	Reserved	
36	Reserved	
37	Reserved	
38	eCAPEVT0	PWMSS0
39	eCAPEVT1	PWMSS1
40	CAN0_IF1DMA	DCAN0
41	CAN0_IF2DMA	DCAN0
42	SPI1XEVT0	MCSPI1

Table 10-23. Direct Mapped (continued)

Event Number	Event Name	Source Module
43	SPI1REVT0	MCSPI1
44	SPI1XEVT1	MCSPI1
45	SPI1REVT1	MCSPI1
46	eQEPEVT0	PWMSS0
47	CAN0_IF3DMA	DCAN0
48	TINT4	DMTIMER4
49	TINT5	DMTIMER5
50	TINT6	DMTIMER6
51	TINT7	DMTIMER7
52	GPMCEVT	GPMC
53	adc0_FIFOevent_dpend	ADC0
54	adc1_FIFOevent_dpend	ADC1
55	adc1_FIFO1event_dpend	ADC1
56	eQEPEVT1	PWMSS1
57	adc0_FIFO1event_dpend	ADC0
58	I2CTXEVT0	I2C0
59	I2CRXEVT0	I2C0
60	I2CTXEVT1	I2C1
61	I2CRXEVT1	I2C1
62	eCAPEVT2	PWMSS2
63	eHRPWMEVT2	PWMSS2

(1) pr1_host_intr[0:7] corresponds to Host-2 to Host-9 of the PRU-ICSS interrupt controller.

10.3.20.2 Crossbar Mapped

Table 10-24. Crossbar Mapped

Event Number	Event Name	Source Module
1	SDTXEVT2	MMCSD2
2	SDRXEVT2	MMCSD2
3	I2CTXEVT2	I2C2
4	I2CRXEVT2	I2C2
5	SPI1XEVT2	MCSPI1
6	SPI1REVT2	MCSPI1
7	UTXEVT3	UART3
8	URXEVT3	UART3
9	UTXEVT4	UART4
10	URXEVT4	UART4
11	UTXEVT5	UART5
12	URXEVT5	UART5
13	CAN1_IF1DMA	DCAN1
14	CAN1_IF2DMA	DCAN1
15	CAN1_IF3DMA	DCAN1
16	SPI2XEVT0	MCSPI2
17	SPI2REVT0	MCSPI2
18	SPI2XEVT1	MCSPI2
19	SPI2REVT1	MCSPI2

Table 10-24. Crossbar Mapped (continued)

Event Number	Event Name	Source Module
20	SPI2XEVT2	MCSPI2
21	SPI2REVT2	MCSPI2
22	TINT0	DMTIMER0
23	pi_x_dma_event_intr3	DMA_INTR_PIN3
24	TINT2	DMTIMER2
25	TINT3	DMTIMER3
26	SPI0XEVT3	MCSPI0
27	SPI0REVT3	MCSPI0
28	pi_x_dma_event_intr0	DMA_INTR_PIN0
29	pi_x_dma_event_intr1	DMA_INTR_PIN1
30	pi_x_dma_event_intr2	DMA_INTR_PIN2
31	eQEPEVT2	PWMSS2
32	GPIO2EVT0	GPIO2
33	DSS_DMA_REQ0	DSS
34	DSS_DMA_REQ1	DSS
35	DSS_DMA_REQ2	DSS
36	DSS_DMA_REQ3	DSS
37	DSS_LINE_TRIGGER	DSS
38	GPIO3EVT0	GPIO3
39	GPIO4EVT0	GPIO4
40	SPI1XEVT3	MCSPI1
41	SPI1REVT3	MCSPI1
42	SPI2XEVT3	MCSPI2
43	SPI2REVT3	MCSPI2
44	Reserved	
45	Reserved	
46	Reserved	
47	Reserved	
48	TINT8	DMTIMER8
49	TINT9	DMTIMER9
50	TINT10	DMTIMER10
51	TINT11	DMTIMER11
52	pi_x_dma_event_intr5	DMA_INTR_PIN5
53	SPI3XEVT0	MCSPI3
54	SPI3REVT0	MCSPI3
55	SPI4XEVT0	MCSPI4
56	SPI4REVT0	MCSPI4
57	SPI3XEVT1	MCSPI3
58	SPI3REVT1	MCSPI3
59	SPI4XEVT1	MCSPI4
60	SPI4REVT1	MCSPI4
61	eHRPWMEVT3	PWMSS3
62	eHRPWMEVT4	PWMSS4
63	eHRPWMEVT5	PWMSS5

10.4 Registers

NOTE: The EDMA3CC register descriptions in Section 10.4.1 do not describe shadow registers. For a description of shadow registers, see Section 10.3.7, *EDMA3 Channel Controller Regions*.

10.4.1 EDMA3CC Registers

Table 10-25 lists the memory-mapped registers for the EDMA3CC. All register offset addresses not listed in Table 10-25 should be considered as reserved locations and the register contents should not be modified.

Table 10-25. EDMA3CC Registers

Offset	Acronym	Register Name	Section
0h	PID	Peripheral Identification Register	Section 10.4.2.1
4h	CCCFG	EDMA3CC Configuration Register	Section 10.4.1.2
10h	SYSCONFIG	EDMA3CC System Configuration Register	Section 20.1.3.2
100h to 1FCh	DCHMAP_0 to DCHMAP_63	DMA Channel Mapping Registers 0-63	Section 10.4.1.4
200h to 21Ch	QCHMAP_0 to QCHMAP_7	QDMA Channel Mapping Registers 0-7	Section 10.4.1.5
240h to 25Ch	DMAQNUM_0 to DMAQNUM_7	DMA Queue Number Registers 0-7	Section 10.4.1.6
260h	QDMAQNUM	QDMA Queue Number Register	Section 10.4.1.7
284h	QUEPRI	Queue Priority Register	Section 10.4.1.8
300h	EMR	Event Missed Register	Section 10.4.1.9
304h	EMRH	Event Missed Register High	Section 10.4.1.10
308h	EMCR	Event Missed Clear Register	Section 10.4.1.11
30Ch	EMCRH	Event Missed Clear Register High	Section 10.4.1.12
310h	QEMR	QDMA Event Missed Register	Section 10.4.1.13
314h	QEMCR	QDMA Event Missed Clear Register	Section 10.4.1.14
318h	CCERR	EDMA3CC Error Register	Section 10.4.1.15
31Ch	CCERRCLR	EDMA3CC Error Clear Register	Section 10.4.1.16
320h	EEVAL	Error Evaluate Register	Section 10.4.1.17
340h	DRAE0	DMA Region Access Enable Register for Region 0	Section 10.4.1.18
344h	DRAEH0	DMA Region Access Enable Register High for Region 0	Section 10.4.1.19
348h	DRAE1	DMA Region Access Enable Register for Region 1	Section 10.4.1.20
34Ch	DRAEH1	DMA Region Access Enable Register High for Region 1	Section 10.4.1.21
350h	DRAE2	DMA Region Access Enable Register for Region 2	Section 10.4.1.22
354h	DRAEH2	DMA Region Access Enable Register High for Region 2	Section 10.4.1.23
358h	DRAE3	DMA Region Access Enable Register for Region 3	Section 10.4.1.24
35Ch	DRAEH3	DMA Region Access Enable Register High for Region 3	Section 10.4.1.25
380h to 38Ch	QRAE_0 to QRAE_3	QDMA Region Access Enable Registers for Region 0-3	Section 10.4.1.26
400h to 43Ch	Q0E_0 to Q0E_15	Event Queue 0 Entry y Register	Section 10.4.1.27
440h to 47Ch	Q1E_0 to Q1E_15	Event Queue 1 Entry y Register	Section 10.4.1.28
480h to 4BCh	Q2E_0 to Q2E_15	Event Queue 2 Entry y Register	Section 10.4.1.29
600h to 608h	QSTAT_0 to QSTAT_2	Queue Status Registers 0-2	Section 10.4.1.30
620h	QWMTHRA	Queue Watermark Threshold A Register	Section 10.4.1.31
640h	CCSTAT	EDMA3CC Status Register	Section 10.4.1.32
800h	MPFAR	Memory Protection Fault Address Register	Section 10.4.1.33
804h	MPFSR	Memory Protection Fault Status Register	Section 10.4.1.34
808h	MPFCR	Memory Protection Fault Command Register	Section 10.4.1.35
80Ch	MPPAG	Memory Protection Page Attribute Register Global	Section 10.4.1.36
810h to 81Ch	MPPA_0 to MPPA_3	Memory Protection Page Attributes Registers	Section 10.4.1.37

Table 10-25. EDMA3CC Registers (continued)

Offset	Acronym	Register Name	Section
1000h	ER	Event Register	Section 10.4.1.38
1004h	ERH	Event Register High	Section 10.4.1.39
1008h	ECR	Event Clear Register	Section 10.4.1.40
100Ch	ECRH	Event Clear Register High	Section 10.4.1.41
1010h	ESR	Event Set Register	Section 10.4.1.42
1014h	ESRH	Event Set Register High	Section 10.4.1.43
1018h	CER	Chained Event Register	Section 10.4.1.44
101Ch	CERH	Chained Event Register High	Section 10.4.1.45
1020h	EER	Event Enable Register	Section 10.4.1.46
1024h	EERH	Event Enable Register High	Section 10.4.1.47
1028h	EECR	Event Enable Clear Register	Section 10.4.1.48
102Ch	EECRH	Event Enable Clear Register High	Section 10.4.1.49
1030h	EESR	Event Enable Set Register	Section 10.4.1.50
1034h	EESRH	Event Enable Set Register High	Section 10.4.1.51
1038h	SER	Secondary Event Register	Section 10.4.1.52
103Ch	SERH	Secondary Event Register High	Section 10.4.1.53
1040h	SECR	Secondary Event Clear Register	Section 10.4.1.54
1044h	SECRH	Secondary Event Clear Register High	Section 10.4.1.55
1050h	IER	Interrupt Enable Register	Section 10.4.1.56
1054h	IERH	Interrupt Enable Register High	Section 10.4.1.57
1058h	IECR	Interrupt Enable Clear Register	Section 10.4.1.58
105Ch	IECRH	Interrupt Enable Clear Register High	Section 10.4.1.59
1060h	IESR	Interrupt Enable Set Register	Section 10.4.1.60
1064h	IESRH	Interrupt Enable Set Register High	Section 10.4.1.61
1068h	IPR	Interrupt Pending Register	Section 10.4.1.62
106Ch	IPRH	Interrupt Pending Register High	Section 10.4.1.63
1070h	ICR	Interrupt Clear Register	Section 10.4.1.64
1074h	ICRH	Interrupt Clear Register High	Section 10.4.1.65
1078h	IEVAL	Interrupt Evaluate Register	Section 10.4.1.66
1080h	QER	QDMA Event Register	Section 10.4.1.67
1084h	QEER	QDMA Event Enable Register	Section 10.4.1.68
1088h	QEECR	QDMA Event Enable Clear Register	Section 10.4.1.69
108Ch	QEESR	QDMA Event Enable Set Register	Section 10.4.1.70
1090h	QSER	QDMA Secondary Event Register	Section 10.4.1.71
1094h	QSECR	QDMA Secondary Event Clear Register	Section 10.4.1.72

10.4.1.1 PID Register (Offset = 0h) [reset = 0h]

PID is shown in Figure 10-114 and described in Table 10-99.

Return to Summary Table.

The peripheral identification register (PID) uniquely identifies the EDMA3CC and the specific revision of the EDMA3CC.

Figure 10-42. PID Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID															
R-0h															

Table 10-26. PID Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	PID	R	0h	Peripheral identifier uniquely identifies the EDMA3CC and the specific revision of the EDMA3CC. Value 0 to FFFF FFFFh. Reset value for PID[31] to PID[16] is 4001h. Peripheral identifier uniquely identifies the EDMA3CC and the specific revision of the EDMA3CC. Value 0 to FFFF FFFFh. Reset value for PID[15] to PID[0] is 4C00h.

10.4.1.2 CCCFG Register (Offset = 4h) [reset = 03224445h]

CCCFG is shown in Figure 10-43 and described in Table 10-27.

Return to Summary Table.

The EDMA3CC configuration register (CCCFG) provides the features/resources for the EDMA3CC in a particular device.

Figure 10-43. CCCFG Register

31	30	29	28	27	26	25	24
RESERVED						MP_EXIST	CHMAP_EXIST
R-0h						R-1h	R-1h
23	22	21	20	19	18	17	16
RESERVED		NUM_REGN		RESERVED	NUM_EVQUE		
R-0h		R-2h		R-0h	R-2h		
15	14	13	12	11	10	9	8
RESERVED	NUM_PAENTRY			RESERVED	NUM_INTCH		
R-0h	R-4h			R-0h	R-4h		
7	6	5	4	3	2	1	0
RESERVED	NUM_QDMACH			RESERVED	NUM_DMACH		
R-0h	R-4h			R-0h	R-5h		

Table 10-27. CCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25	MP_EXIST	R	1h	Memory protection existence. 0h (R/W) = Reserved. 1h (R/W) = Memory protection logic included.
24	CHMAP_EXIST	R	1h	Channel mapping existence. 0h (R/W) = Reserved. 1h (R/W) = Channel mapping logic included.
23-22	RESERVED	R	0h	
21-20	NUM_REGN	R	2h	Number of MP and shadow regions. 0h (R/W) = Reserved. 1h (R/W) = Reserved 2h (R/W) = 4 regions. 3h (R/W) = Reserved.
19	RESERVED	R	0h	
18-16	NUM_EVQUE	R	2h	Number of queues/number of TCs. 0h (R/W) = Reserved. 1h (R/W) = Reserved. 2h (R/W) = 3 EDMA3TCs/Event Queues 3h (R/W) = Reserved from 3h to 7h. 7h (R/W) = Reserved.
15	RESERVED	R	0h	
14-12	NUM_PAENTRY	R	4h	Number of PaRAM sets. 0h (R/W) = Reserved from 0h to 3h. 3h (R/W) = Reserved 4h (R/W) = 256 PaRAM sets. 5h (R/W) = Reserved from 5h to 7h. 7h (R/W) = Reserved.
11	RESERVED	R	0h	

Table 10-27. CCCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-8	NUM_INTCH	R	4h	Number of interrupt channels. 0h (R/W) = Reserved from 0h to 3h. 3h (R/W) = Reserved. 4h (R/W) = 64 interrupt channels. 5h (R/W) = Reserved from 5h to 7. 7h (R/W) = Reserved.
7	RESERVED	R	0h	
6-4	NUM_QDMACH	R	4h	Number of QDMA channels. 0h (R/W) = Reserved from 0h to 3h. 3h (R/W) = Reserved. 4h (R/W) = 8 QDMA channels. 5h (R/W) = Reserved from 5h to 7. 7h (R/W) = Reserved.
3	RESERVED	R	0h	
2-0	NUM_DMACH	R	5h	Number of DMA channels. 0h (R/W) = Reserved from 0h to 4h. 4h (R/W) = Reserved. 5h (R/W) = 64 DMA channels. 6h (R/W) = Reserved. 7h (R/W) = Reserved.

10.4.1.3 SYSCONFIG Register (Offset = 10h) [reset = 8h]

SYSCONFIG is shown in Figure 20-4 and described in Table 20-7.

Return to Summary Table.

The EDMA3CC system configuration register is used for clock management configuration.

Figure 10-44. SYSCONFIG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="2">RESERVED</td><td colspan="2">IDLEMODE</td><td>RESERVED</td><td>RESERVED</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="2">R-0h</td><td colspan="2">R/W-2h</td><td>R-0h</td><td>R-0h</td></tr>
</table>

Table 10-28. SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5-4	RESERVED	R	0h	
3-2	IDLEMODE	R/W	2h	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state. 0h (R/W) = Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements. Backup mode, for debug only. 1h (R/W) = No-idle mode: local target never enters idle state. Backup mode, for debug only. 2h (R/W) = Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wakeup events. 3h (R/W) = Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state. Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented.
1	RESERVED	R	0h	
0	RESERVED	R	0h	

10.4.1.4 DCHMAP_0 to DCHMAP_63 Register (Offset = 100h to 1FCh) [reset = 0h]

DCHMAP_0 to DCHMAP_63 is shown in Figure 10-45 and described in Table 10-29.

Return to Summary Table.

Figure 10-45. DCHMAP_0 to DCHMAP_63 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="9">PAENTRY</td><td colspan="5">RESERVED</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="9">R/W-0h</td><td colspan="5">R-0h</td></tr>
</table>

Table 10-29. DCHMAP_0 to DCHMAP_63 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	
13-5	PAENTRY	R/W	0h	Points to the PaRAM set number for DMA channel n. Value 0 to FFh.
4-0	RESERVED	R	0h	

10.4.1.5 QCHMAP_0 to QCHMAP_7 Register (Offset = 200h to 21Ch) [reset = 0h]

QCHMAP_0 to QCHMAP_7 is shown in Figure 10-46 and described in Table 10-30.

Return to Summary Table.

Each QDMA channel in EDMA3CC can be associated with any PaRAM set available on the device. Furthermore, the specific trigger word (0-7) of the PaRAM set can be programmed. The PaRAM set association and trigger word for every QDMA channel register is configurable using the QDMA channel map register (QCHMAPn). At reset the QDMA channel map registers for all QDMA channels point to PaRAM set 0. If an application makes use of both a DMA channel that points to PaRAM set 0 and any QDMA channels, ensure that QCHMAP is programmed appropriately to point to a different PaRAM entry.

Figure 10-46. QCHMAP_0 to QCHMAP_7 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">PAENTRY</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="6">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">PAENTRY</td><td colspan="3">TRWORD</td><td colspan="2">RESERVED</td></tr>
<tr><td colspan="3">R/W-0h</td><td colspan="3">R/W-0h</td><td colspan="2">R-0h</td></tr>
</table>

Table 10-30. QCHMAP_0 to QCHMAP_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	
13-5	PAENTRY	R/W	0h	PAENTRY points to the PaRAM set number for QDMA channel . 0h (R/W) = Parameter entry 0 through 255, from 0 to FFh. 1h (R/W) = Reserved, from 100h to 1FFh. Always write 0 to this bit. Writes of 1 to this bit are not supported and attempts to do so may result in undefine behavior.
4-2	TRWORD	R/W	0h	Points to the specific trigger word of the PaRAM set defined by PAENTRY. A write to the trigger word results in a QDMA event being recognized.
1-0	RESERVED	R	0h	

10.4.1.6 DMAQNUM_0 to DMAQNUM_7 Register (Offset = 240h to 25Ch) [reset = 0h]

DMAQNUM_0 to DMAQNUM_7 is shown in Figure 10-47 and described in Table 10-31.

Return to Summary Table.

The DMA channel queue number register (DMAQNUMn) allows programmability of each of the 64 DMA channels in the EDMA3CC to submit its associated synchronization event to any event queue in the EDMA3CC. At reset, all channels point to event queue 0. Because the event queues in EDMA3CC have a fixed association to the transfer controllers, that is, Q0 TRs are submitted to TC0, Q1 TRs are submitted to TC1, etc., by programming DMAQNUM for a particular DMA channel also dictates which transfer controller is utilized for the data movement (or which EDMA3TC receives the TR request).

Figure 10-47. DMAQNUM_0 to DMAQNUM_7 Register

31	30	29	28	27	26	25	24
RESERVED	E7			RESERVED	E6		
R-0h	R/W-0h			R-0h	R/W-0h		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED	E5			RESERVED	E4		
R-0h	R/W-0h			R-0h	R/W-0h		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED	E3			RESERVED	E2		
R-0h	R/W-0h			R-0h	R/W-0h		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	E1			RESERVED	E0		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 10-31. DMAQNUM_0 to DMAQNUM_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	E7	R/W	0h	DMA queue number. Contains the event queue number to be used for the corresponding DMA channel. Programming DMAQNUM for an event queue number to a value more then the number of queues available in the EDMA3CC results in undefined behavior. On DMAQNUM0, E[30] to E[28] is E7. On DMAQNUM1, E[30] to E[28] is E15. On DMAQNUM2, E[30] to E[28] is E23. On DMAQNUM3, E[30] to E[28] is E31. On DMAQNUM4, E[30] to E[28] is E39. On DMAQNUM5, E[30] to E[28] is E47. On DMAQNUM6, E[30] to E[28] is E55. On DMAQNUM7, E[30] to E[28] is E63. 0h (R/W) = Event n is queued on Q0. 1h (R/W) = Event n is queued on Q1. 2h (R/W) = Event n is queued on Q2. 3h (R/W) = Reserved, from 3h to 7h. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. 7h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
27	RESERVED	R	0h	

Table 10-31. DMAQNUM_0 to DMAQNUM_7 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
26-24	E6	R/W	0h	DMA queue number. Contains the event queue number to be used for the corresponding DMA channel. Programming DMAQNUM for an event queue number to a value more then the number of queues available in the EDMA3CC results in undefined behavior. On DMAQNUM0, E[26] to E[24] is E6. On DMAQNUM1, E[26] to E[24] is E14. On DMAQNUM2, E[26] to E[24] is E22. On DMAQNUM3, E[26] to E[24] is E30. On DMAQNUM4, E[26] to E[24] is E38. On DMAQNUM5, E[26] to E[24] is E46. On DMAQNUM6, E[26] to E[24] is E54. On DMAQNUM7, E[26] to E[24] is E62. 0h (R/W) = Event n is queued on Q0. 1h (R/W) = Event n is queued on Q1. 2h (R/W) = Event n is queued on Q2. 3h (R/W) = Reserved, from 3h to 7h. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. 7h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
23	RESERVED	R	0h	
22-20	E5	R/W	0h	DMA queue number. Contains the event queue number to be used for the corresponding DMA channel. Programming DMAQNUM for an event queue number to a value more then the number of queues available in the EDMA3CC results in undefined behavior. On DMAQNUM0, E[22] to E[20] is E5. On DMAQNUM1, E[22] to E[20] is E13. On DMAQNUM2, E[22] to E[20] is E21. On DMAQNUM3, E[22] to E[20] is E29. On DMAQNUM4, E[22] to E[20] is E37. On DMAQNUM5, E[22] to E[20] is E45. On DMAQNUM6, E[22] to E[20] is E53. On DMAQNUM7, E[22] to E[20] is E61. 0h (R/W) = Event n is queued on Q0. 1h (R/W) = Event n is queued on Q1. 2h (R/W) = Event n is queued on Q2. 3h (R/W) = Reserved, from 3h to 7h. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. 7h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
19	RESERVED	R	0h	

Table 10-31. DMAQNUM_0 to DMAQNUM_7 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18-16	E4	R/W	0h	DMA queue number. Contains the event queue number to be used for the corresponding DMA channel. Programming DMAQNUM for an event queue number to a value more then the number of queues available in the EDMA3CC results in undefined behavior. On DMAQNUM0, E[18] to E[16] is E4. On DMAQNUM1, E[18] to E[16] is E12. On DMAQNUM2, E[18] to E[16] is E20. On DMAQNUM3, E[18] to E[16] is E28. On DMAQNUM4, E[18] to E[16] is E36. On DMAQNUM5, E[18] to E[16] is E44. On DMAQNUM6, E[18] to E[16] is E52. On DMAQNUM7, E[18] to E[16] is E60. 0h (R/W) = Event n is queued on Q0. 1h (R/W) = Event n is queued on Q1. 2h (R/W) = Event n is queued on Q2. 3h (R/W) = Reserved, from 3h to 7h. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. 7h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
15	RESERVED	R	0h	
14-12	E3	R/W	0h	DMA queue number. Contains the event queue number to be used for the corresponding DMA channel. Programming DMAQNUM for an event queue number to a value more then the number of queues available in the EDMA3CC results in undefined behavior. On DMAQNUM0, E[14] to E[12] is E3. On DMAQNUM1, E[14] to E[12] is E11. On DMAQNUM2, E[14] to E[12] is E19. On DMAQNUM3, E[14] to E[12] is E27. On DMAQNUM4, E[14] to E[12] is E35. On DMAQNUM5, E[14] to E[12] is E43. On DMAQNUM6, E[14] to E[12] is E51. On DMAQNUM7, E[14] to E[12] is E59. 0h (R/W) = Event n is queued on Q0. 1h (R/W) = Event n is queued on Q1. 2h (R/W) = Event n is queued on Q2. 3h (R/W) = Reserved, from 3h to 7h. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. 7h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
11	RESERVED	R	0h	

Table 10-31. DMAQNUM_0 to DMAQNUM_7 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-8	E2	R/W	0h	DMA queue number. Contains the event queue number to be used for the corresponding DMA channel. Programming DMAQNUM for an event queue number to a value more then the number of queues available in the EDMA3CC results in undefined behavior. On DMAQNUM0, E[10] to E[8] is E2. On DMAQNUM1, E[10] to E[8] is E10. On DMAQNUM2, E[10] to E[8] is E18. On DMAQNUM3, E[10] to E[8] is E26. On DMAQNUM4, E[10] to E[8] is E34. On DMAQNUM5, E[10] to E[8] is E42. On DMAQNUM6, E[10] to E[8] is E50. On DMAQNUM7, E[10] to E[8] is E58. 0h (R/W) = Event n is queued on Q0. 1h (R/W) = Event n is queued on Q1. 2h (R/W) = Event n is queued on Q2. 3h (R/W) = Reserved, from 3h to 7h. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. 7h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7	RESERVED	R	0h	
6-4	E1	R/W	0h	DMA queue number. Contains the event queue number to be used for the corresponding DMA channel. Programming DMAQNUM for an event queue number to a value more then the number of queues available in the EDMA3CC results in undefined behavior. On DMAQNUM0, E[6] to E[4] is E1. On DMAQNUM1, E[6] to E[4] is E9. On DMAQNUM2, E[6] to E[4] is E17. On DMAQNUM3, E[6] to E[4] is E25. On DMAQNUM4, E[6] to E[4] is E33. On DMAQNUM5, E[6] to E[4] is E41. On DMAQNUM6, E[6] to E[4] is E49. On DMAQNUM7, E[6] to E[4] is E57. 0h (R/W) = Event n is queued on Q0. 1h (R/W) = Event n is queued on Q1. 2h (R/W) = Event n is queued on Q2. 3h (R/W) = Reserved, from 3h to 7h. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. 7h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
3	RESERVED	R	0h	

Table 10-31. DMAQNUM_0 to DMAQNUM_7 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	E0	R/W	0h	DMA queue number. Contains the event queue number to be used for the corresponding DMA channel. Programming DMAQNUM for an event queue number to a value more then the number of queues available in the EDMA3CC results in undefined behavior. On DMAQNUM0, E[2] to E[0] is E0. On DMAQNUM1, E[2] to E[0] is E8. On DMAQNUM2, E[2] to E[0] is E16. On DMAQNUM3, E[2] to E[0] is E24. On DMAQNUM4, E[2] to E[0] is E32. On DMAQNUM5, E[2] to E[0] is E40. On DMAQNUM6, E[2] to E[0] is E48. On DMAQNUM7, E[2] to E[0] is E56. 0h (R/W) = Event n is queued on Q0. 1h (R/W) = Event n is queued on Q1. 2h (R/W) = Event n is queued on Q2. 3h (R/W) = Reserved, from 3h to 7h. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. 7h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.

10.4.1.7 QDMAQNUM Register (Offset = 260h) [reset = 0h]

QDMAQNUM is shown in Figure 10-48 and described in Table 10-32.

Return to Summary Table.

The QDMA channel queue number register (QDMAQNUMn) is used to program all the QDMA channels in the EDMA3CC to submit the associated QDMA event to any of the event queues in the EDMA3CC.

Figure 10-48. QDMAQNUM Register

31	30	29	28	27	26	25	24
RESERVED	E7			RESERVED	E6		
R-0h	R/W-0h			R-0h	R/W-0h		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED	E5			RESERVED	E4		
R-0h	R/W-0h			R-0h	R/W-0h		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED	E3			RESERVED	E2		
R-0h	R/W-0h			R-0h	R/W-0h		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	E1			RESERVED	E0		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 10-32. QDMAQNUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	E7	R/W	0h	QDMA queue number. Contains the event queue number to be used for the corresponding QDMA channel. 0h (R/W) = Event n is queued on Q0. 1h (R/W) = Event n is queued on Q1. 2h (R/W) = Event n is queued on Q2. 3h (R/W) = Reserved, from 3h to 7h. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. 7h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
27	RESERVED	R	0h	
26-24	E6	R/W	0h	QDMA queue number. Contains the event queue number to be used for the corresponding QDMA channel. 0h (R/W) = Event n is queued on Q0. 1h (R/W) = Event n is queued on Q1. 2h (R/W) = Event n is queued on Q2. 3h (R/W) = Reserved, from 3h to 7h. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. 7h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
23	RESERVED	R	0h	

Table 10-32. QDMAQNUM Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22-20	E5	R/W	0h	QDMA queue number. Contains the event queue number to be used for the corresponding QDMA channel. 0h (R/W) = Event n is queued on Q0. 1h (R/W) = Event n is queued on Q1. 2h (R/W) = Event n is queued on Q2. 3h (R/W) = Reserved, from 3h to 7h. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. 7h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
19	RESERVED	R	0h	
18-16	E4	R/W	0h	QDMA queue number. Contains the event queue number to be used for the corresponding QDMA channel. 0h (R/W) = Event n is queued on Q0. 1h (R/W) = Event n is queued on Q1. 2h (R/W) = Event n is queued on Q2. 3h (R/W) = Reserved, from 3h to 7h. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. 7h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
15	RESERVED	R	0h	
14-12	E3	R/W	0h	QDMA queue number. Contains the event queue number to be used for the corresponding QDMA channel. 0h (R/W) = Event n is queued on Q0. 1h (R/W) = Event n is queued on Q1. 2h (R/W) = Event n is queued on Q2. 3h (R/W) = Reserved, from 3h to 7h. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. 7h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
11	RESERVED	R	0h	
10-8	E2	R/W	0h	QDMA queue number. Contains the event queue number to be used for the corresponding QDMA channel. 0h (R/W) = Event n is queued on Q0. 1h (R/W) = Event n is queued on Q1. 2h (R/W) = Event n is queued on Q2. 3h (R/W) = Reserved, from 3h to 7h. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. 7h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7	RESERVED	R	0h	

Table 10-32. QDMAQNUM Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6-4	E1	R/W	0h	QDMA queue number. Contains the event queue number to be used for the corresponding QDMA channel. 0h (R/W) = Event n is queued on Q0. 1h (R/W) = Event n is queued on Q1. 2h (R/W) = Event n is queued on Q2. 3h (R/W) = Reserved, from 3h to 7h. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. 7h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
3	RESERVED	R	0h	
2-0	E0	R/W	0h	QDMA queue number. Contains the event queue number to be used for the corresponding QDMA channel. 0h (R/W) = Event n is queued on Q0. 1h (R/W) = Event n is queued on Q1. 2h (R/W) = Event n is queued on Q2. 3h (R/W) = Reserved, from 3h to 7h. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. 7h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.

10.4.1.8 QUEPRI Register (Offset = 284h) [reset = 777h]

QUEPRI is shown in Figure 10-49 and described in Table 10-33.

Return to Summary Table.

The queue priority register (QUEPRI) allows you to change the priority of the individual queues and the priority of the transfer request (TR) associated with the events queued in the queue. Because the queue to EDMA3TC mapping is fixed, programming QUEPRI essentially governs the priority of the associated transfer controller(s) read/write commands with respect to the other bus masters in the device. You can modify the EDMA3TC priority to obtain the desired system performance.

Figure 10-49. QUEPRI Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="5">RESERVED</td><td colspan="3">PRIQ2</td></tr>
<tr><td colspan="5">R-0h</td><td colspan="3">R/W-7h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>RESERVED</td><td colspan="3">PRIQ1</td><td>RESERVED</td><td colspan="3">PRIQ0</td></tr>
<tr><td>R-0h</td><td colspan="3">R/W-7h</td><td>R-0h</td><td colspan="3">R/W-7h</td></tr>
</table>

Table 10-33. QUEPRI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10-8	PRIQ2	R/W	7h	Priority level for queue 2. Dictates the priority level used by TC2 relative to other masters in the device. A value of 0 means highest priority and a value of 7 means lowest priority.
7	RESERVED	R	0h	
6-4	PRIQ1	R/W	7h	Priority level for queue 1. Dictates the priority level used by TC1 relative to other masters in the device. A value of 0 means highest priority and a value of 7 means lowest priority.
3	RESERVED	R	0h	
2-0	PRIQ0	R/W	7h	Priority level for queue 0. Dictates the priority level used by TC0 relative to other masters in the device. A value of 0 means highest priority and a value of 7 means lowest priority.

10.4.1.9 EMR Register (Offset = 300h) [reset = 0h]

EMR is shown in Figure 10-50 and described in Table 10-34.

Return to Summary Table.

For a particular DMA channel, if a second event is received prior to the first event getting cleared/serviced, the bit corresponding to that channel is set/asserted in the event missed registers (EMR/EMRH). All trigger types are treated individually, that is, manual triggered (ESR/ESRH), chain triggered (CER/CERH), and event triggered (ER/ERH) are all treated separately. The EMR/EMRH bits for a channel are also set if an event on that channel encounters a NULL entry (or a NULL TR is serviced). If any EMR/EMRH bit is set (and all errors, including bits in other error registers (QEMR, CCERR) were previously cleared), the EDMA3CC generates an error interrupt. For details on EDMA3CC error interrupt generation, see Error Interrupts. This register is part of a set of registers that provide information on missed DMA and/or QDMA events, and instances when event queue thresholds are exceeded. If any of the bits in these registers is set, it results in the EDMA3CC generating an error interrupt.

Figure 10-50. EMR Register

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="32">En</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 10-34. EMR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R	0h	Channel 0 to 31 event missed. En is cleared by writing a 1 to the corresponding bit in the event missed clear register (EMCR). 0h (R/W) = No missed event. 1h (R/W) = Missed event occurred.

10.4.1.10 EMRH Register (Offset = 304h) [reset = 0h]

EMRH is shown in Figure 10-51 and described in Table 10-35.

Return to Summary Table.

For a particular DMA channel, if a second event is received prior to the first event getting cleared/serviced, the bit corresponding to that channel is set/asserted in the event missed registers (EMR/EMRH). All trigger types are treated individually, that is, manual triggered (ESR/ESRH), chain triggered (CER/CERH), and event triggered (ER/ERH) are all treated separately. The EMR/EMRH bits for a channel are also set if an event on that channel encounters a NULL entry (or a NULL TR is serviced). If any EMR/EMRH bit is set (and all errors, including bits in other error registers (QEMR, CCERR) were previously cleared), the EDMA3CC generates an error interrupt. For details on EDMA3CC error interrupt generation, see Error Interrupts. This register is part of a set of registers that provide information on missed DMA and/or QDMA events, and instances when event queue thresholds are exceeded. If any of the bits in these registers is set, it results in the EDMA3CC generating an error interrupt.

Figure 10-51. EMRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
En																															
R-0h																															

Table 10-35. EMRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R	0h	Channel 32 to 63 event missed. En is cleared by writing a 1 to the corresponding bit in the event missed clear register high (EMCRH). 0h (R/W) = No missed event. 1h (R/W) = Missed event occurred.

10.4.1.11 EMCR Register (Offset = 308h) [reset = 0h]

EMCR is shown in Figure 10-52 and described in Table 10-36.

Return to Summary Table.

Once a missed event is posted in the event missed registers (EMR/EMRH), the bit remains set and you need to clear the set bit(s). This is done by way of CPU writes to the event missed clear registers (EMCR/EMCRH). Writing a 1 to any of the bits clears the corresponding missed event (bit) in EMR/EMRH; writing a 0 has no effect. This register is part of a set of registers that provide information on missed DMA and/or QDMA events, and instances when event queue thresholds are exceeded. If any of the bits in these registers is set, it results in the EDMA3CC generating an error interrupt.

Figure 10-52. EMCR Register

31-0
En
W-0h

Table 10-36. EMCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	W	0h	Event missed 0 to 31 clear. All error bits must be cleared before additional error interrupts will be asserted by the EDMA3CC. 0h (R/W) = No effect. 1h (R/W) = Corresponding missed event bit in the event missed register (EMR) is cleared (En = 0).

10.4.1.12 EMCRH Register (Offset = 30Ch) [reset = 0h]

EMCRH is shown in Figure 10-53 and described in Table 10-37.

Return to Summary Table.

Once a missed event is posted in the event missed registers (EMR/EMRH), the bit remains set and you need to clear the set bit(s). This is done by way of CPU writes to the event missed clear registers (EMCR/EMCRH). Writing a 1 to any of the bits clears the corresponding missed event (bit) in EMR/EMRH; writing a 0 has no effect. This register is part of a set of registers that provide information on missed DMA and/or QDMA events, and instances when event queue thresholds are exceeded. If any of the bits in these registers is set, it results in the EDMA3CC generating an error interrupt.

Figure 10-53. EMCRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
En																															
W-0h																															

Table 10-37. EMCRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	W	0h	Event missed 32 to 63 clear. All error bits must be cleared before additional error interrupts will be asserted by the EDMA3CC. 0h (R/W) = No effect. 1h (R/W) = Corresponding missed event bit in the event missed register high (EMRH) is cleared (En = 0).

10.4.1.13 QEMR Register (Offset = 310h) [reset = 0h]

QEMR is shown in Figure 10-54 and described in Table 10-38.

Return to Summary Table.

For a particular QDMA channel, if two QDMA events are detected without the first event getting cleared/serviced, the bit corresponding to that channel is set/asserted in the QDMA event missed register (QEMR). The QEMR bits for a channel are also set if a QDMA event on the channel encounters a NULL entry (or a NULL TR is serviced). If any QEMR bit is set (and all errors, including bits in other error registers (EMR/EMRH, CCERR) were previously cleared), the EDMA3CC generates an error interrupt. For details on EDMA3CC error interrupt generation, see Error Interrupts. This register is part of a set of registers that provide information on missed DMA and/or QDMA events, and instances when event queue thresholds are exceeded. If any of the bits in these registers is set, it results in the EDMA3CC generating an error interrupt.

Figure 10-54. QEMR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								En							
R-0h																								R-0h							

Table 10-38. QEMR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	En	R	0h	Channel 0 to 7 QDMA event missed. En is cleared by writing a 1 to the corresponding bit in the QDMA event missed clear register (QEMCR). 0h (R/W) = No missed event. 1h (R/W) = Missed event occurred.

10.4.1.14 QEMCR Register (Offset = 314h) [reset = 0h]

QEMCR is shown in Figure 10-55 and described in Table 10-39.

Return to Summary Table.

Once a missed event is posted in the QDMA event missed registers (QEMR), the bit remains set and you need to clear the set bit(s). This is done by way of CPU writes to the QDMA event missed clear registers (QEMCR). Writing a 1 to any of the bits clears the corresponding missed event (bit) in QEMR; writing a 0 has no effect. This register is part of a set of registers that provide information on missed DMA and/or QDMA events, and instances when event queue thresholds are exceeded. If any of the bits in these registers is set, it results in the EDMA3CC generating an error interrupt.

Figure 10-55. QEMCR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								En							
R-0h																								W-0h							

Table 10-39. QEMCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	En	W	0h	QDMA event missed clear. All error bits must be cleared before additional error interrupts will be asserted by the EDMA3CC. 0h (R/W) = No effect. 1h (R/W) = Corresponding missed event bit in the QDMA event missed register (QEMR) is cleared (En= 0).

10.4.1.15 CCERR Register (Offset = 318h) [reset = 0h]

CCERR is shown in Figure 10-56 and described in Table 10-40.

Return to Summary Table.

The EDMA3CC error register (CCERR) indicates whether or not at any instant of time the number of events queued up in any of the event queues exceeds or equals the threshold/watermark value that is set in the queue watermark threshold register (QWMTHRA). Additionally, CCERR also indicates if when the number of outstanding TRs that have been programmed to return transfer completion code (TRs which have the TCINTEN or TCCHEN bit in OPT set) to the EDMA3CC has exceeded the maximum allowed value of 63. If any bit in CCERR is set (and all errors, including bits in other error registers (EMR/EMRH, QEMR) were previously cleared), the EDMA3CC generates an error interrupt. For details on EDMA3CC error interrupt generation, see Error Interrupts. Once the error bits are set in CCERR, they can only be cleared by writing to the corresponding bits in the EDMA3CC error clear register (CCERRCLR). This register is part of a set of registers that provide information on missed DMA and/or QDMA events, and instances when event queue thresholds are exceeded. If any of the bits in these registers is set, it results in the EDMA3CC generating an error interrupt.

Figure 10-56. CCERR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							TCCERR
R-0h							R-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					QTHRXCD2	QTHRXCD1	QTHRXCD0
R-0h					R-0h	R-0h	R-0h

Table 10-40. CCERR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	TCCERR	R	0h	Transfer completion code error. TCCERR is cleared by writing a 1 to the corresponding bit in the EDMA3CC error clear register (CCERRCLR). 0h (R/W) = Total number of allowed TCCs outstanding has not been reached. 1h (R/W) = Total number of allowed TCCs has been reached.
15-3	RESERVED	R	0h	
2	QTHRXCD2	R	0h	Queue threshold error for queue 2. QTHRXCD2 is cleared by writing a 1 to the corresponding bit in the EDMA3CC error clear register (CCERRCLR). 0h (R/W) = Watermark/threshold has not been exceeded. 1h (R/W) = Watermark/threshold has been exceeded.
1	QTHRXCD1	R	0h	Queue threshold error for queue 1 . QTHRXCD1 is cleared by writing a 1 to the corresponding bit in the EDMA3CC error clear register (CCERRCLR). 0h (R/W) = Watermark/threshold has not been exceeded. 1h (R/W) = Watermark/threshold has been exceeded.
0	QTHRXCD0	R	0h	Queue threshold error for queue 0. QTHRXCD0 is cleared by writing a 1 to the corresponding bit in the EDMA3CC error clear register (CCERRCLR). 0h (R/W) = Watermark/threshold has not been exceeded. 1h (R/W) = Watermark/threshold has been exceeded.

10.4.1.16 CCERRCLR Register (Offset = 31Ch) [reset = 0h]

CCERRCLR is shown in Figure 10-57 and described in Table 10-41.

Return to Summary Table.

The EDMA3CC error clear register (CCERRCLR) is used to clear any error bits that are set in the EDMA3CC error register (CCERR). In addition, CCERRCLR also clears the values of some bit fields in the queue status registers (QSTATn) associated with a particular event queue. Writing a 1 to any of the bits clears the corresponding bit in CCERR; writing a 0 has no effect. This register is part of a set of registers that provide information on missed DMA and/or QDMA events, and instances when event queue thresholds are exceeded. If any of the bits in these registers is set, it results in the EDMA3CC generating an error interrupt.

Figure 10-57. CCERRCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							TCCERR
R-0h							W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				QTHRXCD3	QTHRXCD2	QTHRXCD1	QTHRXCD0
R-0h				W-0h	W-0h	W-0h	W-0h

Table 10-41. CCERRCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	TCCERR	W	0h	Transfer completion code error clear. 0h (R/W) = No effect. 1h (R/W) = Clears the TCCERR bit in the EDMA3CC error register (CCERR).
15-4	RESERVED	R	0h	
3	QTHRXCD3	W	0h	Queue threshold error clear for queue 3. 0h (R/W) = No effect. 1h (R/W) = Clears the QTHRXCD3 bit in the EDMA3CC error register (CCERR) and the WM and THRXCD bits in the queue status register 3 (QSTAT3).
2	QTHRXCD2	W	0h	Queue threshold error clear for queue 2. 0h (R/W) = No effect. 1h (R/W) = Clears the QTHRXCD2 bit in the EDMA3CC error register (CCERR) and the WM and THRXCD bits in the queue status register 2 (QSTAT2).
1	QTHRXCD1	W	0h	Queue threshold error clear for queue 1. 0h (R/W) = No effect. 1h (R/W) = Clears the QTHRXCD1 bit in the EDMA3CC error register (CCERR) and the WM and THRXCD bits in the queue status register 1 (QSTAT1).
0	QTHRXCD0	W	0h	Queue threshold error clear for queue 0. 0h (R/W) = No effect. 1h (R/W) = Clears the QTHRXCD0 bit in the EDMA3CC error register (CCERR) and the WM and THRXCD bits in the queue status register 0 (QSTAT0).

10.4.1.17 EEVAL Register (Offset = 320h) [reset = 0h]

EEVAL is shown in Figure 10-58 and described in Table 10-42.

Return to Summary Table.

The EDMA3CC error interrupt is asserted whenever an error bit is set in any of the error registers (EMR/EMRH, QEMR, and CCERR). For subsequent error bits that get set, the EDMA3CC error interrupt is reasserted only when transitioning from an all the error bits cleared to at least one error bit is set . Alternatively, a CPU write of 1 to the EVAL bit in the error evaluation register (EEVAL) results in reasserting the EDMA3CC error interrupt, if there are any outstanding error bits set due to subsequent error conditions. Writes of 0 have no effect. This register is part of a set of registers that provide information on missed DMA and/or QDMA events, and instances when event queue thresholds are exceeded. If any of the bits in these registers is set, it results in the EDMA3CC generating an error interrupt.

Figure 10-58. EEVAL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	EVAL
R-0h						R-0h	W-0h

Table 10-42. EEVAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	RESERVED	R	0h	
0	EVAL	W	0h	Error interrupt evaluate. 0h (R/W) = No effect. 1h (R/W) = Write 1 to clear interrupts when all error registers have been cleared. EDMA3CC error interrupt will remain if any errors have not been cleared in any of the error registers (EMR/EMRH, CCERR, QEMR)

10.4.1.18 DRAE0 Register (Offset = 340h) [reset = 0h]

DRAE0 is shown in Figure 10-59 and described in Table 10-43.

Return to Summary Table.

The DMA region access enable register for shadow region 0 is programmed to allow or disallow read/write accesses on a bit-by-bit bases for all DMA registers in the shadow region 0 view of the DMA channel registers. Additionally, the DRAE0 configuration determines completion of which DMA channels will result in assertion of the shadow region 0 DMA completion interrupt. The DRAE registers are part of the group of the region access enable registers, which includes DRAEm and QRAEm. Where m is the number of shadow regions in the EDMA3CC memory map for a device. You can configure these registers to assign ownership of DMA/QDMA channels to a particular shadow region.

Figure 10-59. DRAE0 Register

31-0
En
R/W-0h

Table 10-43. DRAE0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R/W	0h	DMA region access enable for bit 31 to 0 in region 0. 0h (R/W) = Accesses via region 0 address space to bit 31 to 0 in any DMA channel register are not allowed. Reads return 0 on bit n and writes do not modify the state of bit 31 to 0. Enabled interrupt bits for bit n do not contribute to the generation of a transfer completion interrupt for shadow region 0. 1h (R/W) = Accesses via region 0 address space to bit 31 to 0 in any DMA channel register are allowed. Reads return the value from bit n and writes modify the state of bit 31 to 0. Enabled interrupt bits for bit n contribute to the generation of a transfer completion interrupt for shadow region 0.

10.4.1.19 DRAEH0 Register (Offset = 344h) [reset = 0h]

DRAEH0 is shown in Figure 10-60 and described in Table 10-44.

Return to Summary Table.

The DMA region access enable register for shadow region 0 is programmed to allow or disallow read/write accesses on a bit-by-bit bases for all DMA registers in the shadow region 0 view of the DMA channel registers. Additionally, the DRAE0 configuration determines completion of which DMA channels will result in assertion of the shadow region 0 DMA completion interrupt. The DRAE registers are part of the group of the region access enable registers, which includes DRAEm and QRAEm. Where m is the number of shadow regions in the EDMA3CC memory map for a device. You can configure these registers to assign ownership of DMA/QDMA channels to a particular shadow region.

Figure 10-60. DRAEH0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
En																															
R/W-0h																															

Table 10-44. DRAEH0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R/W	0h	DMA region access enable for bit 63 to 32 in region 0. 0h (R/W) = Accesses via region 0 address space to bit 63 to 32 in any DMA channel register are not allowed. Reads return 0 on bit n and writes do not modify the state of bit 63 to 32. Enabled interrupt bits for bit 31 to 0 do not contribute to the generation of a transfer completion interrupt for shadow region 0. 1h (R/W) = Accesses via region 0 address space to bit 63 to 32 in any DMA channel register are allowed. Reads return the value from bit n and writes modify the state of bit 63 to 32. Enabled interrupt bits for bit 31 to 0 contribute to the generation of a transfer completion interrupt for shadow region 0.

10.4.1.20 DRAE1 Register (Offset = 348h) [reset = 0h]

DRAE1 is shown in Figure 10-61 and described in Table 10-45.

Return to Summary Table.

The DMA region access enable register for shadow region 1 is programmed to allow or disallow read/write accesses on a bit-by-bit bases for all DMA registers in the shadow region 1 view of the DMA channel registers. Additionally, the DRAE1 configuration determines completion of which DMA channels will result in assertion of the shadow region 1 DMA completion interrupt. The DRAE registers are part of the group of the region access enable registers, which includes DRAEm and QRAEm. Where m is the number of shadow regions in the EDMA3CC memory map for a device. You can configure these registers to assign ownership of DMA/QDMA channels to a particular shadow region.

Figure 10-61. DRAE1 Register

31–0
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
En
R/W-0h

Table 10-45. DRAE1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R/W	0h	DMA region access enable for bit 31 to 0 in region 1. 0h (R/W) = Accesses via region 1 address space to bit 31 to 0 in any DMA channel register are not allowed. Reads return 0 on bit n and writes do not modify the state of bit 31 to 0. Enabled interrupt bits for bit n do not contribute to the generation of a transfer completion interrupt for shadow region 1. 1h (R/W) = Accesses via region 1 address space to bit 31 to 0 in any DMA channel register are allowed. Reads return the value from bit n and writes modify the state of bit 31 to 0. Enabled interrupt bits for bit n contribute to the generation of a transfer completion interrupt for shadow region 1.

10.4.1.21 DRAEH1 Register (Offset = 34Ch) [reset = 0h]

DRAEH1 is shown in Figure 10-62 and described in Table 10-46.

Return to Summary Table.

The DMA region access enable register for shadow region 1 is programmed to allow or disallow read/write accesses on a bit-by-bit bases for all DMA registers in the shadow region 1 view of the DMA channel registers. Additionally, the DRAE1 configuration determines completion of which DMA channels will result in assertion of the shadow region 1 DMA completion interrupt. The DRAE registers are part of the group of the region access enable registers, which includes DRAEm and QRAEm. Where m is the number of shadow regions in the EDMA3CC memory map for a device. You can configure these registers to assign ownership of DMA/QDMA channels to a particular shadow region.

Figure 10-62. DRAEH1 Register

31-0
En
R/W-0h

Table 10-46. DRAEH1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R/W	0h	DMA region access enable for bit 63 to 32 in region 1. 0h (R/W) = Accesses via region 1 address space to bit 63 to 32 in any DMA channel register are not allowed. Reads return 0 on bit n and writes do not modify the state of bit 63 to 32. Enabled interrupt bits for bit 31 to 0 do not contribute to the generation of a transfer completion interrupt for shadow region 1. 1h (R/W) = Accesses via region 1 address space to bit 63 to 32 in any DMA channel register are allowed. Reads return the value from bit n and writes modify the state of bit 63 to 32. Enabled interrupt bits for bit 31 to 0 contribute to the generation of a transfer completion interrupt for shadow region 1.

10.4.1.22 DRAE2 Register (Offset = 350h) [reset = 0h]

DRAE2 is shown in Figure 10-63 and described in Table 10-47.

Return to Summary Table.

The DMA region access enable register for shadow region 2 is programmed to allow or disallow read/write accesses on a bit-by-bit bases for all DMA registers in the shadow region 2 view of the DMA channel registers. Additionally, the DRAE2 configuration determines completion of which DMA channels will result in assertion of the shadow region 2 DMA completion interrupt. The DRAE registers are part of the group of the region access enable registers, which includes DRAEm and QRAEm. Where m is the number of shadow regions in the EDMA3CC memory map for a device. You can configure these registers to assign ownership of DMA/QDMA channels to a particular shadow region.

Figure 10-63. DRAE2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
En																															
R/W-0h																															

Table 10-47. DRAE2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R/W	0h	DMA region access enable for bit 31 to 0 in region 2. 0h (R/W) = Accesses via region 2 address space to bit 31 to 0 in any DMA channel register are not allowed. Reads return 0 on bit n and writes do not modify the state of bit 31 to 0. Enabled interrupt bits for bit n do not contribute to the generation of a transfer completion interrupt for shadow region 2. 1h (R/W) = Accesses via region 2 address space to bit 31 to 0 in any DMA channel register are allowed. Reads return the value from bit n and writes modify the state of bit 31 to 0. Enabled interrupt bits for bit n contribute to the generation of a transfer completion interrupt for shadow region 2.

10.4.1.23 DRAEH2 Register (Offset = 354h) [reset = 0h]

DRAEH2 is shown in Figure 10-64 and described in Table 10-48.

Return to Summary Table.

The DMA region access enable register for shadow region 2 is programmed to allow or disallow read/write accesses on a bit-by-bit bases for all DMA registers in the shadow region 2 view of the DMA channel registers. Additionally, the DRAE2 configuration determines completion of which DMA channels will result in assertion of the shadow region 2 DMA completion interrupt. The DRAE registers are part of the group of the region access enable registers, which includes DRAEm and QRAEm. Where m is the number of shadow regions in the EDMA3CC memory map for a device. You can configure these registers to assign ownership of DMA/QDMA channels to a particular shadow region.

Figure 10-64. DRAEH2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
En																															
R/W-0h																															

Table 10-48. DRAEH2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R/W	0h	DMA region access enable for bit 63 to 32 in region 2. 0h (R/W) = Accesses via region 2 address space to bit 63 to 32 in any DMA channel register are not allowed. Reads return 0 on bit n and writes do not modify the state of bit 63 to 32. Enabled interrupt bits for bit 31 to 0 do not contribute to the generation of a transfer completion interrupt for shadow region 2. 1h (R/W) = Accesses via region 2 address space to bit 63 to 32 in any DMA channel register are allowed. Reads return the value from bit n and writes modify the state of bit 63 to 32. Enabled interrupt bits for bit 31 to 0 contribute to the generation of a transfer completion interrupt for shadow region 2.

10.4.1.24 DRAE3 Register (Offset = 358h) [reset = 0h]

DRAE3 is shown in Figure 10-65 and described in Table 10-49.

Return to Summary Table.

The DMA region access enable register for shadow region 3 is programmed to allow or disallow read/write accesses on a bit-by-bit bases for all DMA registers in the shadow region 3 view of the DMA channel registers. Additionally, the DRAE3 configuration determines completion of which DMA channels will result in assertion of the shadow region 3 DMA completion interrupt. The DRAE registers are part of the group of the region access enable registers, which includes DRAEm and QRAEm. Where m is the number of shadow regions in the EDMA3CC memory map for a device. You can configure these registers to assign ownership of DMA/QDMA channels to a particular shadow region.

Figure 10-65. DRAE3 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">En</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 10-49. DRAE3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R/W	0h	DMA region access enable for bit 31 to 0 in region 3. 0h (R/W) = Accesses via region 3 address space to bit 31 to 0 in any DMA channel register are not allowed. Reads return 0 on bit n and writes do not modify the state of bit 31 to 0. Enabled interrupt bits for bit n do not contribute to the generation of a transfer completion interrupt for shadow region 3. 1h (R/W) = Accesses via region 3 address space to bit 31 to 0 in any DMA channel register are allowed. Reads return the value from bit n and writes modify the state of bit 31 to 0. Enabled interrupt bits for bit n contribute to the generation of a transfer completion interrupt for shadow region 3.

10.4.1.25 DRAEH3 Register (Offset = 35Ch) [reset = 0h]

DRAEH3 is shown in Figure 10-66 and described in Table 10-50.

Return to Summary Table.

The DMA region access enable register for shadow region 3 is programmed to allow or disallow read/write accesses on a bit-by-bit bases for all DMA registers in the shadow region 3 view of the DMA channel registers. Additionally, the DRAE3 configuration determines completion of which DMA channels will result in assertion of the shadow region 3 DMA completion interrupt. The DRAE registers are part of the group of the region access enable registers, which includes DRAEm and QRAEm. Where m is the number of shadow regions in the EDMA3CC memory map for a device. You can configure these registers to assign ownership of DMA/QDMA channels to a particular shadow region.

Figure 10-66. DRAEH3 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">En</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 10-50. DRAEH3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R/W	0h	DMA region access enable for bit 63 to 32 in region 3. 0h (R/W) = Accesses via region 3 address space to bit 63 to 32 in any DMA channel register are not allowed. Reads return 0 on bit n and writes do not modify the state of bit 63 to 32. Enabled interrupt bits for bit 31 to 0 do not contribute to the generation of a transfer completion interrupt for shadow region 3. 1h (R/W) = Accesses via region 3 address space to bit 63 to 32 in any DMA channel register are allowed. Reads return the value from bit n and writes modify the state of bit 63 to 32. Enabled interrupt bits for bit 31 to 0 contribute to the generation of a transfer completion interrupt for shadow region 3.

10.4.1.26 QRAE_0 to QRAE_3 Register (Offset = 380h to 38Ch) [reset = 0h]

QRAE_0 to QRAE_3 is shown in Figure 10-67 and described in Table 10-51.

Return to Summary Table.

The QDMA region access enable register for shadow region m (QRAEm) is programmed to allow or disallow read/write accesses on a bit-by-bit bases for all QDMA registers in the shadow region m view of the QDMA registers. This includes all 4-bit QDMA registers. The QRAE register is part of the group of the region access enable registers, which includes DRAEm and QRAEm. Where m is the number of shadow regions in the EDMA3CC memory map for a device. You can configure these registers to assign ownership of DMA/QDMA channels to a particular shadow region.

Figure 10-67. QRAE_0 to QRAE_3 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">RESERVED</td><td colspan="8">En</td></tr>
<tr><td colspan="24">R-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 10-51. QRAE_0 to QRAE_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	En	R/W	0h	QDMA region access enable for bit n/QDMA channel n in region m. 0h (R/W) = Accesses via region m address space to bit n in any QDMA channel register are not allowed. Reads return 0 on bit n and writes do not modify the state of bit n. 1h (R/W) = Accesses via region m address space to bit n in any QDMA channel register are allowed. Reads return the value from bit n and writes modify the state of bit n.

10.4.1.27 Q0E_0 to Q0E_15 Register (Offset = 400h to 43Ch) [reset = 0h]

Q0E_0 to Q0E_15 is shown in Figure 10-68 and described in Table 10-52.

Return to Summary Table.

The event queue entry registers (QxEy) exist for all 16 queue entries (the maximum allowed queue entries) for all event queues in the EDMA3CC. The event queue entry registers range from Q0E0 to Q0E15, Q1E0 to Q1E15, and Q2E0 to Q2E15. Each register details the event number (ENUM) and the event type (ETYPE). For example, if the value in Q1E4 is read as 000 004Fh, this means the 4th entry in queue 1 is a manually-triggered event on DMA channel 15. The Q0E0 register provides visibility into the event queues and a TR life cycle. These are useful for system debug as they provide in-depth visibility for the events queued up in the event queue and also provide information on what parts of the EDMA3CC logic are active once the event has been received by the EDMA3CC.

Figure 10-68. Q0E_0 to Q0E_15 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ETYPE		ENUM					
R-0h								R-0h		R-0h					

Table 10-52. Q0E_0 to Q0E_15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-6	ETYPE	R	0h	Event entry y in queue 0. Specifies the specific event type for the given entry in the event queue. 0h (R/W) = Event triggered via ER. 1h (R/W) = Auto-triggered via QER.
5-0	ENUM	R	0h	Event entry y in queue 0. Event number: QDMA channel number (0 to 3). DMA channel/event number (0 to 63).

10.4.1.28 Q1E_0 to Q1E_15 Register (Offset = 440h to 47Ch) [reset = 0h]

Q1E_0 to Q1E_15 is shown in Figure 10-69 and described in Table 10-53.

Return to Summary Table.

The event queue entry registers (QxEy) exist for all 16 queue entries (the maximum allowed queue entries) for all event queues in the EDMA3CC. The event queue entry registers range from Q0E0 to Q0E15, Q1E0 to Q1E15, and Q2E0 to Q2E15. Each register details the event number (ENUM) and the event type (ETYPE). For example, if the value in Q1E4 is read as 000 004Fh, this means the 4th entry in queue 1 is a manually-triggered event on DMA channel 15. The Q1E0 register provides visibility into the event queues and a TR life cycle. These are useful for system debug as they provide in-depth visibility for the events queued up in the event queue and also provide information on what parts of the EDMA3CC logic are active once the event has been received by the EDMA3CC.

Figure 10-69. Q1E_0 to Q1E_15 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="2">ETYPE</td><td colspan="6">ENUM</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="2">R-0h</td><td colspan="6">R-0h</td></tr>
</table>

Table 10-53. Q1E_0 to Q1E_15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-6	ETYPE	R	0h	Event entry y in queue 1. Specifies the specific event type for the given entry in the event queue. 0h (R/W) = Event triggered via ER. 1h (R/W) = Auto-triggered via QER.
5-0	ENUM	R	0h	Event entry y in queue 1. Event number: QDMA channel number (0 to 3). DMA channel/event number (0 to 63).

10.4.1.29 Q2E_0 to Q2E_15 Register (Offset = 480h to 4BCh) [reset = 0h]

Q2E_0 to Q2E_15 is shown in Figure 10-70 and described in Table 10-54.

Return to Summary Table.

The event queue entry registers (QxEy) exist for all 16 queue entries (the maximum allowed queue entries) for all event queues in the EDMA3CC. The event queue entry registers range from Q0E0 to Q0E15, Q1E0 to Q1E15, and Q2E0 to Q2E15. Each register details the event number (ENUM) and the event type (ETYPE). For example, if the value in Q1E4 is read as 000 004Fh, this means the 4th entry in queue 1 is a manually-triggered event on DMA channel 15. The Q2E0 register provides visibility into the event queues and a TR life cycle. These are useful for system debug as they provide in-depth visibility for the events queued up in the event queue and also provide information on what parts of the EDMA3CC logic are active once the event has been received by the EDMA3CC.

Figure 10-70. Q2E_0 to Q2E_15 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ETYPE		ENUM					
R-0h								R-0h		R-0h					

Table 10-54. Q2E_0 to Q2E_15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-6	ETYPE	R	0h	Event entry y in queue 2. Specifies the specific event type for the given entry in the event queue. 0h (R/W) = Event triggered via ER. 1h (R/W) = Auto-triggered via QER.
5-0	ENUM	R	0h	Event entry y in queue 2. Event number: QDMA channel number (0 to 3). DMA channel/event number (0 to 63).

10.4.1.30 QSTAT_0 to QSTAT_2 Register (Offset = 600h to 608h) [reset = Fh]

QSTAT_0 to QSTAT_2 is shown in Figure 10-71 and described in Table 10-55.

Return to Summary Table.

The queue status register (QSTAT) provides visibility into the event queues and a TR life cycle. These are useful for system debug as they provide in-depth visibility for the events queued up in the event queue and also provide information on what parts of the EDMA3CC logic are active once the event has been received by the EDMA3CC.

Figure 10-71. QSTAT_0 to QSTAT_2 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="7">RESERVED</td><td>THRXCD</td></tr>
<tr><td colspan="7">R-0h</td><td>R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="3">RESERVED</td><td colspan="5">WM</td></tr>
<tr><td colspan="3">R-0h</td><td colspan="5">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="3">RESERVED</td><td colspan="5">NUMVAL</td></tr>
<tr><td colspan="3">R-0h</td><td colspan="5">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">STRTPTR</td></tr>
<tr><td colspan="4">R-0h</td><td colspan="4">R-Fh</td></tr>
</table>

Table 10-55. QSTAT_0 to QSTAT_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	
24	THRXCD	R	0h	Threshold exceeded. THRXCD is cleared by writing a 1 to the corresponding QTHRXCDn bit in the EDMA3CC error clear register (CCERRCLR). 0h (R/W) = Threshold specified by the Qn bit in the queue watermark threshold A register (QWMTHRA) has not been exceeded. 1h (R/W) = Threshold specified by the Qn bit in the queue watermark threshold A register (QWMTHRA) has been exceeded.
23-21	RESERVED	R	0h	
20-16	WM	R	0h	Watermark for maximum queue usage. Watermark tracks the most entries that have been in queue n since reset or since the last time that the watermark (WM) bit was cleared. WM is cleared by writing a 1 to the corresponding QTHRXCDn bit in the EDMA3CC error clear register (CCERRCLR). 0h (R/W) = Legal values are 0 (empty) to 10h (full). 1h (R/W) = Reserved, from 11h to 1Fh. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
15-13	RESERVED	R	0h	
12-8	NUMVAL	R	0h	Number of valid entries in queue n. The total number of entries residing in the queue manager FIFO at a given instant. Always enabled. 0h (R/W) = Legal values are 0 (empty) to 10h (full). 1h (R/W) = Reserved, from 11h to 1Fh. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7-4	RESERVED	R	0h	
3-0	STRTPTR	R	Fh	Start pointer. The offset to the head entry of queue n, in units of entries. Always enabled. Legal values are 0 (0th entry) to Fh (15th entry).

10.4.1.31 QWMTHRA Register (Offset = 620h) [reset = 000A0A0Ah]

QWMTHRA is shown in Figure 10-72 and described in Table 10-56.

Return to Summary Table.

The queue watermark threshold A register (QWMTHRA) provides visibility into the event queues and a TR life cycle. These are useful for system debug as they provide in-depth visibility for the events queued up in the event queue and also provide information on what parts of the EDMA3CC logic are active once the event has been received by the EDMA3CC.

Figure 10-72. QWMTHRA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED											Q2				
R-0h											R/W-Ah				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			Q1					RESERVED			Q0				
R-0h			R/W-Ah					R-0h			R/W-Ah				

Table 10-56. QWMTHRA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R	0h	
20-16	Q2	R/W	Ah	Queue threshold for queue 2 value. The QTHRXCD2 bit in the EDMA3CC error register (CCERR) and the THRXCD bit in the queue status register 2 (QSTAT2) are set when the number of events in queue 2 at an instant in time (visible via the NUMVAL bit in QSTAT2) equals or exceeds the value specified by Q2. 0h (R/W) = From 0h to 10h, The default is 16 (maximum allowed). 11h (R/W) = Disables the threshold errors. 12h (R/W) = From 12h to 1Fh, Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
15-13	RESERVED	R	0h	
12-8	Q1	R/W	Ah	Queue threshold for queue 1 value. The QTHRXCD1 bit in the EDMA3CC error register (CCERR) and the THRXCD bit in the queue status register 1 (QSTAT1) are set when the number of events in queue 1 at an instant in time (visible via the NUMVAL bit in QSTAT1) equals or exceeds the value specified by Q1. 0h (R/W) = From 0h to 10h, The default is 16 (maximum allowed). 11h (R/W) = Disables the threshold errors. 12h (R/W) = From 12h to 1Fh, Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7-5	RESERVED	R	0h	
4-0	Q0	R/W	Ah	Queue threshold for queue 0 value. The QTHRXCD0 bit in the EDMA3CC error register (CCERR) and the THRXCD bit in the queue status register 0 (QSTAT0) are set when the number of events in queue 0 at an instant in time (visible via the NUMVAL bit in QSTAT0) equals or exceeds the value specified by Q0. 0h (R/W) = From 0h to 10h, The default is 16 (maximum allowed). 11h (R/W) = Disables the threshold errors. 12h (R/W) = From 12h to 1Fh, Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.

10.4.1.32 CCSTAT Register (Offset = 640h) [reset = 0h]

CCSTAT is shown in Figure 10-73 and described in Table 10-57.

Return to Summary Table.

The EDMA3CC status register (CCSTAT) has a number of status bits that reflect which parts of the EDMA3CC logic is active at any given instant of time. CCSTAT provides visibility into the event queues and a TR life cycle. These are useful for system debug as they provide in-depth visibility for the events queued up in the event queue and also provide information on what parts of the EDMA3CC logic are active once the event has been received by the EDMA3CC.

Figure 10-73. CCSTAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED					QUEACTV2	QUEACTV1	QUEACTV0
R-0h					R-0h	R-0h	R-0h

15	14	13	12	11	10	9	8
RESERVED		COMPACTV					
R-0h		R-0h					

7	6	5	4	3	2	1	0
RESERVED			ACTV	RESERVED	TRACTV	QEVTACTV	EVTACTV
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h

Table 10-57. CCSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	
18	QUEACTV2	R	0h	Queue 2 active. 0h (R/W) = No events are queued in queue 2. 1h (R/W) = At least one TR is queued in queue 2.
17	QUEACTV1	R	0h	Queue 1 active. 0h (R/W) = No events are queued in queue 1. 1h (R/W) = At least one TR is queued in queue 1.
16	QUEACTV0	R	0h	Queue 0 active. 0h (R/W) = No events are queued in queue 0. 1h (R/W) = At least one TR is queued in queue 0.
15-14	RESERVED	R	0h	
13-8	COMPACTV	R	0h	Completion request active. The COMPACTV field reflects the count for the number of completion requests submitted to the transfer controllers. This count increments every time a TR is submitted and is programmed to report completion (the TCINTEN or TCCCHEN bits in OPT in the parameter entry associated with the TR are set). The counter decrements for every valid TCC received back from the transfer controllers. If at any time the count reaches a value of 63, the EDMA3CC will not service any new TRs until the count is less then 63 (or return a transfer completion code from a transfer controller, which would decrement the count). 0h (R/W) = No completion requests outstanding. 1h (R/W) = Total of 1 completion request to 63 completion requests are outstanding, from 1h to 3Fh.
7-5	RESERVED	R	0h	

Table 10-57. CCSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACTV	R	0h	Channel controller active. Channel controller active is a logical-OR of each of the *ACTV bits. The ACTV bit remains high through the life of a TR. 0h (R/W) = Channel is idle.. 1h (R/W) = Channel is busy.
3	RESERVED	R	0h	
2	TRACTV	R	0h	Transfer request active. 0h (R/W) = Transfer request processing/submission logic is inactive. 1h (R/W) = Transfer request processing/submission logic is active.
1	QEVTACTV	R	0h	QDMA event active. 0h (R/W) = No enabled QDMA events are active within the EDMA3CC. 1h (R/W) = At least one enabled QDMA event (QER) is active within the EDMA3CC.
0	EVTACTV	R	0h	DMA event active. 0h (R/W) = No enabled DMA events are active within the EDMA3CC. 1h (R/W) = At least one enabled DMA event (ER and EER, ESR, CER) is active within the EDMA3CC.

10.4.1.33 MPFAR Register (Offset = 800h) [reset = 0h]

MPFAR is shown in Figure 10-74 and described in Table 10-58.

Return to Summary Table.

A CPU write of 1 to the MPFCLR bit in the memory protection fault command register (MPFCR) causes any error conditions stored in MPFAR to be cleared.

Figure 10-74. MPFAR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">FADDR</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 10-58. MPFAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FADDR	R	0h	Fault address. This 32 bit read-only status register contains the fault address when a memory protection violation is detected. This register can only be cleared via the memory protection fault command register (MPFCR). Value 0 to FFFF FFFFh.

10.4.1.34 MPFSR Register (Offset = 804h) [reset = 0h]

MPFSR is shown in Figure 10-75 and described in Table 10-59.

Return to Summary Table.

A CPU write of 1 to the MPFCLR bit in the memory protection fault command register (MPFCR) causes any error conditions stored in MPFSR to be cleared.

Figure 10-75. MPFSR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			FID				RESERVED			SRE	SWE	SXE	URE	UWE	UXE
R-0h			R-0h				R-0h			R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 10-59. MPFSR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12-9	FID	R	0h	Faulted identification. FID contains valid information if any of the MP error bits (UXE, UWE, URE, SXE, SWE, SRE) are nonzero (that is, if an error has been detected.) The FID field contains the privilege ID for the specific request/requestor that resulted in an MP error. Value 0 to Fh.
8-6	RESERVED	R	0h	
5	SRE	R	0h	Supervisor read error. 0h (R/W) = No error detected. 1h (R/W) = Supervisor level task attempted to read from a MP page without SR permissions.
4	SWE	R	0h	Supervisor write error. 0h (R/W) = No error detected. 1h (R/W) = Supervisor level task attempted to write to a MP page without SW permissions.
3	SXE	R	0h	Supervisor execute error. 0h (R/W) = No error detected. 1h (R/W) = Supervisor level task attempted to execute from a MP page without SX permissions.
2	URE	R	0h	User read error. 0h (R/W) = No error detected. 1h (R/W) = User level task attempted to read from a MP page without UR permissions.
1	UWE	R	0h	User write error. 0h (R/W) = No error detected. 1h (R/W) = User level task attempted to write to a MP page without UW permissions.
0	UXE	R	0h	User execute error. 0h (R/W) = No error detected. 1h (R/W) = User level task attempted to execute from a MP page without UX permissions.

10.4.1.35 MPFCR Register (Offset = 808h) [reset = 0h]

MPFCR is shown in Figure 10-76 and described in Table 10-60.

Return to Summary Table.

Figure 10-76. MPFCR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							MPFCLR
R-0h							W-0h

Table 10-60. MPFCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	MPFCLR	W	0h	Fault clear register. 0h (R/W) = CPU write of 0 has no effect. 1h (R/W) = CPU write of 1 to the MPFCLR bit causes any error conditions stored in the memory protection fault address register (MPFAR) and the memory protection fault status register (MPFSR) to be cleared.

10.4.1.36 MPPAG Register (Offset = 80Ch) [reset = 676h]

MPPAG is shown in Figure 10-77 and described in Table 10-61.

Return to Summary Table.

Figure 10-77. MPPAG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
AIDm						EXT	RESERVED
R/W-1h						R/W-1h	R-0h

7	6	5	4	3	2	1	0
RESERVED		SR	SW	SX	UR	UW	UX
R-1h		R/W-1h	R/W-1h	R/W-0h	R/W-1h	R/W-1h	R/W-0h

Table 10-61. MPPAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-10	AIDm	R/W	1h	Allowed ID 'N' 0h (R/W) = Requests with Privilege ID == N are not allowed to region M, regardless of permission settings (UW, UR, SW, SR). 1h (R/W) = Requests with Privilege ID == N are permitted, if access type is allowed as defined by permission settings (UW, UR, SW, SR).
9	EXT	R/W	1h	External Allowed ID. 0h (R/W) = Requests with Privilege ID >= 6 are not allowed to region M, regardless of permission settings (UW, UR, SW, SR). 1h (R/W) = Requests with Privilege ID >= 6 are permitted, if access type is allowed as defined by permission settings (UW, UR, SW, SR).
8	RESERVED	R	0h	
7-6	RESERVED	R	1h	
5	SR	R/W	1h	Supervisor read permission. 0h (R/W) = Supervisor read accesses are not allowed from region M. 1h (R/W) = Supervisor write accesses are allowed from region M addresses.
4	SW	R/W	1h	Supervisor write permission. 0h (R/W) = Supervisor write accesses are not allowed to region M. 1h (R/W) = Supervisor write accesses are allowed to region N addresses.
3	SX	R/W	0h	Supervisor execute permission. 0h (R/W) = Supervisor execute accesses are not allowed from region M. 1h (R/W) = Supervisor execute accesses are allowed from region M addresses.
2	UR	R/W	1h	User read permission. 0h (R/W) = User read accesses are not allowed from region M. 1h (R/W) = User read accesses are allowed from region N addresses.

Table 10-61. MPPAG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	UW	R/W	1h	User write permission. 0h (R/W) = User write accesses are not allowed to region M. 1h (R/W) = User write accesses are allowed to region M addresses.
0	UX	R/W	0h	User execute permission. 0h (R/W) = User execute accesses are not allowed from region M. 1h (R/W) = User execute accesses are allowed from region M addresses.

10.4.1.37 MPPA_0 to MPPA_3 Register (Offset = 810h to 81Ch) [reset = 676h]

MPPA_0 to MPPA_3 is shown in Figure 10-78 and described in Table 10-62.

Return to Summary Table.

Figure 10-78. MPPA_0 to MPPA_3 Register

31	30	29	28	27	26	25	24
XXXX							

23	22	21	20	19	18	17	16
XXXX							AIDm
							R/W-0h

15	14	13	12	11	10	9	8
AIDm					EXT	RESERVED	RESERVED
R/W-0h					R/W-1h	R-1h	R-0h

7	6	5	4	3	2	1	0
RESERVED	SR	SW	SX	UR	UW	UX	XXXX
R-0h	R/W-1h	R/W-1h	R/W-1h	R/W-0h	R/W-1h	R/W-1h	

Table 10-62. MPPA_0 to MPPA_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
16-11	AIDm	R/W	0h	Allowed ID 'N' 0h (R/W) = Requests with Privilege ID == N are not allowed to region M, regardless of permission settings (UW, UR, SW, SR). 1h (R/W) = Requests with Privilege ID == N are permitted, if access type is allowed as defined by permission settings (UW, UR, SW, SR).
10	EXT	R/W	1h	External Allowed ID. 0h (R/W) = Requests with Privilege ID >= 6 are not allowed to region M, regardless of permission settings (UW, UR, SW, SR). 1h (R/W) = Requests with Privilege ID >= 6 are permitted, if access type is allowed as defined by permission settings (UW, UR, SW, SR).
9	RESERVED	R	1h	
8-7	RESERVED	R	0h	
6	SR	R/W	1h	Supervisor read permission. 0h (R/W) = Supervisor read accesses are not allowed from region M. 1h (R/W) = Supervisor write accesses are allowed from region M addresses.
5	SW	R/W	1h	Supervisor write permission. 0h (R/W) = Supervisor write accesses are not allowed to region M. 1h (R/W) = Supervisor write accesses are allowed to region N addresses.
4	SX	R/W	1h	Supervisor execute permission. 0h (R/W) = Supervisor execute accesses are not allowed from region M. 1h (R/W) = Supervisor execute accesses are allowed from region M addresses.
3	UR	R/W	0h	User read permission. 0h (R/W) = User read accesses are not allowed from region M. 1h (R/W) = User read accesses are allowed from region N addresses.
2	UW	R/W	1h	User write permission. 0h (R/W) = User write accesses are not allowed to region M. 1h (R/W) = User write accesses are allowed to region M addresses.

Table 10-62. MPPA_0 to MPPA_3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	UX	R/W	1h	User execute permission. 0h (R/W) = User execute accesses are not allowed from region M. 1h (R/W) = User execute accesses are allowed from region M addresses.
15-0	RESERVED	R	676h	

10.4.1.38 ER Register (Offset = 1000h) [reset = 0h]

ER is shown in Figure 10-79 and described in Table 10-63.

Return to Summary Table.

All external events are captured in the event register (ER/ERH). The events are latched even when the events are not enabled. If the event bit corresponding to the latched event is enabled (EER.En/EERH.En = 1), then the event is evaluated by the EDMA3CC logic for an associated transfer request submission to the transfer controllers. The event register bits are automatically cleared (ER.En/ERH.En= 0) once the corresponding events are prioritized and serviced. If ER.En/ERH.En are already set and another event is received on the same channel/event, then the corresponding event is latched in the event miss register (EMR.En/EMRH.En), provided that the event was enabled (EER.En/EERH.En = 1). Event n can be cleared by the CPU writing a 1 to corresponding event bit in the event clear register (ECR/ECRH). The setting of an event is a higher priority relative to clear operations (via hardware or software). If set and clear conditions occur concurrently, the set condition wins. If the event was previously set, then EMR/EMRH would be set because an event is lost. If the event was previously clear, then the event remains set and is prioritized for submission to the event queues. The Debug List table provides the type of synchronization events and the EDMA3CC channels associated to each of these external events. This register is part of a set of registers that pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUMn) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format reg_name that corresponds to DMA channels 0 through 31 and reg_name_High that corresponds to DMA channels 32 through 64. For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register. The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAEm/DRAEHm).

Figure 10-79. ER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
En																															
R-0h																															

Table 10-63. ER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R	0h	Event 0 to 31. Events 0 to 31 are captured by the EDMA3CC and are latched into ER. The events are set (En = 1) even when events are disabled (En = 0 in the event enable register, EER). 0h (R/W) = EDMA3CC event is not asserted. 1h (R/W) = EDMA3CC event is asserted. Corresponding DMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC.

10.4.1.39 ERH Register (Offset = 1004h) [reset = 0h]

ERH is shown in Figure 10-80 and described in Table 10-64.

Return to Summary Table.

All external events are captured in the event register (ER/ERH). The events are latched even when the events are not enabled. If the event bit corresponding to the latched event is enabled (EER.En/EERH.En = 1), then the event is evaluated by the EDMA3CC logic for an associated transfer request submission to the transfer controllers. The event register bits are automatically cleared (ER.En/ERH.En= 0) once the corresponding events are prioritized and serviced. If ER.En/ERH.En are already set and another event is received on the same channel/event, then the corresponding event is latched in the event miss register (EMR.En/EMRH.En), provided that the event was enabled (EER.En/EERH.En = 1). Event n can be cleared by the CPU writing a 1 to corresponding event bit in the event clear register (ECR/ECRH). The setting of an event is a higher priority relative to clear operations (via hardware or software). If set and clear conditions occur concurrently, the set condition wins. If the event was previously set, then EMR/EMRH would be set because an event is lost. If the event was previously clear, then the event remains set and is prioritized for submission to the event queues. The Debug List table provides the type of synchronization events and the EDMA3CC channels associated to each of these external events. This register is part of a set of registers that pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUMn) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format reg_name that corresponds to DMA channels 0 through 31 and reg_name_High that corresponds to DMA channels 32 through 64. For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register. The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAEm/DRAEHm).

Figure 10-80. ERH Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">En</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 10-64. ERH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R	0h	Event 32 to 63. Events 32 to 63 are captured by the EDMA3CC and are latched into ERH. The events are set (En = 1) even when events are disabled (En = 0 in the event enable register high, EERH). 0h (R/W) = EDMA3CC event is not asserted. 1h (R/W) = EDMA3CC event is asserted. Corresponding DMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC.

10.4.1.40 ECR Register (Offset = 1008h) [reset = 0h]

ECR is shown in Figure 10-81 and described in Table 10-65.

Return to Summary Table.

Once an event has been posted in the event registers (ER/ERH), the event is cleared in two ways. If the event is enabled in the event enable register (EER/EERH) and the EDMA3CC submits a transfer request for the event to the EDMA3TC, it clears the corresponding event bit in the event register. If the event is disabled in the event enable register (EER/EERH), the CPU can clear the event by way of the event clear registers (ECR/ECRH). Writing a 1 to any of the bits clears the corresponding event; writing a 0 has no effect. Once an event bit is set in the event register, it remains set until EDMA3CC submits a transfer request for that event or the CPU clears the event by setting the corresponding bit in ECR/ECRH. This register is part of a set of registers that pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUMn) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format reg_name that corresponds to DMA channels 0 through 31 and reg_name_High that corresponds to DMA channels 32 through 64. For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register. The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAEm/DRAEHm).

Figure 10-81. ECR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">En</td></tr>
<tr><td colspan="32">W-0h</td></tr>
</table>

Table 10-65. ECR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	W	0h	Event clear for event 0 to 31. Any of the event bits in ECR is set to clear the event (En) in the event register (ER). A write of 0 has no effect. 0h (R/W) = No effect. 1h (R/W) = EDMA3CC event is cleared in the event register (ER).

10.4.1.41 ECRH Register (Offset = 100Ch) [reset = 0h]

ECRH is shown in Figure 10-82 and described in Table 10-66.

Return to Summary Table.

Once an event has been posted in the event registers (ER/ERH), the event is cleared in two ways. If the event is enabled in the event enable register (EER/EERH) and the EDMA3CC submits a transfer request for the event to the EDMA3TC, it clears the corresponding event bit in the event register. If the event is disabled in the event enable register (EER/EERH), the CPU can clear the event by way of the event clear registers (ECR/ECRH). Writing a 1 to any of the bits clears the corresponding event; writing a 0 has no effect. Once an event bit is set in the event register, it remains set until EDMA3CC submits a transfer request for that event or the CPU clears the event by setting the corresponding bit in ECR/ECRH. This register is part of a set of registers that pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUMn) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format reg_name that corresponds to DMA channels 0 through 31 and reg_name_High that corresponds to DMA channels 32 through 64. For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register. The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAEm/DRAEHm).

Figure 10-82. ECRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
En																															
W-0h																															

Table 10-66. ECRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	W	0h	Event clear for event 32 to 63. Any of the event bits in ECRH are set to clear the event (En) in the event register high (ERH). A write of 0 has no effect. 0h (R/W) = No effect. 1h (R/W) = EDMA3CC event is cleared in the event register high (ERH).

10.4.1.42 ESR Register (Offset = 1010h) [reset = 0h]

ESR is shown in Figure 10-83 and described in Table 10-67.

Return to Summary Table.

The event set registers (ESR/ESRH) allow the CPU (EDMA3 programmers) to manually set events to initiate DMA transfer requests. CPU writes of 1 to any event set register (En) bits set the corresponding bits in the registers. The set event is evaluated by the EDMA3CC logic for an associated transfer request submission to the transfer controllers. Writing a 0 has no effect. The event set registers operate independent of the event registers (ER/ERH), and a write of 1 is always considered a valid event regardless of whether the event is enabled (the corresponding event bits are set or cleared in EER.En/EERH.En). Once the event is set in the event set registers, it cannot be cleared by CPU writes, in other words, the event clear registers (ECR/ECRH) have no effect on the state of ESR/ESRH. The bits will only be cleared once the transfer request corresponding to the event has been submitted to the transfer controller. The setting of an event is a higher priority relative to clear operations (via hardware). If set and clear conditions occur concurrently, the set condition wins. If the event was previously set, then EMR/EMRH would be set because an event is lost. If the event was previously clear, then the event remains set and is prioritized for submission to the event queues. Manually-triggered transfers via writes to ESR/ESRH allow the CPU to submit DMA requests in the system, these are relevant for memory-to-memory transfer scenarios. If the ESR.En/ESRH.En bit is already set and another CPU write of 1 is attempted to the same bit, then the corresponding event is latched in the event missed registers (EMR.En/EMRH.En = 1). This register is part of a set of registers that pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUMn) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format reg_name that corresponds to DMA channels 0 through 31 and reg_name_High that corresponds to DMA channels 32 through 64. For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register. The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAEm/DRAEHm).

Figure 10-83. ESR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
En																															
R/W-0h																															

Table 10-67. ESR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R/W	0h	Event set for event 0 to 31. 0h (R/W) = No effect. 1h (R/W) = Corresponding DMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC.

10.4.1.43 ESRH Register (Offset = 1014h) [reset = 0h]

ESRH is shown in Figure 10-84 and described in Table 10-68.

Return to Summary Table.

The event set registers (ESR/ESRH) allow the CPU (EDMA3 programmers) to manually set events to initiate DMA transfer requests. CPU writes of 1 to any event set register (En) bits set the corresponding bits in the registers. The set event is evaluated by the EDMA3CC logic for an associated transfer request submission to the transfer controllers. Writing a 0 has no effect. The event set registers operate independent of the event registers (ER/ERH), and a write of 1 is always considered a valid event regardless of whether the event is enabled (the corresponding event bits are set or cleared in EER.En/EERH.En). Once the event is set in the event set registers, it cannot be cleared by CPU writes, in other words, the event clear registers (ECR/ECRH) have no effect on the state of ESR/ESRH. The bits will only be cleared once the transfer request corresponding to the event has been submitted to the transfer controller. The setting of an event is a higher priority relative to clear operations (via hardware). If set and clear conditions occur concurrently, the set condition wins. If the event was previously set, then EMR/EMRH would be set because an event is lost. If the event was previously clear, then the event remains set and is prioritized for submission to the event queues. Manually-triggered transfers via writes to ESR/ESRH allow the CPU to submit DMA requests in the system, these are relevant for memory-to-memory transfer scenarios. If the ESR.En/ESRH.En bit is already set and another CPU write of 1 is attempted to the same bit, then the corresponding event is latched in the event missed registers (EMR.En/EMRH.En = 1). This register is part of a set of registers that pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUMn) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format reg_name that corresponds to DMA channels 0 through 31 and reg_name_High that corresponds to DMA channels 32 through 64. For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register. The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAEm/DRAEHm).

Figure 10-84. ESRH Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">En</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 10-68. ESRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R/W	0h	Event set for event 32 to 63. 0h (R/W) = No effect. 1h (R/W) = Corresponding DMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC.

10.4.1.44 CER Register (Offset = 1018h) [reset = 0h]

CER is shown in Figure 10-85 and described in Table 10-69.

Return to Summary Table.

When the OPTIONS parameter for a PaRAM entry is programmed to returned a chained completion code (ITCCHEN = 1 and/or TCCHEN = 1), then the value dictated by the TCC[5:0] (also programmed in OPT) forces the corresponding event bit to be set in the chained event registers (CER/CERH). The set chained event is evaluated by the EDMA3CC logic for an associated transfer request submission to the transfer controllers. This results in a chained-triggered transfer. The chained event registers do not have any enables. The generation of a chained event is essentially enabled by the PaRAM entry that has been configured for intermediate and/or final chaining on transfer completion. The En bit is set (regardless of the state of EER.En/EERH.En) when a chained completion code is returned from one of the transfer controllers or is generated by the EDMA3CC via the early completion path. The bits in the chained event register are cleared when the corresponding events are prioritized and serviced. If the En bit is already set and another chaining completion code is return for the same event, then the corresponding event is latched in the event missed registers (EMR.En/EMRH.En= 1). The setting of an event is a higher priority relative to clear operations (via hardware). If set and clear conditions occur concurrently, the set condition wins. If the event was previously set, then EMR/EMRH would be set because an event is lost. If the event was previously clear, then the event remains set and is prioritized for submission to the event queues. This register is part of a set of registers that pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUMn) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format reg_name that corresponds to DMA channels 0 through 31 and reg_name_High that corresponds to DMA channels 32 through 64. For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register. The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAEm/DRAEHm).

Figure 10-85. CER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
En																															
R-0h																															

Table 10-69. CER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R	0h	Chained event for event 0 to 31. 0h (R/W) = No effect. 1h (R/W) = Corresponding DMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC.

10.4.1.45 CERH Register (Offset = 101Ch) [reset = 0h]

CERH is shown in Figure 10-86 and described in Table 10-70.

Return to Summary Table.

When the OPTIONS parameter for a PaRAM entry is programmed to returned a chained completion code (ITCCHEN = 1 and/or TCCHEN = 1), then the value dictated by the TCC[5:0] (also programmed in OPT) forces the corresponding event bit to be set in the chained event registers (CER/CERH). The set chained event is evaluated by the EDMA3CC logic for an associated transfer request submission to the transfer controllers. This results in a chained-triggered transfer. The chained event registers do not have any enables. The generation of a chained event is essentially enabled by the PaRAM entry that has been configured for intermediate and/or final chaining on transfer completion. The En bit is set (regardless of the state of EER.En/EERH.En) when a chained completion code is returned from one of the transfer controllers or is generated by the EDMA3CC via the early completion path. The bits in the chained event register are cleared when the corresponding events are prioritized and serviced. If the En bit is already set and another chaining completion code is return for the same event, then the corresponding event is latched in the event missed registers (EMR.En/EMRH.En= 1). The setting of an event is a higher priority relative to clear operations (via hardware). If set and clear conditions occur concurrently, the set condition wins. If the event was previously set, then EMR/EMRH would be set because an event is lost. If the event was previously clear, then the event remains set and is prioritized for submission to the event queues. This register is part of a set of registers that pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUMn) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format reg_name that corresponds to DMA channels 0 through 31 and reg_name_High that corresponds to DMA channels 32 through 64. For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register. The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAEm/DRAEHm).

Figure 10-86. CERH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
En																															
R-0h																															

Table 10-70. CERH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R	0h	Chained event set for event 32 to 63. 0h (R/W) = No effect. 1h (R/W) = Corresponding DMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC.

10.4.1.46 EER Register (Offset = 1020h) [reset = 0h]

EER is shown in Figure 10-87 and described in Table 10-71.

Return to Summary Table.

The EDMA3CC provides the option of selectively enabling/disabling each event in the event registers (ER/ERH) by using the event enable registers (EER/EERH). If an event bit in EER/EERH is set (using the event enable set registers, EESR/EESRH), it will enable that corresponding event. Alternatively, if an event bit in EER/EERH is cleared (using the event enable clear registers, EECR/EECRH), it will disable the corresponding event. The event registers latch all events that are captured by EDMA3CC, even if the events are disabled (although EDMA3CC does not process it). Enabling an event with a pending event already set in the event registers enables the EDMA3CC to process the already set event like any other new event. The EER/EERH settings do not have any effect on chained events (CER.En/CERH.En= 1) and manually set events (ESR.En/ESRH.En= 1). This register is part of a set of registers that pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUMn) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format reg_name that corresponds to DMA channels 0 through 31 and reg_name_High that corresponds to DMA channels 32 through 64. For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register. The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAEm/DRAEHm).

Figure 10-87. EER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">En</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 10-71. EER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R	0h	Event enable for events 0 to 31. 0h (R/W) = Event is not enabled. An external event latched in the event register (ER) is not evaluated by the EDMA3CC. 1h (R/W) = Event is enabled. An external event latched in the event register (ER) is evaluated by the EDMA3CC.

10.4.1.47 EERH Register (Offset = 1024h) [reset = 0h]

EERH is shown in Figure 10-88 and described in Table 10-72.

Return to Summary Table.

The EDMA3CC provides the option of selectively enabling/disabling each event in the event registers (ER/ERH) by using the event enable registers (EER/EERH). If an event bit in EER/EERH is set (using the event enable set registers, EESR/EESRH), it will enable that corresponding event. Alternatively, if an event bit in EER/EERH is cleared (using the event enable clear registers, EECR/EECRH), it will disable the corresponding event. The event registers latch all events that are captured by EDMA3CC, even if the events are disabled (although EDMA3CC does not process it). Enabling an event with a pending event already set in the event registers enables the EDMA3CC to process the already set event like any other new event. The EER/EERH settings do not have any effect on chained events (CER.En/CERH.En= 1) and manually set events (ESR.En/ESRH.En= 1). This register is part of a set of registers that pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUMn) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format reg_name that corresponds to DMA channels 0 through 31 and reg_name_High that corresponds to DMA channels 32 through 64. For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register. The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAEm/DRAEHm).

Figure 10-88. EERH Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">En</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 10-72. EERH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R	0h	Event enable for events 32 to 63. 0h (R/W) = Event is not enabled. An external event latched in the event register high (ERH) is not evaluated by the EDMA3CC. 1h (R/W) = Event is enabled. An external event latched in the event register high (ERH) is evaluated by the EDMA3CC.

10.4.1.48 EECR Register (Offset = 1028h) [reset = 0h]

EECR is shown in Figure 10-89 and described in Table 10-73.

Return to Summary Table.

The event enable registers (EER/EERH) cannot be modified by directly writing to them. The intent is to ease the software burden for the case where multiple tasks are attempting to simultaneously modify these registers. The event enable clear registers (EECR/EECRH) are used to disable events. Writes of 1 to the bits in EECR/EECRH clear the corresponding event bits in EER/EERH; writes of 0 have no effect. This register is part of a set of registers that pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUMn) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format reg_name that corresponds to DMA channels 0 through 31 and reg_name_High that corresponds to DMA channels 32 through 64. For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register. The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAEm/DRAEHm).

Figure 10-89. EECR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
En																															
W-0h																															

Table 10-73. EECR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	W	0h	Event enable clear for events 0 to 31. 0h (R/W) = No effect. 1h (R/W) = Event is disabled. Corresponding bit in the event enable register (EER) is cleared (En = 0).

10.4.1.49 EECRH Register (Offset = 102Ch) [reset = 0h]

EECRH is shown in Figure 10-90 and described in Table 10-74.

Return to Summary Table.

The event enable registers (EER/EERH) cannot be modified by directly writing to them. The intent is to ease the software burden for the case where multiple tasks are attempting to simultaneously modify these registers. The event enable clear registers (EECR/EECRH) are used to disable events. Writes of 1 to the bits in EECR/EECRH clear the corresponding event bits in EER/EERH; writes of 0 have no effect. This register is part of a set of registers that pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUMn) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format reg_name that corresponds to DMA channels 0 through 31 and reg_name_High that corresponds to DMA channels 32 through 64. For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register. The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAEm/DRAEHm).

Figure 10-90. EECRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
En																															
W-0h																															

Table 10-74. EECRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	W	0h	Event enable clear for events 32 to 63. 0h (R/W) = No effect. 1h (R/W) = Event is disabled. Corresponding bit in the event enable register high (EERH) is cleared (En = 0).

10.4.1.50 EESR Register (Offset = 1030h) [reset = 0h]

EESR is shown in Figure 10-91 and described in Table 10-75.

Return to Summary Table.

The event enable registers (EER/EERH) cannot be modified by directly writing to them. The intent is to ease the software burden for the case where multiple tasks are attempting to simultaneously modify these registers. The event enable set registers (EESR/EESRH) are used to enable events. Writes of 1 to the bits in EESR/EESRH set the corresponding event bits in EER/EERH; writes of 0 have no effect. This register is part of a set of registers that pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUMn) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format reg_name that corresponds to DMA channels 0 through 31 and reg_name_High that corresponds to DMA channels 32 through 64. For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register. The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAEm/DRAEHm).

Figure 10-91. EESR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
En																															
W-0h																															

Table 10-75. EESR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	W	0h	Event enable set for events 0 to 31. 0h (R/W) = No effect. 1h (R/W) = Event is enabled. Corresponding bit in the event enable register (EER) is set (En = 1).

10.4.1.51 EESRH Register (Offset = 1034h) [reset = 0h]

EESRH is shown in Figure 10-92 and described in Table 10-76.

Return to Summary Table.

The event enable registers (EER/EERH) cannot be modified by directly writing to them. The intent is to ease the software burden for the case where multiple tasks are attempting to simultaneously modify these registers. The event enable set registers (EESR/EESRH) are used to enable events. Writes of 1 to the bits in EESR/EESRH set the corresponding event bits in EER/EERH; writes of 0 have no effect. This register is part of a set of registers that pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUMn) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format reg_name that corresponds to DMA channels 0 through 31 and reg_name_High that corresponds to DMA channels 32 through 64. For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register. The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAEm/DRAEHm).

Figure 10-92. EESRH Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">En</td></tr>
<tr><td colspan="32">W-0h</td></tr>
</table>

Table 10-76. EESRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	W	0h	Event enable set for events 32 to 63. 0h (R/W) = No effect. 1h (R/W) = Event is enabled. Corresponding bit in the event enable register high (EERH) is set (En= 1).

10.4.1.52 SER Register (Offset = 1038h) [reset = 0h]

SER is shown in Figure 10-93 and described in Table 10-77.

Return to Summary Table.

The secondary event registers (SER/SERH) provide information on the state of a DMA channel or event (0 through 63). If the EDMA3CC receives a TR synchronization due to a manual-trigger, event-trigger, or chained-trigger source (ESR.En/ESRH.En= 1, ER.En/ERH.En= 1, or CER.En/CERH.En= 1), which results in the setting of a corresponding event bit in SER/SERH (SER.En/SERH.En = 1), it implies that the corresponding DMA event is in the queue. Once a bit corresponding to an event is set in SER/SERH, the EDMA3CC does not prioritize additional events on the same DMA channel. Depending on the condition that lead to the setting of the SER bits, either the EDMA3CC hardware or the software (using SECR/SECRH) needs to clear the SER/SERH bits for the EDMA3CC to evaluate subsequent events (subsequent transfers) on the same channel. For additional conditions that can cause the secondary event registers to be set, see EDMA Overview. This register is part of a set of registers that pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUMn) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format reg_name that corresponds to DMA channels 0 through 31 and reg_name_High that corresponds to DMA channels 32 through 64. For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register. The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAEm/DRAEHm).

Figure 10-93. SER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">En</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 10-77. SER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R	0h	Secondary event register. The secondary event register is used along with the event register (ER) to provide information on the state of an event. 0h (R/W) = Event is not currently stored in the event queue. 1h (R/W) = Event is currently stored in the event queue. Event arbiter will not prioritize additional events.

10.4.1.53 SERH Register (Offset = 103Ch) [reset = 0h]

SERH is shown in Figure 10-94 and described in Table 10-78.

Return to Summary Table.

The secondary event registers (SER/SERH) provide information on the state of a DMA channel or event (0 through 63). If the EDMA3CC receives a TR synchronization due to a manual-trigger, event-trigger, or chained-trigger source (ESR.En/ESRH.En= 1, ER.En/ERH.En= 1, or CER.En/CERH.En= 1), which results in the setting of a corresponding event bit in SER/SERH (SER.En/SERH.En = 1), it implies that the corresponding DMA event is in the queue. Once a bit corresponding to an event is set in SER/SERH, the EDMA3CC does not prioritize additional events on the same DMA channel. Depending on the condition that lead to the setting of the SER bits, either the EDMA3CC hardware or the software (using SECR/SECRH) needs to clear the SER/SERH bits for the EDMA3CC to evaluate subsequent events (subsequent transfers) on the same channel. For additional conditions that can cause the secondary event registers to be set, see EDMA Overview. This register is part of a set of registers that pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUMn) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format reg_name that corresponds to DMA channels 0 through 31 and reg_name_High that corresponds to DMA channels 32 through 64. For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register. The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAEm/DRAEHm).

Figure 10-94. SERH Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">En</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 10-78. SERH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	R	0h	Secondary event register. The secondary event register is used along with the event register high (ERH) to provide information on the state of an event. 0h (R/W) = Event is not currently stored in the event queue. 1h (R/W) = Event is currently stored in the event queue. Event submission/prioritization logic will not prioritize additional events.

10.4.1.54 SECR Register (Offset = 1040h) [reset = 0h]

SECR is shown in Figure 10-95 and described in Table 10-79.

Return to Summary Table.

The secondary event clear registers (SECR/SECRH) clear the status of the secondary event registers (SER/SERH). CPU writes of 1 clear the corresponding set bits in SER/SERH. Writes of 0 have no effect. This register is part of a set of registers that pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUMn) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format reg_name that corresponds to DMA channels 0 through 31 and reg_name_High that corresponds to DMA channels 32 through 64. For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register. The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAEm/DRAEHm).

Figure 10-95. SECR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
En																															
W-0h																															

Table 10-79. SECR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	W	0h	Secondary event clear register. 0h (R/W) = No effect. 1h (R/W) = Corresponding bit in the secondary event register (SER) is cleared (En = 0).

10.4.1.55 SECRH Register (Offset = 1044h) [reset = 0h]

SECRH is shown in Figure 10-96 and described in Table 10-80.

Return to Summary Table.

The secondary event clear registers (SECR/SECRH) clear the status of the secondary event registers (SER/SERH). CPU writes of 1 clear the corresponding set bits in SER/SERH. Writes of 0 have no effect. This register is part of a set of registers that pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUMn) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format reg_name that corresponds to DMA channels 0 through 31 and reg_name_High that corresponds to DMA channels 32 through 64. For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register. The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAEm/DRAEHm).

Figure 10-96. SECRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
En																															
W-0h																															

Table 10-80. SECRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	En	W	0h	Secondary event clear register. 0h (R/W) = No effect. 1h (R/W) = Corresponding bit in the secondary event registers high (SERH) is cleared (En = 0).

10.4.1.56 IER Register (Offset = 1050h) [reset = 0h]

IER is shown in Figure 10-97 and described in Table 10-81.

Return to Summary Table.

Interrupt enable registers (IER/IERH) are used to enable/disable the transfer completion interrupt generation by the EDMA3CC for all DMA/QDMA channels. The IER/IERH cannot be written to directly. To set any interrupt bit in IER/IERH, a 1 must be written to the corresponding interrupt bit in the interrupt enable set registers (IESR/IESRH). Similarly, to clear any interrupt bit in IER/IERH, a 1 must be written to the corresponding interrupt bit in the interrupt enable clear registers (IECR/IECRH). All DMA/QDMA channels can be set to assert an EDMA3CC completion interrupt to the CPU on transfer completion, by appropriately configuring the PaRAM entry associated with the channels. This register is used for the transfer completion interrupt reporting/generating by the EDMA3CC. For more details on EDMA3CC completion interrupt generation, see EDMA3 Interrupts.

Figure 10-97. IER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
In_																															
R-0h																															

Table 10-81. IER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	In_	R	0h	Interrupt enable for channels 0 to 31. 0h (R/W) = Interrupt is not enabled. 1h (R/W) = Interrupt is enabled.

10.4.1.57 IERH Register (Offset = 1054h) [reset = 0h]

IERH is shown in Figure 10-98 and described in Table 10-82.

Return to Summary Table.

Interrupt enable registers (IER/IERH) are used to enable/disable the transfer completion interrupt generation by the EDMA3CC for all DMA/QDMA channels. The IER/IERH cannot be written to directly. To set any interrupt bit in IER/IERH, a 1 must be written to the corresponding interrupt bit in the interrupt enable set registers (IESR/IESRH). Similarly, to clear any interrupt bit in IER/IERH, a 1 must be written to the corresponding interrupt bit in the interrupt enable clear registers (IECR/IECRH). All DMA/QDMA channels can be set to assert an EDMA3CC completion interrupt to the CPU on transfer completion, by appropriately configuring the PaRAM entry associated with the channels. This register is used for the transfer completion interrupt reporting/generating by the EDMA3CC. For more details on EDMA3CC completion interrupt generation, see EDMA3 Interrupts.

Figure 10-98. IERH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
In																															
R-0h																															

Table 10-82. IERH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	In	R	0h	Interrupt enable for channels 32 to 63. 0h (R/W) = Interrupt is not enabled. 1h (R/W) = Interrupt is enabled.

10.4.1.58 IECR Register (Offset = 1058h) [reset = 0h]

IECR is shown in Figure 10-99 and described in Table 10-83.

Return to Summary Table.

The interrupt enable clear registers (IECR/IECRH) are used to clear interrupts. Writes of 1 to the bits in IECR/IECRH clear the corresponding interrupt bits in the interrupt enable registers (IER/IERH); writes of 0 have no effect. All DMA/QDMA channels can be set to assert an EDMA3CC completion interrupt to the CPU on transfer completion, by appropriately configuring the PaRAM entry associated with the channels. This register is used for the transfer completion interrupt reporting/generating by the EDMA3CC. For more details on EDMA3CC completion interrupt generation, see EDMA3 Interrupts.

Figure 10-99. IECR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
In																															
W-0h																															

Table 10-83. IECR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	In	W	0h	Interrupt enable clear for channels 0 to 31. 0h (R/W) = No effect. 1h (R/W) = Corresponding bit in the interrupt enable register (IER) is cleared (In = 0).

10.4.1.59 IECRH Register (Offset = 105Ch) [reset = 0h]

IECRH is shown in Figure 10-100 and described in Table 10-84.

Return to Summary Table.

The interrupt enable clear registers (IECR/IECRH) are used to clear interrupts. Writes of 1 to the bits in IECR/IECRH clear the corresponding interrupt bits in the interrupt enable registers (IER/IERH); writes of 0 have no effect. All DMA/QDMA channels can be set to assert an EDMA3CC completion interrupt to the CPU on transfer completion, by appropriately configuring the PaRAM entry associated with the channels. This register is used for the transfer completion interrupt reporting/generating by the EDMA3CC. For more details on EDMA3CC completion interrupt generation, see EDMA3 Interrupts.

Figure 10-100. IECRH Register

31-0
In
W-0h

Table 10-84. IECRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	In	W	0h	Interrupt enable clear for channels 32 to 63. 0h (R/W) = No effect. 1h (R/W) = Corresponding bit in the interrupt enable register high (IERH) is cleared (In = 0).

10.4.1.60 IESR Register (Offset = 1060h) [reset = 0h]

IESR is shown in Figure 10-101 and described in Table 10-85.

Return to Summary Table.

The interrupt enable set registers (IESR/IESRH) are used to enable interrupts. Writes of 1 to the bits in IESR/IESRH set the corresponding interrupt bits in the interrupt enable registers (IER/IERH); writes of 0 have no effect. All DMA/QDMA channels can be set to assert an EDMA3CC completion interrupt to the CPU on transfer completion, by appropriately configuring the PaRAM entry associated with the channels. This register is used for the transfer completion interrupt reporting/generating by the EDMA3CC. For more details on EDMA3CC completion interrupt generation, see EDMA3 Interrupts.

Figure 10-101. IESR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">In</td></tr>
<tr><td colspan="32">W-0h</td></tr>
</table>

Table 10-85. IESR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	In	W	0h	Interrupt enable set for channels 0 to 31. 0h (R/W) = No effect. 1h (R/W) = Corresponding bit in the interrupt enable register (IER) is set (In = 1).

10.4.1.61 IESRH Register (Offset = 1064h) [reset = 0h]

IESRH is shown in Figure 10-102 and described in Table 10-86.

Return to Summary Table.

The interrupt enable set registers (IESR/IESRH) are used to enable interrupts. Writes of 1 to the bits in IESR/IESRH set the corresponding interrupt bits in the interrupt enable registers (IER/IERH); writes of 0 have no effect. All DMA/QDMA channels can be set to assert an EDMA3CC completion interrupt to the CPU on transfer completion, by appropriately configuring the PaRAM entry associated with the channels. This register is used for the transfer completion interrupt reporting/generating by the EDMA3CC. For more details on EDMA3CC completion interrupt generation, see EDMA3 Interrupts.

Figure 10-102. IESRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
In																															
W-0h																															

Table 10-86. IESRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	In	W	0h	Interrupt enable clear for channels 32 to 63. 0h (R/W) = No effect. 1h (R/W) = Corresponding bit in the interrupt enable register high (IERH) is set (In = 1).

10.4.1.62 IPR Register (Offset = 1068h) [reset = 0h]

IPR is shown in Figure 10-103 and described in Table 10-87.

Return to Summary Table.

If the TCINTEN and/or ITCINTEN bit in the channel option parameter (OPT) is set in the PaRAM entry associated with the channel (DMA or QDMA), then the EDMA3TC (for normal completion) or the EDMA3CC (for early completion) returns a completion code on transfer or intermediate transfer completion. The value of the returned completion code is equal to the TCC bit in OPT for the PaRAM entry associated with the channel. When an interrupt transfer completion code with TCC = n is detected by the EDMA3CC, then the corresponding bit is set in the interrupt pending register (IPR.In, if n = 0 to 31; IPRH.In, if n = 32 to 63). Note that once a bit is set in the interrupt pending registers, it remains set; it is your responsibility to clear these bits. The bits set in IPR/IPRH are cleared by writing a 1 to the corresponding bits in the interrupt clear registers (ICR/ICRH). All DMA/QDMA channels can be set to assert an EDMA3CC completion interrupt to the CPU on transfer completion, by appropriately configuring the PaRAM entry associated with the channels. This register is used for the transfer completion interrupt reporting/generating by the EDMA3CC. For more details on EDMA3CC completion interrupt generation, see EDMA3 Interrupts.

Figure 10-103. IPR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">In</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 10-87. IPR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	In	R	0h	Interrupt pending for TCC = 0 to 31. 0h (R/W) = Interrupt transfer completion code is not detected or was cleared. 1h (R/W) = Interrupt transfer completion code is detected (In = 1, n = EDMA3TC[2:0]).

10.4.1.63 IPRH Register (Offset = 106Ch) [reset = 0h]

IPRH is shown in Figure 10-104 and described in Table 10-88.

Return to Summary Table.

If the TCINTEN and/or ITCINTEN bit in the channel option parameter (OPT) is set in the PaRAM entry associated with the channel (DMA or QDMA), then the EDMA3TC (for normal completion) or the EDMA3CC (for early completion) returns a completion code on transfer or intermediate transfer completion. The value of the returned completion code is equal to the TCC bit in OPT for the PaRAM entry associated with the channel. When an interrupt transfer completion code with TCC = n is detected by the EDMA3CC, then the corresponding bit is set in the interrupt pending register (IPR.In, if n = 0 to 31; IPRH.In, if n = 32 to 63). Note that once a bit is set in the interrupt pending registers, it remains set; it is your responsibility to clear these bits. The bits set in IPR/IPRH are cleared by writing a 1 to the corresponding bits in the interrupt clear registers (ICR/ICRH). All DMA/QDMA channels can be set to assert an EDMA3CC completion interrupt to the CPU on transfer completion, by appropriately configuring the PaRAM entry associated with the channels. This register is used for the transfer completion interrupt reporting/generating by the EDMA3CC. For more details on EDMA3CC completion interrupt generation, see EDMA3 Interrupts.

Figure 10-104. IPRH Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">In</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 10-88. IPRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	In	R	0h	Interrupt pending for TCC = 32 to 63. 0h (R/W) = Interrupt transfer completion code is not detected or was cleared. 1h (R/W) = Interrupt transfer completion code is detected (In = 1, n = EDMA3TC[2:0]).

10.4.1.64 ICR Register (Offset = 1070h) [reset = 0h]

ICR is shown in Figure 10-105 and described in Table 10-89.

Return to Summary Table.

The bits in the interrupt pending registers (IPR/IPRH) are cleared by writing a 1 to the corresponding bits in the interrupt clear registers(ICR/ICRH). Writes of 0 have no effect. All set bits in IPR/IPRH must be cleared to allow EDMA3CC to assert additional transfer completion interrupts. All DMA/QDMA channels can be set to assert an EDMA3CC completion interrupt to the CPU on transfer completion, by appropriately configuring the PaRAM entry associated with the channels. This register is used for the transfer completion interrupt reporting/generating by the EDMA3CC. For more details on EDMA3CC completion interrupt generation, see EDMA3 Interrupts.

Figure 10-105. ICR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
In																															
W-0h																															

Table 10-89. ICR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	In	W	0h	Interrupt clear register for TCC = 0 to 31. 0h (R/W) = No effect. 1h (R/W) = Corresponding bit in the interrupt pending register (IPR) is cleared (In = 0).

10.4.1.65 ICRH Register (Offset = 1074h) [reset = 0h]

ICRH is shown in Figure 10-106 and described in Table 10-90.

Return to Summary Table.

The bits in the interrupt pending registers (IPR/IPRH) are cleared by writing a 1 to the corresponding bits in the interrupt clear registers(ICR/ICRH). Writes of 0 have no effect. All set bits in IPR/IPRH must be cleared to allow EDMA3CC to assert additional transfer completion interrupts. All DMA/QDMA channels can be set to assert an EDMA3CC completion interrupt to the CPU on transfer completion, by appropriately configuring the PaRAM entry associated with the channels. This register is used for the transfer completion interrupt reporting/generating by the EDMA3CC. For more details on EDMA3CC completion interrupt generation, see EDMA3 Interrupts.

Figure 10-106. ICRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
In																															
W-0h																															

Table 10-90. ICRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	In	W	0h	Interrupt clear register for TCC = 32 to 63. 0h (R/W) = No effect. 1h (R/W) = Corresponding bit in the interrupt pending register high (IPRH) is cleared (In = 0).

10.4.1.66 IEVAL Register (Offset = 1078h) [reset = 0h]

IEVAL is shown in Figure 10-107 and described in Table 10-91.

Return to Summary Table.

The interrupt evaluate register (IEVAL) is the only register that physically exists in both the global region and the shadow regions. In other words, the read/write accessibility for the shadow region IEVAL is not affected by the DMA/QDMA region access registers (DRAEm/DRAEHm, QRAEn/QRAEHn). IEVAL is needed for robust ISR operations to ensure that interrupts are not missed by the CPU. All DMA/QDMA channels can be set to assert an EDMA3CC completion interrupt to the CPU on transfer completion, by appropriately configuring the PaRAM entry associated with the channels. This register is used for the transfer completion interrupt reporting/generating by the EDMA3CC. For more details on EDMA3CC completion interrupt generation, see EDMA3 Interrupts.

Figure 10-107. IEVAL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						RESERVED	EVAL
R-0h						R-0h	W-0h

Table 10-91. IEVAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	RESERVED	R	0h	
0	EVAL	W	0h	Interrupt evaluate. The EDMA3CC completion interrupt that is pulsed depends on which IEVAL is being exercised. For example, writing to the EVAL bit in IEVAL pulses the global completion interrupt, but writing to the EVAL bit in IEVAL0 pulses the region 0 completion interrupt. 0h (R/W) = No effect. 1h (R/W) = Causes EDMA3CC completion interrupt to be pulsed, if any enabled (IERn/IERHn = 1) interrupts are still pending (IPRn/IPRHn = 1).

10.4.1.67 QER Register (Offset = 1080h) [reset = 0h]

QER is shown in Figure 10-108 and described in Table 10-92.

Return to Summary Table.

The QDMA event register (QER) channel n bit is set (En = 1) when the CPU or any EDMA3 programmer (including EDMA3) performs a write to the trigger word (using the QDMA channel mapping register (QCHMAPn)) in the PaRAM entry associated with QDMA channel n (which is also programmed using QCHMAPn). The En bit is also set when the EDMA3CC performs a link update on a PaRAM address that matches the QCHMAPn settings. The QDMA event is latched only if the QDMA event enable register (QEER) channel n bit is also enabled (QEER.En = 1). Once a bit is set in QER, then the corresponding QDMA event (auto-trigger) is evaluated by the EDMA3CC logic for an associated transfer request submission to the transfer controllers. For additional conditions that can lead to the setting of QER bits, see EDMA Overview. The setting of an event is a higher priority relative to clear operations (via hardware). If set and clear conditions occur concurrently, the set condition wins. If the event was previously set, then the QDMA event missed register (QEMR) would be set because an event is lost. If the event was previously clear, then the event remains set and is prioritized for submission to the event queues. The set bits in QER are only cleared when the transfer request associated with the corresponding channels has been processed by the EDMA3CC and submitted to the transfer controller. If the En bit is already set and a QDMA event for the same QDMA channel occurs prior to the original being cleared, then the second missed event is latched in QEMR (En = 1). QER is part of a set of register that control the QDMA channels in EDMA3CC. QDMA channels (with the exception of the QDMA queue number register) consist of a set of registers, each of which have a bit location. Each bit position corresponds to a QDMA channel number. The QDMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write accessibility in the shadow region address region is controlled by the QDMA region access registers (QRAEn/QRAEHn).

Figure 10-108. QER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">RESERVED</td><td colspan="8">En</td></tr>
<tr><td colspan="24">R-0h</td><td colspan="8">R-0h</td></tr>
</table>

Table 10-92. QER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	En	R	0h	QDMA event for channels 0 to 7. 0h (R/W) = No effect. 1h (R/W) = Corresponding QDMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC.

10.4.1.68 QEER Register (Offset = 1084h) [reset = 0h]

QEER is shown in Figure 10-109 and described in Table 10-93.

Return to Summary Table.

The EDMA3CC provides the option of selectively enabling/disabling each channel in the QDMA event register (QER) by using the QDMA event enable register (QEER). If any of the event bits in QEER is set (using the QDMA event enable set register, QEESR), it will enable that corresponding event. Alternatively, if any event bit in QEER is cleared (using the QDMA event enable clear register, QEECR), it will disable the corresponding QDMA channel. The QDMA event register will not latch any event for a QDMA channel, if it is not enabled via QEER. QEER is part of a set of register that control the QDMA channels in EDMA3CC. QDMA channels (with the exception of the QDMA queue number register) consist of a set of registers, each of which have a bit location. Each bit position corresponds to a QDMA channel number. The QDMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write accessibility in the shadow region address region is controlled by the QDMA region access registers (QRAEn/QRAEHn).

Figure 10-109. QEER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">RESERVED</td><td colspan="8">En</td></tr>
<tr><td colspan="24">R-0h</td><td colspan="8">R-0h</td></tr>
</table>

Table 10-93. QEER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	En	R	0h	QDMA event enable for channels 0 to 7. 0h (R/W) = QDMA channel n is not enabled. QDMA event will not be recognized and will not latch in the QDMA event register (QER). 1h (R/W) = QDMA channel n is enabled. QDMA events will be recognized and will get latched in the QDMA event register (QER).

10.4.1.69 QEECR Register (Offset = 1088h) [reset = 0h]

QEECR is shown in Figure 10-110 and described in Table 10-94.

Return to Summary Table.

The QDMA event enable register (QEER) cannot be modified by directly writing to the register, to ease the software burden when multiple tasks are attempting to simultaneously modify these registers. The QDMA event enable clear register (QEECR) is used to disable events. Writes of 1 to the bits in QEECR clear the corresponding QDMA channel bits in QEER; writes of 0 have no effect. QEECR is part of a set of register that control the QDMA channels in EDMA3CC. QDMA channels (with the exception of the QDMA queue number register) consist of a set of registers, each of which have a bit location. Each bit position corresponds to a QDMA channel number. The QDMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write accessibility in the shadow region address region is controlled by the QDMA region access registers (QRAEn/QRAEHn).

Figure 10-110. QEECR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">RESERVED</td><td colspan="8">En</td></tr>
<tr><td colspan="24">R-0h</td><td colspan="8">W-0h</td></tr>
</table>

Table 10-94. QEECR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	En	W	0h	QDMA event enable clear for channels 0 to 7. 0h (R/W) = No effect. 1h (R/W) = QDMA event is disabled. Corresponding bit in the QDMA event enable register (QEER) is cleared (En = 0).

10.4.1.70 QEESR Register (Offset = 108Ch) [reset = 0h]

QEESR is shown in Figure 10-111 and described in Table 10-95.

Return to Summary Table.

The QDMA event enable register (QEER) cannot be modified by directly writing to the register, to ease the software burden when multiple tasks are attempting to simultaneously modify these registers. The QDMA event enable set register (QEESR) is used to enable events. Writes of 1 to the bits in QEESR set the corresponding QDMA channel bits in QEER; writes of 0 have no effect.

Figure 10-111. QEESR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">RESERVED</td><td colspan="8">En</td></tr>
<tr><td colspan="24">R-0h</td><td colspan="8">W-0h</td></tr>
</table>

Table 10-95. QEESR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	En	W	0h	QDMA event enable set for channels 0 to 7. 0h (R/W) = No effect. 1h (R/W) = QDMA event is enabled. Corresponding bit in the QDMA event enable register (QEER) is set (En = 1).

10.4.1.71 QSER Register (Offset = 1090h) [reset = 0h]

QSER is shown in Figure 10-112 and described in Table 10-96.

Return to Summary Table.

The QDMA secondary event register (QSER) provides information on the state of a QDMA event. If at any time a bit corresponding to a QDMA channel is set in QSER, that implies that the corresponding QDMA event is in the queue. Once a bit corresponding to a QDMA channel is set in QSER, the EDMA3CC does not prioritize additional events on the same QDMA channel. Depending on the condition that lead to the setting of the QSER bits, either the EDMA3CC hardware or the software (using QSECR) needs to clear the QSER bits for the EDMA3CC to evaluate subsequent QDMA events on the channel. Based on whether the associated TR request is valid, or it is a null or dummy TR, the implications on the state of QSER and the required user actions to submit another QDMA transfer might be different. For additional conditions that can cause the secondary event registers (QSER\SER) to be set, see EDMA Overview. QSER is part of a set of register that control the QDMA channels in EDMA3CC. QDMA channels (with the exception of the QDMA queue number register) consist of a set of registers, each of which have a bit location. Each bit position corresponds to a QDMA channel number. The QDMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write accessibility in the shadow region address region is controlled by the QDMA region access registers (QRAEn/QRAEHn).

Figure 10-112. QSER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">RESERVED</td><td colspan="8">En</td></tr>
<tr><td colspan="24">R-0h</td><td colspan="8">R-0h</td></tr>
</table>

Table 10-96. QSER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	En	R	0h	QDMA secondary event register for channels 0 to 7. 0h (R/W) = QDMA event is not currently stored in the event queue. 1h (R/W) = QDMA event is currently stored in the event queue. EDMA3CC will not prioritize additional events.

10.4.1.72 QSECR Register (Offset = 1094h) [reset = 0h]

QSECR is shown in Figure 10-113 and described in Table 10-97.

Return to Summary Table.

The QDMA secondary event clear register (QSECR) clears the status of the QDMA secondary event register (QSER) and the QDMA event register (QER). CPU writes of 1 clear the corresponding set bits in QSER and QER. Writes of 0 have no effect. Note that this differs from the secondary event clear register (SECR) operation, which only clears the secondary event register (SER) bits and does not affect the event registers. QESCR is part of a set of register that control the QDMA channels in EDMA3CC. QDMA channels (with the exception of the QDMA queue number register) consist of a set of registers, each of which have a bit location. Each bit position corresponds to a QDMA channel number. The QDMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write accessibility in the shadow region address region is controlled by the QDMA region access registers (QRAEn/QRAEHn).

Figure 10-113. QSECR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								En							
R-0h																								W-0h							

Table 10-97. QSECR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	En	W	0h	QDMA secondary event clear register for channels 0 to 7. 0h (R/W) = No effect. 1h (R/W) = Corresponding bit in the QDMA secondary event register (QSER) and the QDMA event register (QER) is cleared (En = 0).

10.4.2 EDMA3TC Registers

Table 10-98 lists the memory-mapped registers for the EDMA3TC. All register offset addresses not listed in Table 10-98 should be considered as reserved locations and the register contents should not be modified.

Table 10-98. EDMA3TC Registers

Offset	Acronym	Register Name	Section
0h	PID	Peripheral Identification Register	Section 10.4.2.1
4h	TCCFG	EDMA3TC Configuration Register	Section 10.4.2.2
10h	SYSCONFIG	EDMA3TC System Configuration Register	Section 20.1.3.2
100h	TCSTAT	EDMA3TC Channel Status Register	Section 10.4.2.4
120h	ERRSTAT	Error Register	Section 10.4.2.5
124h	ERREN	Error Enable Register	Section 10.4.2.6
128h	ERRCLR	Error Clear Register	Section 10.4.2.7
12Ch	ERRDET	Error Details Register	Section 10.4.2.8
130h	ERRCMD	Error Interrupt Command Register	Section 10.4.2.9
140h	RDRATE	Read Rate Register	Section 10.4.2.10
240h	SAOPT	Source Active Options Register	Section 10.4.2.11
244h	SASRC	Source Active Source Address Register	Section 10.4.2.12
248h	SACNT	Source Active Count Register	Section 10.4.2.13
24Ch	SADST	Source Active Destination Address Register	Section 10.4.2.14
250h	SABIDX	Source Active Source B-Index Register	Section 10.4.2.15
254h	SAMPPRXY	Source Active Memory Protection Proxy Register	Section 10.4.2.16
258h	SACNTRLD	Source Active Count Reload Register	Section 10.4.2.17

Table 10-98. EDMA3TC Registers (continued)

Offset	Acronym	Register Name	Section
25Ch	SASRCBREF	Source Active Source Address B-Reference Register	Section 10.4.2.18
260h	SADSTBREF	Source Active Destination Address B-Reference Register	Section 10.4.2.19
280h	DFCNTRLD	Destination FIFO Set Count Reload	Section 10.4.2.20
284h	DFSRCBREF	Destination FIFO Set Destination Address B Reference Register	Section 10.4.2.21
288h	DFDSTBREF	Destination FIFO Set Destination Address B Reference Register	Section 10.4.2.22
300h	DFOPT0	Destination FIFO Options Register 0	Section 10.4.2.23
304h	DFSRC0	Destination FIFO Source Address Register 0	Section 10.4.2.24
308h	DFCNT0	Destination FIFO Count Register 0	Section 10.4.2.25
30Ch	DFDST0	Destination FIFO Destination Address Register 0	Section 10.4.2.26
310h	DFBIDX0	Destination FIFO BIDX Register 0	Section 10.4.2.27
314h	DFMPPRXY0	Destination FIFO Memory Protection Proxy Register 0	Section 10.4.2.28
340h	DFOPT1	Destination FIFO Options Register 1	Section 10.4.2.29
344h	DFSRC1	Destination FIFO Source Address Register 1	Section 10.4.2.30
348h	DFCNT1	Destination FIFO Count Register 1	Section 10.4.2.31
34Ch	DFDST1	Destination FIFO Destination Address Register 1	Section 10.4.2.32
350h	DFBIDX1	Destination FIFO BIDX Register 1	Section 10.4.2.33
354h	DFMPPRXY1	Destination FIFO Memory Protection Proxy Register 1	Section 10.4.2.34
380h	DFOPT2	Destination FIFO Options Register 2	Section 10.4.2.35
384h	DFSRC2	Destination FIFO Source Address Register 2	Section 10.4.2.36
388h	DFCNT2	Destination FIFO Count Register 2	Section 10.4.2.37
38Ch	DFDST2	Destination FIFO Destination Address Register 2	Section 10.4.2.38
390h	DFBIDX2	Destination FIFO BIDX Register 2	Section 10.4.2.39
394h	DFMPPRXY2	Destination FIFO Memory Protection Proxy Register 2	Section 10.4.2.40
3C0h	DFOPT3	Destination FIFO Options Register 3	Section 10.4.2.41
3C4h	DFSRC3	Destination FIFO Source Address Register 3	Section 10.4.2.42
3C8h	DFCNT3	Destination FIFO Count Register 3	Section 10.4.2.43
3CCh	DFDST3	Destination FIFO Destination Address Register 3	Section 10.4.2.44
3D0h	DFBIDX3	Destination FIFO BIDX Register 3	Section 10.4.2.45
3D4h	DFMPPRXY3	Destination FIFO Memory Protection Proxy Register 3	Section 10.4.2.46

10.4.2.1 PID Register (offset = 0h) [reset = 0h]

PID is shown in Figure 10-114 and described in Table 10-99.

The peripheral identification register (PID) is a constant register that uniquely identifies the EDMA3TC and specific revision of the EDMA3TC.

Figure 10-114. PID Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">PID</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">R-0h</td></tr>
</table>

Table 10-99. PID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	PID	R	0h	Peripheral identifier, value 0 to FFFF FFFFh. Reset for PID[31] to PID[16] is 4000h. Reset for PID[15] to PID[0] is 7C00h.

10.4.2.2 TCCFG Register (offset = 4h) [reset = 224h]

TCCFG is shown in Figure 10-115 and described in Table 10-100.

Figure 10-115. TCCFG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED						DREGDEPTH	
R-0h						R-2h	

7	6	5	4	3	2	1	0
RESERVED		BUSWIDTH		RESERVED	FIFOSIZE		
R-0h		R-2h		R-0h	R-4h		

Table 10-100. TCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9-8	DREGDEPTH	R	2h	Destination register FIFO depth parameterization. 0h (R/W) = Reserved. 1h (R/W) = Reserved. 2h (R/W) = 4 entry (for TC0, TC1, and TC2) 3h (R/W) = Reserved.
7-6	RESERVED	R	0h	
5-4	BUSWIDTH	R	2h	Bus width parameterization. 0h (R/W) = Reserved. 1h (R/W) = Reserved. 2h (R/W) = 128-bit. 3h (R/W) = Reserved.
3	RESERVED	R	0h	
2-0	FIFOSIZE	R	4h	FIFO size 0h (R/W) = Reserved. 1h (R/W) = Reserved. 2h (R/W) = Reserved. 3h (R/W) = Reserved. 4h (R/W) = 512 byte FIFO 5h (R/W) = Reserved. 6h (R/W) = Reserved. 7h (R/W) = Reserved.

10.4.2.3 SYSCONFIG Register (offset = 10h) [reset = 28h]

SYSCONFIG is shown in Figure 20-4 and described in Table 20-7.

Figure 10-116. SYSCONFIG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="2">STANDBYMODE</td><td colspan="2">IDLEMODE</td><td colspan="2">RESERVED</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="2">R/W-2h</td><td colspan="2">R/W-2h</td><td colspan="2">R-0h</td></tr>
</table>

Table 10-101. SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5-4	STANDBYMODE	R/W	2h	Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of STANDBY state. 0h (R/W) = Force-standby mode: local initiator is unconditionally placed in standby state. Backup mode, for debug only. 1h (R/W) = No-standby mode: local initiator is unconditionally placed out of standby state. Backup mode, for debug only. 2h (R/W) = Smart-standby mode: local initiator standby status depends on local conditions, i.e., the module's functional requirement from the initiator. IP module should not generate (initiator-related) wakeup events. 3h (R/W) = Reserved.
3-2	IDLEMODE	R/W	2h	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state. 0h (R/W) = Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements. Backup mode, for debug only. 1h (R/W) = No-idle mode: local target never enters idle state. Backup mode, for debug only. 2h (R/W) = Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ or DMA-request-related) wakeup events. 3h (R/W) = Reserved.
1-0	RESERVED	R	0h	

10.4.2.4 TCSTAT Register (offset = 100h) [reset = 100h]

TCSTAT is shown in Figure 10-117 and described in Table 10-102.

Figure 10-117. TCSTAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED		DFSTRTPTR		RESERVED			RESERVED
R-0h		R-0h		R-0h			R-1h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	DSTACTV			RESERVED	WSACTV	SRCACTV	PROGBUSY
R-0h	R-0h			R-0h	R-0h	R-0h	R-0h

Table 10-102. TCSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	
13-12	DFSTRTPTR	R	0h	Destination FIFO start pointer. Represents the offset to the head entry of the destination register FIFO, in units of entries. Value 0 to 3h.
11-9	RESERVED	R	0h	
8	RESERVED	R	1h	
7	RESERVED	R	0h	
6-4	DSTACTV	R	0h	Destination active state. Specifies the number of transfer requests (TRs) that are resident in the destination register FIFO at a given instant. This bit field can be primarily used for advanced debugging. Legal values are constrained by the destination register FIFO depth parameterization (DSTREGDEPTH) parameter. 0h (R/W) = Destination FIFO is empty. 1h (R/W) = Destination FIFO contains 1 TR. 2h (R/W) = Destination FIFO contains 2 TRs. 3h (R/W) = Destination FIFO contains 3 TRs. 4h (R/W) = Destination FIFO contains 4 TRs. (Full if DSTREGDEPTH==4). If the destination register FIFO is empty, then any TR written to Prog Set immediately transitions to the destination register FIFO. If the destination register FIFO is not empty and not full, then any TR written to Prog Set immediately transitions to the destination register FIFO set if the source active state (SRCACTV) bit is set to idle. If the destination register FIFO is full, then TRs cannot transition to the destination register FIFO. The destination register FIFO becomes not full when the TR at the head of the destination register FIFO is completed. 5h (R/W) = Reserved. 6h (R/W) = Reserved. 7h (R/W) = Reserved.
3	RESERVED	R	0h	
2	WSACTV	R	0h	Write status active 0h (R/W) = Write status is not pending. Write status has been received for all previously issued write commands. 1h (R/W) = Write status is pending. Write status has not been received for all previously issued write commands.

Table 10-102. TCSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	SRCACTV	R	0h	Source active state 0h (R/W) = Source controller is idle. Source active register set contains a previously processed transfer request. 1h (R/W) = Source controller is busy servicing a transfer request.
0	PROGBUSY	R	0h	Program register set busy 0h (R/W) = Program set idle and is available for programming by the EDMA3CC. 1h (R/W) = Program set busy

10.4.2.5 ERRSTAT Register (offset = 120h) [reset = 0h]

ERRSTAT is shown in Figure 10-118 and described in Table 10-103.

Figure 10-118. ERRSTAT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td>MMRAERR</td><td>TRERR</td><td>RESERVED</td><td>BUSERR</td></tr>
<tr><td colspan="4">R-0h</td><td>R-0h</td><td>R-0h</td><td>R-0h</td><td>R-0h</td></tr>
</table>

Table 10-103. ERRSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	MMRAERR	R	0h	MMR address error. 0h (R/W) = Condition is not detected. 1h (R/W) = User attempted to read or write to an invalid address in configuration memory map.
2	TRERR	R	0h	Transfer request (TR) error event. 0h (R/W) = Condition is not detected. 1h (R/W) = TR detected that violates constant addressing mode transfer (SAM or DAM is set) alignment rules or has ACNT or BCNT == 0.
1	RESERVED	R	0h	
0	BUSERR	R	0h	Bus error event. 0h (R/W) = Condition is not detected. 1h (R/W) = EDMA3TC has detected an error at source or destination address. Error information can be read from the error details register (ERRDET).

10.4.2.6 ERREN Register (offset = 124h) [reset = 0h]

ERREN is shown in Figure 10-119 and described in Table 10-104.

When any of the enable bits are set, a bit set in the corresponding ERRSTAT causes an assertion of the EDMA3TC interrupt.

Figure 10-119. ERREN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				MMRAERR	TRERR	RESERVED	BUSERR
R-0h				R/W-0h	R/W-0h	R-0h	R/W-0h

Table 10-104. ERREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	MMRAERR	R/W	0h	Interrupt enable for MMR address error (MMRAERR). 0h (R/W) = MMRAERR is disabled. 1h (R/W) = MMRAERR is enabled and contributes to the state of EDMA3TC error interrupt generation
2	TRERR	R/W	0h	Interrupt enable for transfer request error (TRERR). 0h (R/W) = TRERR is disabled. 1h (R/W) = TRERR is enabled and contributes to the state of EDMA3TC error interrupt generation.
1	RESERVED	R	0h	
0	BUSERR	R/W	0h	Interrupt enable for bus error (BUSERR). 0h (R/W) = BUSERR is disabled. 1h (R/W) = BUSERR is enabled and contributes to the state of EDMA3TC error interrupt generation.

10.4.2.7 ERRCLR Register (offset = 128h) [reset = 0h]

ERRCLR is shown in Figure 10-120 and described in Table 10-105.

Figure 10-120. ERRCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				MMRAERR	TRERR	RESERVED	BUSERR
R-0h				W-0h	W-0h	R-0h	W-0h

Table 10-105. ERRCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	MMRAERR	W	0h	Interrupt enable clear for the MMRAERR bit in the error status register (ERRSTAT). 0h (R/W) = No effect. 1h (R/W) = Clears the MMRAERR bit in ERRSTAT but does not clear the error details register (ERRDET).
2	TRERR	W	0h	Interrupt enable clear for the TRERR bit in the error status register (ERRSTAT). 0h (R/W) = No effect. 1h (R/W) = Clears the TRERR bit in ERRSTAT but does not clear the error details register (ERRDET).
1	RESERVED	R	0h	
0	BUSERR	W	0h	Interrupt clear for the BUSERR bit in the error status register (ERRSTAT). 0h (R/W) = No effect. 1h (R/W) = Clears the BUSERR bit in ERRSTAT and clears the error details register (ERRDET).

10.4.2.8 ERRDET Register (offset = 12Ch) [reset = 0h]

ERRDET is shown in Figure 10-121 and described in Table 10-106.

Figure 10-121. ERRDET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						TCCHEN	TCINTEN
R-0h						R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED		TCC					
R-0h		R-0h					
7	6	5	4	3	2	1	0
RESERVED				STAT			
R-0h				R-0h			

Table 10-106. ERRDET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	
17	TCCHEN	R	0h	Transfer completion chaining enable. Contains the TCCHEN value in the channel options parameter (OPT) programmed by the channel controller for the read or write transaction that resulted in an error.
16	TCINTEN	R	0h	Transfer completion interrupt enable. Contains the TCINTEN value in the channel options parameter (OPT) programmed by the channel controller for the read or write transaction that resulted in an error.
15-14	RESERVED	R	0h	
13-8	TCC	R	0h	Transfer complete code. Contains the TCC value in the channel options parameter (OPT) programmed by the channel controller for the read or write transaction that resulted in an error.
7-4	RESERVED	R	0h	
3-0	STAT	R	0h	Transaction status. Stores the nonzero status/error code that was detected on the read status or write status bus. If read status and write status are returned on the same cycle, then the EDMA3TC chooses nonzero version. If both are nonzero, then the write status is treated as higher priority. 0h (R/W) = No error. 1h (R/W) = From 1h to 7h, Read error. 8h (R/W) = From 8h to Fh, Write error.

10.4.2.9 ERRCMD Register (offset = 130h) [reset = 0h]

ERRCMD is shown in Figure 10-122 and described in Table 10-107.

Figure 10-122. ERRCMD Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>RESERVED</td><td>EVAL</td></tr>
<tr><td colspan="6">R-0h</td><td>R-0h</td><td>W-0h</td></tr>
</table>

Table 10-107. ERRCMD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	RESERVED	R	0h	
0	EVAL	W	0h	Error evaluate 0h (R/W) = No effect 1h (R/W) = EDMA3TC error line is pulsed if any of the error status register (ERRSTAT) bits are set.

10.4.2.10 RDRATE Register (offset = 140h) [reset = 0h]

RDRATE is shown in Figure 10-123 and described in Table 10-108.

The EDMA3 transfer controller issues read commands at a rate controlled by the read rate register (RDRATE). The RDRATE defines the number of idle cycles that the read controller must wait before issuing subsequent commands. This applies both to commands within a transfer request packet (TRP) and for commands that are issued for different transfer requests (TRs). For instance, if RDRATE is set to 4 cycles between reads, there are 3 inactive cycles between reads. RDRATE allows flexibility in transfer controller access requests to an endpoint. For an application, RDRATE can be manipulated to slow down the access rate, so that the endpoint may service requests from other masters during the inactive EDMA3TC cycles. Note: The RDRATE value for a transfer controller is expected to be static, as it is decided based on the application requirement. It is not recommended to change the RDRATE value on the fly.

Figure 10-123. RDRATE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													RDRATE		
R-0h													R/W-0h		

Table 10-108. RDRATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2-0	RDRATE	R/W	0h	Read rate. Controls the number of cycles between read commands. This is a global setting that applies to all TRs for this EDMA3TC. 0h (R/W) = Reads issued as fast as possible. 1h (R/W) = 4 cycles between reads. 2h (R/W) = 8 cycles between reads. 3h (R/W) = 16 cycles between reads. 4h (R/W) = 32 cycles between reads. 5h (R/W) = Reserved. 6h (R/W) = Reserved. 7h (R/W) = Reserved.

10.4.2.11 SAOPT Register (offset = 240h) [reset = 0h]

SAOPT is shown in Figure 10-124 and described in Table 10-109.

The Source Active Options Register (SAOPT) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Source Active Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing the status of the transfer controller (TC) during a transfer.

Figure 10-124. SAOPT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>RESERVED</td><td>TCCHEN</td><td>RESERVED</td><td>TCINTEN</td><td colspan="2">RESERVED</td><td colspan="2">TCC</td></tr>
<tr><td>R-0h</td><td>R-0h</td><td>R-0h</td><td>R-0h</td><td colspan="2">R-0h</td><td colspan="2">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="4">TCC</td><td>RESERVED</td><td colspan="3">FWID</td></tr>
<tr><td colspan="4">R-0h</td><td>R-0h</td><td colspan="3">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>RESERVED</td><td colspan="3">PRI</td><td colspan="2">RESERVED</td><td>DAM</td><td>SAM</td></tr>
<tr><td>R-0h</td><td colspan="3">R-0h</td><td colspan="2">R-0h</td><td>R-0h</td><td>R-0h</td></tr>
</table>

Table 10-109. SAOPT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	0h	
22	TCCHEN	R	0h	Transfer complete chaining enable 0h (R/W) = Transfer complete chaining is disabled. 1h (R/W) = Transfer complete chaining is enabled.
21	RESERVED	R	0h	
20	TCINTEN	R	0h	Transfer complete interrupt enable. 0h (R/W) = Transfer complete interrupt is disabled. 1h (R/W) = Transfer complete interrupt is enabled.
19-18	RESERVED	R	0h	
17-12	TCC	R	0h	Transfer complete code. This 6 bit code is used to set the relevant bit in CER or IPR of the EDMA3PCC module.
11	RESERVED	R	0h	
10-8	FWID	R	0h	FIFO width. Applies if either SAM or DAM is set to constant addressing mode. 0h (R/W) = FIFO width is 8-bit. 1h (R/W) = FIFO width is 16-bit. 2h (R/W) = FIFO width is 32-bit. 3h (R/W) = FIFO width is 64-bit. 4h (R/W) = FIFO width is 128-bit. 5h (R/W) = FIFO width is 256-bit. 6h (R/W) = Reserved. 7h (R/W) = Reserved.
7	RESERVED	R	0h	
6-4	PRI	R	0h	Transfer priority. Reflects the values programmed in the QUEPRI register in the EDMACC. 0h (R/W) = Priority 0 - Highest priority 1h (R/W) = From 1h to 6h, Priority 1 to priority 6 7h (R/W) = Priority 7 - Lowest priority

Table 10-109. SAOPT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	RESERVED	R	0h	
1	DAM	R	0h	Destination address mode within an array 0h (R/W) = Increment (INCR) mode. Destination addressing within an array increments. 1h (R/W) = Constant addressing (CONST) mode. Destination addressing within an array wraps around upon reaching FIFO width.
0	SAM	R	0h	Source address mode within an array 0h (R/W) = Increment (INCR) mode. Source addressing within an array increments. 1h (R/W) = Constant addressing (CONST) mode. Source addressing within an array wraps around upon reaching FIFO width.

10.4.2.12 SASRC Register (offset = 244h) [reset = 0h]

SASRC is shown in Figure 10-125 and described in Table 10-110.

The Source Active Source Address Register (SASRC) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Source Active Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing the status of the transfer controller (TC) during a transfer.

Figure 10-125. SASRC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR																															
R-0h																															

Table 10-110. SASRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SADDR	R	0h	Source address for Source Active Register Set. EDMA3TC updates value according to source addressing mode (SAM bit in the source active options register, SAOPT). This register does not update during a transfer. Value 0 to FFFFh.

10.4.2.13 SACNT Register (offset = 248h) [reset = 0h]

SACNT is shown in Figure 10-126 and described in Table 10-111.

The Source Active Count Register (SACNT) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Source Active Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing the status of the transfer controller (TC) during a transfer.

Figure 10-126. SACNT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">BCNT</td><td colspan="16">ACNT</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">R-0h</td></tr>
</table>

Table 10-111. SACNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	BCNT	R	0h	B dimension count remaining for the Source Active Register Set. Number of arrays to be transferred, where each array is ACNT in length.
15-0	ACNT	R	0h	A dimension count remaining for the Source Active Register Set. Number of bytes to be transferred in first dimension.

10.4.2.14 SADST Register (offset = 24Ch) [reset = 0h]

SADST is shown in Figure 10-127 and described in Table 10-112.

The Source Active Destination Address Register (SADST) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Source Active Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing the status of the transfer controller (TC) during a transfer. Note: Destination address is not applicable for Source Active Register Set. Read returns 0.

Figure 10-127. SADST Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-0h																															

Table 10-112. SADST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	

10.4.2.15 SABIDX Register (offset = 250h) [reset = 0h]

SABIDX is shown in Figure 10-128 and described in Table 10-113.

The Source Active Source B-Dimension Index Register (SABIDX) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Source Active Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing the status of the transfer controller (TC) during a transfer.

Figure 10-128. SABIDX Register

31-16	15-0
DBIDX	SBIDX
R-0h	R-0h

Table 10-113. SABIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DBIDX	R	0h	B-Index offset between destination arrays. Represents the offset in bytes between the starting address of each destination array (there are BCND arrays of ACNT elements). DBIDX is always used regardless of whether DAM is in Increment or FIFO mode.
15-0	SBIDX	R	0h	B-Index offset between source arrays. Represents the offset in bytes between the starting address of each source array (there are BCNT arrays of ACNT elements). SBIDX is always used regardless of whether SAM is in Increment or FIFO mode. Value 0 to FFFFh.

10.4.2.16 SAMPPRXY Register (offset = 254h) [reset = 0h]

SAMPPRXY is shown in Figure 10-129 and described in Table 10-114.

The Source Active Memory Protection Proxy Register (SAMPPRXY) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Source Active Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing the status of the transfer controller (TC) during a transfer.

Figure 10-129. SAMPPRXY Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>PRIV</td></tr>
<tr><td colspan="7">R-0h</td><td>R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">PRIVID</td></tr>
<tr><td colspan="4">R-0h</td><td colspan="4">R-0h</td></tr>
</table>

Table 10-114. SAMPPRXY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	PRIV	R	0h	Privilege level. The privilege level used by the host to set up the parameter entry in the channel controller. This field is set up when the associated TR is submitted to the EDMA3TC. The privilege ID is used while issuing read and write command to the target endpoints so that the target endpoints can perform memory protection checks based on the PRIV of the host that set up the DMA transaction. 0h (R/W) = User-level privilege. 1h (R/W) = Supervisor-level privilege.
7-4	RESERVED	R	0h	
3-0	PRIVID	R	0h	Privilege ID. This contains the privilege ID of the host that set up the parameter entry in the channel controller. This field is set up when the associated TR is submitted to the EDMA3TC. This PRIVID value is used while issuing read and write commands to the target endpoints so that the target endpoints can perform memory protection checks based on the PRIVID of the host that set up the DMA transaction. Value 0 to Fh.

10.4.2.17 SACNTRLD Register (offset = 258h) [reset = 0h]

SACNTRLD is shown in Figure 10-130 and described in Table 10-115.

The Source Active Count Reload Register (SACNTRLD) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Source Active Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing the status of the transfer controller (TC) during a transfer.

Figure 10-130. SACNTRLD Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">ACNTRLD</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">R-0h</td></tr>
</table>

Table 10-115. SACNTRLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	ACNTRLD	R	0h	A-count reload value. Represents the originally programmed value of ACNT. The reload value is used to reinitialize ACNT after each array is serviced (that is, ACNT decrements to 0) by the source offset in bytes between the starting address or each source array (there are BCNT arrays of ACNT bytes). Value 0 to FFFFh.

10.4.2.18 SASRCBREF Register (offset = 25Ch) [reset = 0h]

SASRCBREF is shown in Figure 10-131 and described in Table 10-116.

The Source Active Source Address B-Reference Register (SASRCBREF) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Source Active Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing the status of the transfer controller (TC) during a transfer.

Figure 10-131. SASRCBREF Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">SADDRBREF</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 10-116. SASRCBREF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SADDRBREF	R	0h	Source address B-reference. Represents the starting address for the array currently being read. The next array's starting address is calculated as the reference address plus the source b-idx value. Value 0 to FFFF FFFFh.

10.4.2.19 SADSTBREF Register (offset = 260h) [reset = 0h]

SADSTBREF is shown in Figure 10-132 and described in Table 10-117.

The Source Active Destination Address B-Reference Register (SADSTBREF) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Source Active Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing the status of the transfer controller (TC) during a transfer. Note: Destination address reference is not applicable for the Source Active Register Set. Read returns 0.

Figure 10-132. SADSTBREF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-0h																															

Table 10-117. SADSTBREF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	

10.4.2.20 DFCNTRLD Register (offset = 280h) [reset = 0h]

DFCNTRLD is shown in Figure 10-133 and described in Table 10-118.

The Destination FIFO Count Reload Register (DFCNTRLD) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-133. DFCNTRLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ACNTRLD															
R-0h																R-0h															

Table 10-118. DFCNTRLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	ACNTRLD	R	0h	A-count reload value for the Destination FIFO Register Set. Represents the originally programmed value of ACNT. The reload value is used to reinitialize ACNT after each array is serviced (that is, ACNT decrements to 0) by the source offset in bytes between the starting address of each source array (there are BCNT arrays of ACNT bytes). Value 0 to FFFFh.

10.4.2.21 DFSRCBREF Register (offset = 284h) [reset = 0h]

DFSRCBREF is shown in Figure 10-134 and described in Table 10-119.

The Destination FIFO Source Address B-Reference Register (DFSRCBREF) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers. Note: Source address reference is not applicable for Destination FIFO Register Set. Read returns 0.

Figure 10-134. DFSRCBREF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-0h																															

Table 10-119. DFSRCBREF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	

10.4.2.22 DFDSTBREF Register (offset = 288h) [reset = 0h]

DFDSTBREF is shown in Figure 10-135 and described in Table 10-120.

The Destination FIFO Destination Address B-Reference Register (DFDSTBREF) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-135. DFDSTBREF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DADDRBREF																															
R-0h																															

Table 10-120. DFDSTBREF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DADDRBREF	R	0h	Destination address reference for the destination FIFO register set. Represents the starting address for the array currently being written. The next array's starting address is calculated as the reference address plus the destination B-Index value. Value 0 to FFFF FFFFh.

10.4.2.23 DFOPT0 Register (offset = 300h) [reset = 0h]

DFOPT0 is shown in Figure 10-136 and described in Table 10-121.

The Destination FIFO Options Register (DFOPT) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-136. DFOPT0 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED	TCCHEN	RESERVED	TCINTEN	RESERVED		TCC	
R-0h	R-0h	R-0h	R-0h	R-0h		R-0h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TCC				RESERVED	FWID		
R-0h				R-0h	R-0h		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	PRI			RESERVED		DAM	SAM
R-0h	R-0h			R-0h		R-0h	R-0h

Table 10-121. DFOPT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	0h	
22	TCCHEN	R	0h	Transfer complete chaining enable 0h (R/W) = Transfer complete chaining is disabled 1h (R/W) = Transfer complete chaining is enabled
21	RESERVED	R	0h	
20	TCINTEN	R	0h	Transfer complete interrupt enable. 0h (R/W) = Transfer complete interrupt is disabled. 1h (R/W) = Transfer complete interrupt is enabled.
19-18	RESERVED	R	0h	
17-12	TCC	R	0h	Transfer complete code. This 6 bit code is used to set the relevant bit in CER or IPR of the EDMA3PCC module.
11	RESERVED	R	0h	
10-8	FWID	R	0h	FIFO width. Applies if either SAM or DAM is set to constant addressing mode. 0h (R/W) = FIFO width is 8-bit. 1h (R/W) = FIFO width is 16-bit. 2h (R/W) = FIFO width is 32-bit. 3h (R/W) = FIFO width is 64-bit. 4h (R/W) = FIFO width is 128-bit. 5h (R/W) = FIFO width is 256-bit. 6h (R/W) = Reserved. 7h (R/W) = Reserved.
7	RESERVED	R	0h	

Table 10-121. DFOPT0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6-4	PRI	R	0h	Transfer priority 0h (R/W) = Priority 0 - Highest priority 1h (R/W) = From 1h to 6h, Priority 1 to priority 6 7h (R/W) = Priority 7 - Lowest priority
3-2	RESERVED	R	0h	
1	DAM	R	0h	Destination address mode within an array 0h (R/W) = Increment (INCR) mode. Destination addressing within an array increments. 1h (R/W) = Constant addressing (CONST) mode. Destination addressing within an array wraps around upon reaching FIFO width.
0	SAM	R	0h	Source address mode within an array 0h (R/W) = Increment (INCR) mode. Source addressing within an array increments. 1h (R/W) = Constant addressing (CONST) mode. Source addressing within an array wraps around upon reaching FIFO width.

10.4.2.24 DFSRC0 Register (offset = 304h) [reset = 0h]

DFSRC0 is shown in Figure 10-137 and described in Table 10-122.

The Destination FIFO Source Address Register (DFSRC) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers. Note: Source address is not applicable for Destination FIFO Register Set. Read returns 0.

Figure 10-137. DFSRC0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">RESERVED</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 10-122. DFSRC0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	

10.4.2.25 DFCNT0 Register (offset = 308h) [reset = 0h]

DFCNT0 is shown in Figure 10-138 and described in Table 10-123.

The Destination FIFO Count Register (DFCNT) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-138. DFCNT0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">BCNT</td><td colspan="16">ACNT</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">R-0h</td></tr>
</table>

Table 10-123. DFCNT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	BCNT	R	0h	B-dimension count remaining for Destination Register Set. Represents the amount of data remaining to be written. For the final TR in the Destination Register FIFO: TC decrements ACNT and BCNT as necessary after each write dataphase is issued. The final value should be 0 when TR is complete. For a non-final TR in the Destination Register FIFO: CNT will hold the originally programmed value or the optimized BCNT value after 2D optimization calculation.
15-0	ACNT	R	0h	A-dimension count remaining for Destination Register Set. Represents the amount of data remaining to be written. For the final TR in the Destination Register FIFO: TC decrements ACNT and BCNT as necessary after each write dataphase is issued. The final value should be 0 when TR is complete. For a non-final TR in the Destination Register FIFO: CNT will hold the originally programmed value or the optimized BCNT value after 2D optimization calculation. Value 0 to FFFFh.

10.4.2.26 DFDST0 Register (offset = 30Ch) [reset = 0h]

DFDST0 is shown in Figure 10-139 and described in Table 10-124.

The Destination FIFO Destination Address Register (DFDST) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-139. DFDST0 Register

31-0
DADDR
R-0h

Table 10-124. DFDST0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DADDR	R	0h	ARRAY(0x24369c0) Note: If DAM == CONST, the 'active' address will increment internally as if the transfer were an 'Increment' transfer. The address issued on the write command interface will correctly issue the same address programmed by the user.

10.4.2.27 DFBIDX0 Register (offset = 310h) [reset = 0h]

DFBIDX0 is shown in Figure 10-140 and described in Table 10-125.

The Destination FIFO B-Index Register (DFBIDX) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-140. DFBIDX0 Register

31-16	15-0
DBIDX	SBIDX
R-0h	R-0h

Table 10-125. DFBIDX0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DBIDX	R	0h	B-Index offset between destination arrays for the Destination FIFO Register Set. Represents the offset in bytes between the starting address of each destination array (there are BCNT arrays of ACNT elements). DBIDX is always used regardless of whether DAM is in Increment or FIFO mode. Value 0 to FFFFh.
15-0	SBIDX	R	0h	B-Index offset between source arrays for the Destination FIFO Register Set. Represents the offset in bytes between the starting address of each source array (there are BCNT arrays of ACNT elements). SBIDX is always used regardless of whether SAM is in Increment or FIFO mode.

10.4.2.28 DFMPPRXY0 Register (offset = 314h) [reset = 0h]

DFMPPRXY0 is shown in Figure 10-141 and described in Table 10-126.

The Destination FIFO Memory Protection Proxy Register (DFMPPRXY) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-141. DFMPPRXY0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>PRIV</td></tr>
<tr><td colspan="7">R-0h</td><td>R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">PRIVID</td></tr>
<tr><td colspan="4">R-0h</td><td colspan="4">R-0h</td></tr>
</table>

Table 10-126. DFMPPRXY0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	PRIV	R	0h	Privilege level. This contains the Privilege level used by the EDMA3 programmer to set up the parameter entry in the channel controller. This field is set up when the associated TR is submitted to the EDMA3TC. The privilege ID is used while issuing read and write command to the target endpoints so that the target endpoints can perform memory protection checks based on the PRIV of the host that set up the DMA transaction. 0h (R/W) = User-level privilege 1h (R/W) = Supervisor-level privilege
7-4	RESERVED	R	0h	
3-0	PRIVID	R	0h	Privilege ID. This contains the Privilege ID of the EDMA3 programmer that set up the parameter entry in the channel controller. This field is set up when the associated TR is submitted to the EDMA3TC. This PRIVID value is used while issuing read and write commands to the target endpoints so that the target endpoints can perform memory protection checks based on the PRIVID of the host that set up the DMA transaction. Value 0 to Fh.

10.4.2.29 DFOPT1 Register (offset = 340h) [reset = 0h]

DFOPT1 is shown in Figure 10-142 and described in Table 10-127.

The Destination FIFO Options Register (DFOPT) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-142. DFOPT1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED	TCCHEN	RESERVED	TCINTEN	RESERVED		TCC	
R-0h	R-0h	R-0h	R-0h	R-0h		R-0h	
15	14	13	12	11	10	9	8
TCC				RESERVED	FWID		
R-0h				R-0h	R-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI			RESERVED		DAM	SAM
R-0h	R-0h			R-0h		R-0h	R-0h

Table 10-127. DFOPT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	0h	
22	TCCHEN	R	0h	Transfer complete chaining enable 0h (R/W) = Transfer complete chaining is disabled 1h (R/W) = Transfer complete chaining is enabled
21	RESERVED	R	0h	
20	TCINTEN	R	0h	Transfer complete interrupt enable. 0h (R/W) = Transfer complete interrupt is disabled. 1h (R/W) = Transfer complete interrupt is enabled.
19-18	RESERVED	R	0h	
17-12	TCC	R	0h	Transfer complete code. This 6 bit code is used to set the relevant bit in CER or IPR of the EDMA3PCC module.
11	RESERVED	R	0h	
10-8	FWID	R	0h	FIFO width. Applies if either SAM or DAM is set to constant addressing mode. 0h (R/W) = FIFO width is 8-bit. 1h (R/W) = FIFO width is 16-bit. 2h (R/W) = FIFO width is 32-bit. 3h (R/W) = FIFO width is 64-bit. 4h (R/W) = FIFO width is 128-bit. 5h (R/W) = FIFO width is 256-bit. 6h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. 7h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7	RESERVED	R	0h	

Table 10-127. DFOPT1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6-4	PRI	R	0h	Transfer priority 0h (R/W) = Priority 0 - Highest priority 1h (R/W) = From 1h to 6h, Priority 1 to priority 6 7h (R/W) = Priority 7 - Lowest priority
3-2	RESERVED	R	0h	
1	DAM	R	0h	Destination address mode within an array 0h (R/W) = Increment (INCR) mode. Destination addressing within an array increments. 1h (R/W) = Constant addressing (CONST) mode. Destination addressing within an array wraps around upon reaching FIFO width.
0	SAM	R	0h	Source address mode within an array 0h (R/W) = Increment (INCR) mode. Source addressing within an array increments. 1h (R/W) = Constant addressing (CONST) mode. Source addressing within an array wraps around upon reaching FIFO width.

10.4.2.30 DFSRC1 Register (offset = 344h) [reset = 0h]

DFSRC1 is shown in Figure 10-143 and described in Table 10-128.

The Destination FIFO Source Address Register (DFSRC) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers. Note: Source address is not applicable for the Destination FIFO Register Set. Read returns 0.

Figure 10-143. DFSRC1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-0h																															

Table 10-128. DFSRC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	

10.4.2.31 DFCNT1 Register (offset = 348h) [reset = 0h]

DFCNT1 is shown in Figure 10-144 and described in Table 10-129.

The Destination FIFO Count Register (DFCNT) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-144. DFCNT1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">BCNT</td><td colspan="16">ACNT</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">R-0h</td></tr>
</table>

Table 10-129. DFCNT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	BCNT	R	0h	B-dimension count remaining for Destination Register Set. Represents the amount of data remaining to be written. For the final TR in the Destination Register FIFO: TC decrements ACNT and BCNT as necessary after each write dataphase is issued. The final value should be 0 when TR is complete. For a non-final TR in the Destination Register FIFO: CNT will hold the originally programmed value or the optimized BCNT value after 2D optimization calculation.
15-0	ACNT	R	0h	A-dimension count remaining for Destination Register Set. Represents the amount of data remaining to be written. For the final TR in the Destination Register FIFO: TC decrements ACNT and BCNT as necessary after each write dataphase is issued. The final value should be 0 when TR is complete. For a non-final TR in the Destination Register FIFO: CNT will hold the originally programmed value or the optimized BCNT value after 2D optimization calculation. Value 0 to FFFFh.

10.4.2.32 DFDST1 Register (offset = 34Ch) [reset = 0h]

DFDST1 is shown in Figure 10-145 and described in Table 10-130.

The Destination FIFO Destination Address Register (DFDST) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-145. DFDST1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">DADDR</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 10-130. DFDST1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DADDR	R	0h	ARRAY(0x245d830) Note: If DAM == CONST, the 'active' address will increment internally as if the transfer were an 'Increment' transfer. The address issued on the write command interface will correctly issue the same address programmed by the user.

10.4.2.33 DFBIDX1 Register (offset = 350h) [reset = 0h]

DFBIDX1 is shown in Figure 10-146 and described in Table 10-131.

The Destination FIFO B-Index Register (DFBIDX) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-146. DFBIDX1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">DBIDX</td><td colspan="16">SBIDX</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">R-0h</td></tr>
</table>

Table 10-131. DFBIDX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DBIDX	R	0h	B-Index offset between destination arrays for the Destination FIFO Register Set. Represents the offset in bytes between the starting address of each destination array (there are BCNT arrays of ACNT elements). DBIDX is always used regardless of whether DAM is in Increment or FIFO mode. Value 0 to FFFFh.
15-0	SBIDX	R	0h	B-Index offset between source arrays for the Destination FIFO Register Set. Represents the offset in bytes between the starting address of each source array (there are BCNT arrays of ACNT elements). SBIDX is always used regardless of whether SAM is in Increment or FIFO mode.

10.4.2.34 DFMPPRXY1 Register (offset = 354h) [reset = 0h]

DFMPPRXY1 is shown in Figure 10-147 and described in Table 10-132.

The Destination FIFO Memory Protection Proxy Register (DFMPPRXY) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-147. DFMPPRXY1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>PRIV</td></tr>
<tr><td colspan="7">R-0h</td><td>R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">PRIVID</td></tr>
<tr><td colspan="4">R-0h</td><td colspan="4">R-0h</td></tr>
</table>

Table 10-132. DFMPPRXY1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	PRIV	R	0h	Privilege level. This contains the Privilege level used by the EDMA3 programmer to set up the parameter entry in the channel controller. This field is set up when the associated TR is submitted to the EDMA3TC. The privilege ID is used while issuing read and write command to the target endpoints so that the target endpoints can perform memory protection checks based on the PRIV of the host that set up the DMA transaction. 0h (R/W) = User-level privilege 1h (R/W) = Supervisor-level privilege
7-4	RESERVED	R	0h	
3-0	PRIVID	R	0h	Privilege ID. This contains the Privilege ID of the EDMA3 programmer that set up the parameter entry in the channel controller. This field is set up when the associated TR is submitted to the EDMA3TC. This PRIVID value is used while issuing read and write commands to the target endpoints so that the target endpoints can perform memory protection checks based on the PRIVID of the host that set up the DMA transaction. Value 0 to Fh.

10.4.2.35 DFOPT2 Register (offset = 380h) [reset = 0h]

DFOPT2 is shown in Figure 10-148 and described in Table 10-133.

The Destination FIFO Options Register (DFOPT) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-148. DFOPT2 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>RESERVED</td><td>TCCHEN</td><td>RESERVED</td><td>TCINTEN</td><td colspan="2">RESERVED</td><td colspan="2">TCC</td></tr>
<tr><td>R-0h</td><td>R-0h</td><td>R-0h</td><td>R-0h</td><td colspan="2">R-0h</td><td colspan="2">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="4">TCC</td><td>RESERVED</td><td colspan="3">FWID</td></tr>
<tr><td colspan="4">R-0h</td><td>R-0h</td><td colspan="3">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>RESERVED</td><td colspan="3">PRI</td><td colspan="2">RESERVED</td><td>DAM</td><td>SAM</td></tr>
<tr><td>R-0h</td><td colspan="3">R-0h</td><td colspan="2">R-0h</td><td>R-0h</td><td>R-0h</td></tr>
</table>

Table 10-133. DFOPT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	0h	
22	TCCHEN	R	0h	Transfer complete chaining enable 0h (R/W) = Transfer complete chaining is disabled 1h (R/W) = Transfer complete chaining is enabled
21	RESERVED	R	0h	
20	TCINTEN	R	0h	Transfer complete interrupt enable. 0h (R/W) = Transfer complete interrupt is disabled. 1h (R/W) = Transfer complete interrupt is enabled.
19-18	RESERVED	R	0h	
17-12	TCC	R	0h	Transfer complete code. This 6 bit code is used to set the relevant bit in CER or IPR of the EDMA3PCC module.
11	RESERVED	R	0h	
10-8	FWID	R	0h	FIFO width. Applies if either SAM or DAM is set to constant addressing mode. 0h (R/W) = FIFO width is 8-bit. 1h (R/W) = FIFO width is 16-bit. 2h (R/W) = FIFO width is 32-bit. 3h (R/W) = FIFO width is 64-bit. 4h (R/W) = FIFO width is 128-bit. 5h (R/W) = FIFO width is 256-bit. 6h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. 7h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7	RESERVED	R	0h	

Table 10-133. DFOPT2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6-4	PRI	R	0h	Transfer priority 0h (R/W) = Priority 0 - Highest priority 1h (R/W) = From 1h to 6h, Priority 1 to priority 6 7h (R/W) = Priority 7 - Lowest priority
3-2	RESERVED	R	0h	
1	DAM	R	0h	Destination address mode within an array 0h (R/W) = Increment (INCR) mode. Destination addressing within an array increments. 1h (R/W) = Constant addressing (CONST) mode. Destination addressing within an array wraps around upon reaching FIFO width.
0	SAM	R	0h	Source address mode within an array 0h (R/W) = Increment (INCR) mode. Source addressing within an array increments. 1h (R/W) = Constant addressing (CONST) mode. Source addressing within an array wraps around upon reaching FIFO width.

10.4.2.36 DFSRC2 Register (offset = 384h) [reset = 0h]

DFSRC2 is shown in Figure 10-149 and described in Table 10-134.

The Destination FIFO Source Address Register (DFSRC) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers. Note: Source address is not applicable for Destination FIFO Register Set. Read returns 0.

Figure 10-149. DFSRC2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-0h																															

Table 10-134. DFSRC2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	

10.4.2.37 DFCNT2 Register (offset = 388h) [reset = 0h]

DFCNT2 is shown in Figure 10-150 and described in Table 10-135.

The Destination FIFO Count Register (DFCNT) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-150. DFCNT2 Register

31-16	15-0
BCNT	ACNT
R-0h	R-0h

Table 10-135. DFCNT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	BCNT	R	0h	B-dimension count remaining for Destination Register Set. Represents the amount of data remaining to be written. For the final TR in the Destination Register FIFO: TC decrements ACNT and BCNT as necessary after each write dataphase is issued. The final value should be 0 when TR is complete. For a non-final TR in the Destination Register FIFO: CNT will hold the originally programmed value or the optimized BCNT value after 2D optimization calculation.
15-0	ACNT	R	0h	A-dimension count remaining for Destination Register Set. Represents the amount of data remaining to be written. For the final TR in the Destination Register FIFO: TC decrements ACNT and BCNT as necessary after each write dataphase is issued. The final value should be 0 when TR is complete. For a non-final TR in the Destination Register FIFO: CNT will hold the originally programmed value or the optimized BCNT value after 2D optimization calculation. Value 0 to FFFFh.

10.4.2.38 DFDST2 Register (offset = 38Ch) [reset = 0h]

DFDST2 is shown in Figure 10-151 and described in Table 10-136.

The Destination FIFO Destination Address Register (DFDST) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-151. DFDST2 Register

31-0
DADDR
R-0h

Table 10-136. DFDST2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DADDR	R	0h	ARRAY(0x248ac60) Note: If DAM == CONST, the 'active' address will increment internally as if the transfer were an 'Increment' transfer. The address issued on the write command interface will correctly issue the same address programmed by the user.

10.4.2.39 DFBIDX2 Register (offset = 390h) [reset = 0h]

DFBIDX2 is shown in Figure 10-152 and described in Table 10-137.

The Destination FIFO B-Index Register (DFBIDX) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-152. DFBIDX2 Register

31-16	15-0
DBIDX	SBIDX
R-0h	R-0h

Table 10-137. DFBIDX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DBIDX	R	0h	B-Index offset between destination arrays for the Destination FIFO Register Set. Represents the offset in bytes between the starting address of each destination array (there are BCNT arrays of ACNT elements). DBIDX is always used regardless of whether DAM is in Increment or FIFO mode. Value 0 to FFFFh.
15-0	SBIDX	R	0h	B-Index offset between source arrays for the Destination FIFO Register Set. Represents the offset in bytes between the starting address of each source array (there are BCNT arrays of ACNT elements). SBIDX is always used regardless of whether SAM is in Increment or FIFO mode.

10.4.2.40 DFMPPRXY2 Register (offset = 394h) [reset = 0h]

DFMPPRXY2 is shown in Figure 10-153 and described in Table 10-138.

The Destination FIFO Memory Protection Proxy Register (DFMPPRXY) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-153. DFMPPRXY2 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>PRIV</td></tr>
<tr><td colspan="7">R-0h</td><td>R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">PRIVID</td></tr>
<tr><td colspan="4">R-0h</td><td colspan="4">R-0h</td></tr>
</table>

Table 10-138. DFMPPRXY2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	PRIV	R	0h	Privilege level. This contains the Privilege level used by the EDMA3 programmer to set up the parameter entry in the channel controller. This field is set up when the associated TR is submitted to the EDMA3TC. The privilege ID is used while issuing read and write command to the target endpoints so that the target endpoints can perform memory protection checks based on the PRIV of the host that set up the DMA transaction. 0h (R/W) = User-level privilege 1h (R/W) = Supervisor-level privilege
7-4	RESERVED	R	0h	
3-0	PRIVID	R	0h	Privilege ID. This contains the Privilege ID of the EDMA3 programmer that set up the parameter entry in the channel controller. This field is set up when the associated TR is submitted to the EDMA3TC. This PRIVID value is used while issuing read and write commands to the target endpoints so that the target endpoints can perform memory protection checks based on the PRIVID of the host that set up the DMA transaction. Value 0 to Fh.

10.4.2.41 DFOPT3 Register (offset = 3C0h) [reset = 0h]

DFOPT3 is shown in Figure 10-154 and described in Table 10-139.

The Destination FIFO Options Register (DFOPT) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-154. DFOPT3 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>RESERVED</td><td>TCCHEN</td><td>RESERVED</td><td>TCINTEN</td><td colspan="2">RESERVED</td><td colspan="2">TCC</td></tr>
<tr><td>R-0h</td><td>R-0h</td><td>R-0h</td><td>R-0h</td><td colspan="2">R-0h</td><td colspan="2">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="4">TCC</td><td>RESERVED</td><td colspan="3">FWID</td></tr>
<tr><td colspan="4">R-0h</td><td>R-0h</td><td colspan="3">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>RESERVED</td><td colspan="3">PRI</td><td colspan="2">RESERVED</td><td>DAM</td><td>SAM</td></tr>
<tr><td>R-0h</td><td colspan="3">R-0h</td><td colspan="2">R-0h</td><td>R-0h</td><td>R-0h</td></tr>
</table>

Table 10-139. DFOPT3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	0h	
22	TCCHEN	R	0h	Transfer complete chaining enable 0h (R/W) = Transfer complete chaining is disabled 1h (R/W) = Transfer complete chaining is enabled
21	RESERVED	R	0h	
20	TCINTEN	R	0h	Transfer complete interrupt enable. 0h (R/W) = Transfer complete interrupt is disabled. 1h (R/W) = Transfer complete interrupt is enabled.
19-18	RESERVED	R	0h	
17-12	TCC	R	0h	Transfer complete code. This 6 bit code is used to set the relevant bit in CER or IPR of the EDMA3PCC module.
11	RESERVED	R	0h	
10-8	FWID	R	0h	FIFO width. Applies if either SAM or DAM is set to constant addressing mode. 0h (R/W) = FIFO width is 8-bit. 1h (R/W) = FIFO width is 16-bit. 2h (R/W) = FIFO width is 32-bit. 3h (R/W) = FIFO width is 64-bit. 4h (R/W) = FIFO width is 128-bit. 5h (R/W) = FIFO width is 256-bit. 6h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. 7h (R/W) = Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7	RESERVED	R	0h	

Table 10-139. DFOPT3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6-4	PRI	R	0h	Transfer priority 0h (R/W) = Priority 0 - Highest priority 1h (R/W) = From 1h to 6h, Priority 1 to priority 6 7h (R/W) = Priority 7 - Lowest priority
3-2	RESERVED	R	0h	
1	DAM	R	0h	Destination address mode within an array 0h (R/W) = Increment (INCR) mode. Destination addressing within an array increments. 1h (R/W) = Constant addressing (CONST) mode. Destination addressing within an array wraps around upon reaching FIFO width.
0	SAM	R	0h	Source address mode within an array 0h (R/W) = Increment (INCR) mode. Source addressing within an array increments. 1h (R/W) = Constant addressing (CONST) mode. Source addressing within an array wraps around upon reaching FIFO width.

10.4.2.42 DFSRC3 Register (offset = 3C4h) [reset = 0h]

DFSRC3 is shown in Figure 10-155 and described in Table 10-140.

The Destination FIFO Source Address Register (DFSRC) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers. Note: Source address is not applicable for Destination FIFO Register Set. Read returns 0.

Figure 10-155. DFSRC3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-0h																															

Table 10-140. DFSRC3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	

10.4.2.43 DFCNT3 Register (offset = 3C8h) [reset = 0h]

DFCNT3 is shown in Figure 10-156 and described in Table 10-141.

The Destination FIFO Count Register (DFCNT) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-156. DFCNT3 Register

31-16	15-0
BCNT	ACNT
R-0h	R-0h

Table 10-141. DFCNT3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	BCNT	R	0h	B-dimension count remaining for Destination Register Set. Represents the amount of data remaining to be written. For the final TR in the Destination Register FIFO: TC decrements ACNT and BCNT as necessary after each write dataphase is issued. The final value should be 0 when TR is complete. For a non-final TR in the Destination Register FIFO: CNT will hold the originally programmed value or the optimized BCNT value after 2D optimization calculation.
15-0	ACNT	R	0h	A-dimension count remaining for Destination Register Set. Represents the amount of data remaining to be written. For the final TR in the Destination Register FIFO: TC decrements ACNT and BCNT as necessary after each write dataphase is issued. The final value should be 0 when TR is complete. For a non-final TR in the Destination Register FIFO: CNT will hold the originally programmed value or the optimized BCNT value after 2D optimization calculation. Value 0 to FFFFh.

10.4.2.44 DFDST3 Register (offset = 3CCh) [reset = 0h]

DFDST3 is shown in Figure 10-157 and described in Table 10-142.

The Destination FIFO Destination Address Register (DFDST) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-157. DFDST3 Register

31–0
DADDR
R-0h

Table 10-142. DFDST3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DADDR	R	0h	ARRAY(0x24b1d10) Note: If DAM == CONST, the 'active' address will increment internally as if the transfer were an 'Increment' transfer. The address issued on the write command interface will correctly issue the same address programmed by the user.

10.4.2.45 DFBIDX3 Register (offset = 3D0h) [reset = 0h]

DFBIDX3 is shown in Figure 10-158 and described in Table 10-143.

The Destination FIFO B-Index Register (DFBIDX) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-158. DFBIDX3 Register

31-16	15-0
DBIDX	SBIDX
R-0h	R-0h

Table 10-143. DFBIDX3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DBIDX	R	0h	B-Index offset between destination arrays for the Destination FIFO Register Set. Represents the offset in bytes between the starting address of each destination array (there are BCNT arrays of ACNT elements). DBIDX is always used regardless of whether DAM is in Increment or FIFO mode. Value 0 to FFFFh.
15-0	SBIDX	R	0h	B-Index offset between source arrays for the Destination FIFO Register Set. Represents the offset in bytes between the starting address of each source array (there are BCNT arrays of ACNT elements). SBIDX is always used regardless of whether SAM is in Increment or FIFO mode.

10.4.2.46 DFMPPRXY3 Register (offset = 3D4h) [reset = 0h]

DFMPPRXY3 is shown in Figure 10-159 and described in Table 10-144.

The Destination FIFO Memory Protection Proxy Register (DFMPPRXY) is an EDMA3TC channel register. This EDMA3TC channel register is part of the Destination Register FIFO Register Set. It is read-only and provided to facilitate debugging by providing a window into how the transfer controller (TC) was programmed by the channel controller (CC), as well as showing status of the transfer controller (TC) during a transfer. The number of destination FIFO register sets depends on the destination FIFO depth. TC0, TC1, and TC2 each have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

Figure 10-159. DFMPPRXY3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							PRIV
R-0h							R-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				PRIVID			
R-0h				R-0h			

Table 10-144. DFMPPRXY3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	PRIV	R	0h	Privilege level. This contains the Privilege level used by the EDMA3 programmer to set up the parameter entry in the channel controller. This field is set up when the associated TR is submitted to the EDMA3TC. The privilege ID is used while issuing read and write command to the target endpoints so that the target endpoints can perform memory protection checks based on the PRIV of the host that set up the DMA transaction. 0h (R/W) = User-level privilege 1h (R/W) = Supervisor-level privilege
7-4	RESERVED	R	0h	
3-0	PRIVID	R	0h	Privilege ID. This contains the Privilege ID of the EDMA3 programmer that set up the parameter entry in the channel controller. This field is set up when the associated TR is submitted to the EDMA3TC. This PRIVID value is used while issuing read and write commands to the target endpoints so that the target endpoints can perform memory protection checks based on the PRIVID of the host that set up the DMA transaction. Value 0 to Fh.

10.5 Appendix A

10.5.1 Debug Checklist

This section lists some tips to keep in mind while debugging applications using the EDMA3.

The following table provides some common issues and their probable causes and resolutions.

Table 10-145. Debug List

Issue	Description/Solution
The transfer associated with the channel does not happen. The channel does not get serviced.	The EDMA3CC may not service a transfer request, even though the associated PaRAM set is programmed appropriately. Check for the following: 1) Verify that events are enabled, i.e., if an external/peripheral event is latched in Event Registers (ER/ERH), make sure that the event is enabled in the Event Enable Registers (EER/EERH). Similarly, for QDMA channels, make sure that QDMA events are appropriately enabled in the QDMA Event Enable Register (QEER). 2) Verify that the DMA or QDMA Secondary Event Register (SER/SERH/QSERH) bits corresponding to the particular event or channel are not set.
The Secondary Event Registers bits are set, not allowing additional transfers to occur on a channel.	It is possible that a trigger event was received when the parameter set associated with the channel/event was a NULL set for a previous transfer on the channel. This is typical in two cases: 1) QDMA channels: Typically if the parameter set is non-static and expected to be terminated by a NULL set (i.e., OPT.STATIC = 0, LINK = 0xFFFF), the parameter set is updated with a NULL set after submission of the last TR. Because QDMA channels are auto-triggered, this update caused the generation of an event. An event generated for a NULL set causes an error condition and results in setting the bits corresponding to the QDMA channel in the QEMR and QSER. This will disable further prioritization of the channel. 2) DMA channels used in a continuous mode: The peripheral may be set up to continuously generate infinite events (for instance, in case of McASP, every time the data shifts out from the DXR register, it generates an XEVT). The parameter set may be programmed to expect only a finite number of events and to be terminated by a NULL link. After the expected number of events, the parameter set is reloaded with a NULL parameter set. Because the peripheral will generate additional events, an error condition is set in the SER.Ex and EMR.Ex set, preventing further event prioritization. You must ensure that the number of events received is limited to the expected number of events for which the parameter set is programmed, or you must ensure that bits corresponding to particular channel or event are not set in the Secondary event registers (SER/SERH/QSER) and Event Missed Registers (EMR/EMRH/QEMR) before trying to perform subsequent transfers for the event/channel.
Completion interrupts are not asserted, or no further interrupts are received after the first completion interrupt.	You must ensure the following: 1) The interrupt generation is enabled in the OPT of the associated PaRAM set (TCINTEN = 1 and/or ITCINTEN = 1). 2) The interrupts are enabled in the EDMA3 Channel Controller, via the Interrupt Enable Registers (IER/IERH). 3) The corresponding interrupts are enabled in the device interrupt controller. 4) The set interrupts are cleared in the interrupt pending registers (IPR/IPRH) before exiting the transfer completion interrupt service routine (ISR). See Section 10.3.9.1.2 for details on writing EDMA3 ISRs. 5) If working with shadow region interrupts, make sure that the DMA Region Access registers (DRAE/DRAEH) are set up properly, because the DRAE/DRAEH registers act as secondary enables for shadow region completion interrupts, along with the IER/IERH registers. If working with shadow region interrupts, make sure that the bits corresponding to the transfer completion code (TCC) value are also enabled in the DRAE/DRAEH registers. For instance, if the PaRAM set associated with Channel 0 returns a completion code of 63 (OPT.TCC=63), ensure that DRAEH.E63 is also set for a shadow region completion interrupt because the interrupt pending register bit set will be IPRH.I63 (not IPR.I0).

10.5.2 Miscellaneous Programming/Debug Tips

1. For several registers, the setting and clearing of bits needs to be done via separate dedicated registers. For example, the Event Register (ER/ERH) can only be cleared by writing a 1 to the corresponding bits in the Event Clear Registers (ECR/ECRH). Similarly, the Event Enable Register (EER/EERH) bits can only be set with writes of 1 to the Event Enable Set Registers (EESR/EESRH) and cleared with writes of 1 to the corresponding bits in the Event Enable Clear Register (EECR/EECRH).
2. Writes to the shadow region memory maps are governed by region access registers (DRAE/DRAEH/QRAE). If the appropriate channels are not enabled in these registers, read/write access to the shadow region memory map is not enabled.
3. When working with shadow region completion interrupts, ensure that the DMA Region Access Registers (DRAE/DRAEH) for every region are set in a mutually exclusive way (unless it is a requirement for an application). If there is an overlap in the allocated channels and transfer completion codes (setting of Interrupt Pending Register bits) in the region resource allocation, it results in multiple

shadow region completion interrupts. For example, if DRAE0.E0 and DRAE1.E0 are both set, then on completion of a transfer that returns a TCC=0, they will generate both shadow region 0 and 1 completion interrupts.

4. While programming a non-dummy parameter set, ensure the CCNT is not left to zero.
5. Enable the EDMA3CC error interrupt in the device controller and attach an interrupt service routine (ISR) to ensure that error conditions are not missed in an application and are appropriately addressed with the ISR.
6. Depending on the application, you may want to break large transfers into smaller transfers and use self-chaining to prevent starvation of other events in an event queue.
7. In applications where a large transfer is broken into sets of small transfers using chaining or other methods, you might choose to use the early chaining option to reduce the time between the sets of transfers and increase the throughput. However, keep in mind that with early completion, all data might have not been received at the end point when completion is reported because the EDMA3CC internally signals completion when the TR is submitted to the EDMA3TC, potentially before any data has been transferred.
8. The event queue entries can be observed to determine the last few events if there is a system failure (provided the entries were not bypassed).

10.5.3 Setting Up a Transfer

The following list provides a quick guide for the typical steps involved in setting up a transfer.

Step 1. Initiating a DMA/QDMA channel

1. Determine the type of channel (QDMA or DMA) to be used.
2. Channel mapping
    - i. If using a QDMA channel, program the QCHMAP with the parameter set number to which the channel maps and the trigger word.
    - ii. If using a DMA channel, program the DCHMAP with the parameter set number to which the channel maps.
3. If the channel is being used in the context of a shadow region, ensure the DRAE/DRAEH for the region is properly set up to allow read write accesses to bits in the event registers and interrupt registers in the Shadow region memory map. The subsequent steps in this process should be done using the respective shadow region registers. (Shadow region descriptions and usage are provided in Section 10.3.7.1.)
4. Determine the type of triggering used.
    - i. If external events are used for triggering (DMA channels), enable the respective event in EER/EERH by writing into EESR/EESRH.
    - ii. If QDMA Channel is used, enable the channel in QEER by writing into QEESR.
5. Queue setup
    - i. If a QDMA channel is used, set up the QDMAQNUM to map the channel to the respective event queue.
    - ii. If a DMA channel is used, set up the DMAQNUM to map the event to the respective event queue.

Step 2. Parameter set setup

1. Program the PaRAM set number associated with the channel. Note that if it is a QDMA channel, the PaRAM entry that is configured as trigger word is written to last. Alternatively, enable the QDMA channel (step 1-b-ii above) just before the write to the trigger word.
   See Section 10.3.19 for parameter set field setups for different types of transfers. See the sections on chaining (Section 10.3.8) and interrupt completion (Section 10.3.9) on how to set up final/intermediate completion chaining and/or interrupts.

Step 3. Interrupt setup

1. Enable the interrupt in the IER/IERH by writing into IESR/IESRH.
2. Ensure that the EDMA3CC completion interrupt (either the global or the shadow region interrupt) is enabled properly in the device interrupt controller.
3. Ensure the EDMA3CC completion interrupt (this refers to either the Global interrupt or the shadow region interrupt) is enabled properly in the Device Interrupt controller.
4. Set up the interrupt controller properly to receive the expected EDMA3 interrupt.

Step 4. Initiate transfer

1. This step is highly dependent on the event trigger source:
    - i. If the source is an external event coming from a peripheral, the peripheral will be enabled to start generating relevant EDMA3 events that can be latched to the ER transfer.
    - ii. For QDMA events, writes to the trigger word (step 2-a above) will initiate the transfer.
    - iii. Manually triggered transfers will be initiated by writes to the Event Set Registers (ESR/ESRH).
    - iv. Chained-trigger events initiate when a previous transfer returns a transfer completion code equal to the chained channel number.

Step 5. Wait for completion

1. If the interrupts are enabled as mentioned in step 3 above, then the EDMA3CC will generate a completion interrupt to the CPU whenever transfer completion results in setting the corresponding bits in the interrupt pending register (IPR/IPRH). The set bits must be cleared in the IPR\IPRH by writing to corresponding bit in ICR\ICRH.
2. If polling for completion (interrupts not enabled in the device controller), then the application code can wait on the expected bits to be set in the IPR\IPRH. Again, the set bits in the IPR\IPRH must be manually cleared via ICR\ICRH before the next set of transfers is performed for the same transfer completion code values.

ADC0: Touchscreen Controller

This chapter describes the touchscreen controller of the device.

NOTE: The Touchscreen Controller (TSC) is not supported for the AMIC120.

Topic **Page**

11.1 Introduction

The touchscreen controller and analog-to-digital converter subsystem (TSC_ADC_SS or ADC0) contains a single-channel ADC connected to an 8-to-1 analog multiplexer which operates as a general-purpose analog-to-digital converter (ADC) with optional support for interleaving touchscreen (TS) conversions for a 4-wire, 5-wire, or 8-wire resistive panel. The TSC_ADC_SS can be configured for use in one of the following applications:

- 8 general-purpose ADC channels
- 4-wire TSC with 4 general-purpose ADC channels
- 5-wire TSC with 3 general-purpose ADC channels
- 8-wire TSC

11.1.1 TSC_ADC (ADC0) Features

The main features of the TSC_ADC_SS include:

- Support for 4-wire, 5-wire, and 8-wire resistive TS panels
- Support for interleaving TS capture and general-purpose ADC modes
- Programmable FSM sequencer that supports 16 steps:
  - Software register bit for start of conversion
  - Optional start of conversion HW synchronized to Pen touch or external HW event (but not both)
  - Single conversion (one-shot)
  - Continuous conversions
  - Sequence through all input channels based on a mask
  - Programmable OpenDelay for each FSM step
  - Programmable sampling delay for each FSM step
  - Programmable averaging of input samples: 16/8/4/2/1
  - Differential or singled ended mode setting for each FSM step
  - Store data in either of two FIFO groups
  - Option to encode step ID number with data
  - Support for servicing FIFOs via DMA or CPU
  - Programmable DMA Request event (for each FIFO)
  - Dynamically enable or disable channel inputs during operation
  - Stop bit to end conversion
- Support for the following interrupts and status, with masking:
  - Interrupt for HW pen event (Pen-down)
  - Interrupt for HW Pen-up event
  - Interrupt after a sequence of conversions (all non-masked channels)
  - Interrupt for FIFO threshold levels
  - Interrupt if sampled data is out of a programmable range
  - Interrupt for FIFO overflow and underflow conditions
  - Status bit to indicate if ADC is busy converting

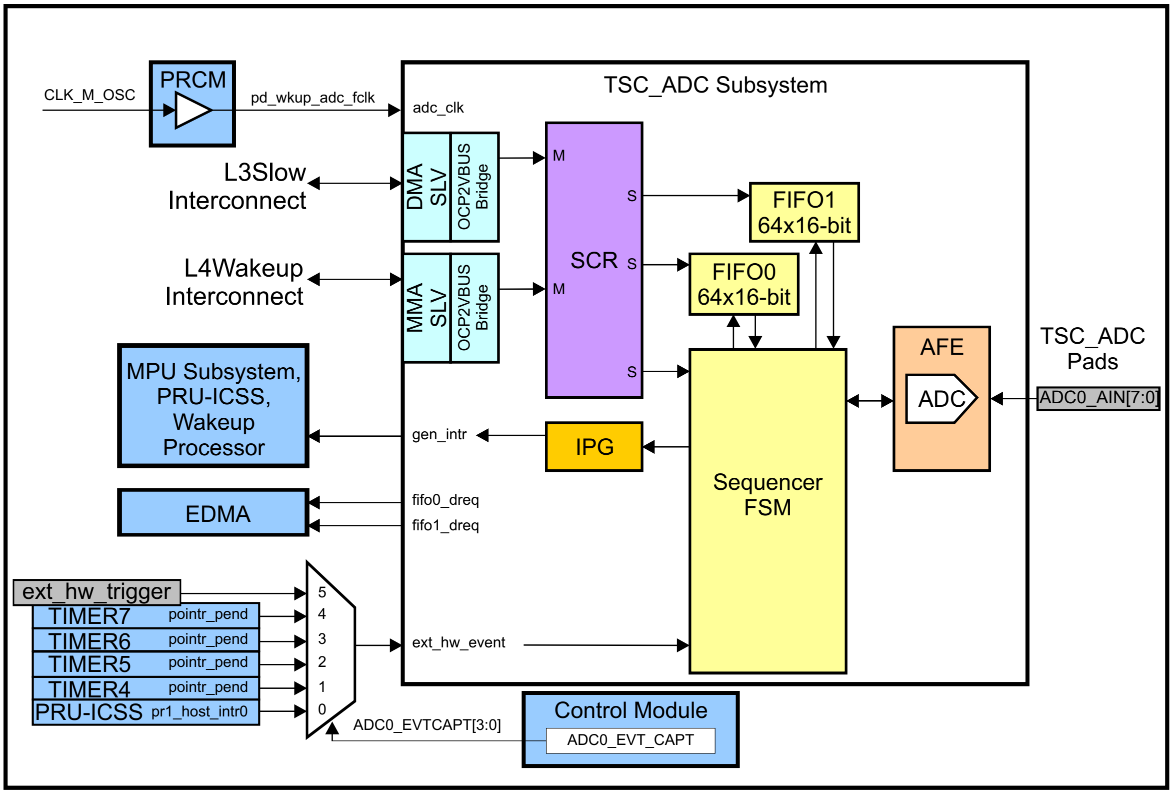
11.1.2 Unsupported TSC_ADC_SS (ADC0) Features

This device supports all TSC_ADC_SS features.

11.2 Integration

Figure 11-1 shows the integration of the TSC_ADC (ADC0) module in the device.

Figure 11-1. TSC_ADC (ADC0) Integration



pr1_host_intr[0:7] corresponds to Host-2 to Host-9 of the PRU-ICSS interrupt controller.

11.2.1 TSC_ADC (ADC0) Connectivity Attributes

The general connectivity attributes for the TSC_ADC module are summarized in Table 11-1.

Table 11-1. TSC_ADC (ADC0) Connectivity Attributes

Attributes	Type
Power domain	Wakeup Domain
Clock domain	PD_PER_L3S_GCLK (OCP) PD_WKUP_ADC_FCLK (Func)
Reset signals	WKUP_DOM_RST_N
Idle/Wakeup signals	Smart idle Wakeup
Interrupt request	1 interrupt to MPU Subsystem (ADC0_GENINT), PRU-ICSS (gen_intr_pend), and Wakeup Processor Swakeup to Wake Processor
DMA request	2 Events (adc0_FIFO0, adc0_FIFO1)
Physical address	L3 Slow slave port (DMA) L4 Wkup slave port (MMR)

11.2.2 TSC_ADC (ADC0) Clock and Reset Management

The TSC_ADC has two clock domains. The ADC uses the adc_clk. The bus interfaces, FIFOs, sequencer, and all other lofic use the ocp_clk.

Table 11-2. TSC_ADC (ADC0) Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
ocp_clk OCP / Functional clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_ls3_gclk From PRCM
adc_clk ADC clock	24 MHz (typ)	CLK_M_OSC	pd_wkup_adc_fclk From PRCM

11.2.3 TSC_ADC (ADC0) Pin List

The TSC_ADC external interface signals are shown in Table 11-3.

Table 11-3. TSC_ADC (ADC0) Pin List

Pin	Type	Description
ADC0_AIN[7:0]	I	Analog Input
VREFN	Analog	Analog Reference Input Negative Terminal
VREFP	Analog	Analog Reference Input Positive Terminal

11.3 Functional Description

Before enabling the TSC_ADC_SS module, the user must first program the Step Configuration registers in order to configure a channel input to be sampled. There are 16 programmable Step Configuration registers which are used by the sequencer to control which switches to turn on or off (inputs to the AFE), which channel to sample, and which mode to use (hardware-triggered or software-enabled, one-shot or continuous, averaging, where to save the FIFO data, and more).

11.3.1 Hardware Synchronized or Software Enabled

The user can control the start behavior of each step by deciding if a channel should be sampled immediately (software-enabled) after it is enabled, or if the channel should wait for a hardware (HW) event to occur first (a HW event must either be mapped to the touch screen PEN event or mapped to the HW event input signal, but not both). Each step can be configured independently using the STEPCONFIG_x register.

11.3.2 Open Delay and Sample Delay

The user can program the delay between driving the inputs to the AFE and the time to send the start of conversion signal. This delay can be used to allow the voltages to stabilize on the touch screen panel before sampling. This delay is called “open delay” and can also be programmed to zero. The user also has control of the sampling time (width of the start of conversion signal) to the AFE which is called the “sample delay”. The open delay and sample delay for each step can be independently configured using the STEPDELAY_x register.

11.3.3 Averaging of Samples (1, 2, 4, 8, and 16)

Each step has the capability to average the sampled data. The valid averaging options are 1 (no average), 2, 4, 8, and 16. If averaging is turned on, then the channel is immediately sampled again (up to 16 times) and final averaged sample data is stored in the FIFO. Each step can be independently configured using the STEPCONFIG_x registers.

11.3.4 One-Shot (Single) or Continuous Mode

When the sequencer finishes cycling through all the enabled steps, the user can decide if the sequencer should stop (one-shot), or loop back and schedule the step again (continuous).

If one-shot mode is selected, the sequencer will take care of disabling the step enable bit after the conversion. If continuous mode is selected, it is the software’s responsibility to turn off the step enable bit.

11.3.5 Interrupts

The following interrupts are supported through enable bits and are maskable.

The HW Pen event interrupt, also known as the Pen-down interrupt, is generated when the user presses the touchscreen. This can only occur if the AFE is configured properly (that is, one of the Pen Ctrl bits must be enabled, and also the correct setting for a path to ground in the STEPCONFIG_x registers). Although the Pen-down interrupt can be disabled by the software (SW), the event will still trigger the sequencer to start if the step is configured as a HW-synchronized event. The Pen-down interrupt is an asynchronous event and can be used even if the TSC_ADC_SS clocks are disabled. The Pen-down interrupt can be used as a wakeup source.

An END_OF_SEQUENCE interrupt is generated after the sequencer finishes servicing the last enabled step.

A Pen-up event interrupt, also known as the Pen-up interrupt, can only be generated when using HW steps with the charge steps enabled. If a Pen-down event caused the HW steps to be scheduled and no Pen-down is present after the sequencer finished servicing the charge step, then a Pen-up interrupt is generated. To detect Pen-up interrupts, the charge step must share the same configuration as the idle step.

Each FIFO has support for generating interrupts when the FIFO word count has reached a programmable threshold level. The user can program the desired word count at which the CPU should be interrupted. Whenever the threshold counter value is reached, it sets the FIFOTHR_x interrupt flag, and the CPU is interrupted if the FIFOTHR_x interrupt enable bit is set. The user can clear the interrupt flag, after emptying the FIFO, by writing a ‘1’ to the FIFOTHR_x interrupt status bit. To determine how many samples are currently in the FIFO at a given moment, the FIFOCOUNT_x register can be read by the CPU.

The FIFO can also generate FIFOx_OVERRUN and FIFOx_UNDERFLOW interrupts. The user can mask these events by programming the IRQEN_CLR register. To clear a FIFO underflow or FIFO overrun interrupt, the user should write a ‘1’ to the status bit in the IRQSTS register. The TSC_ADC_SS does not recover from these conditions automatically. Therefore, the software will need to disable and then again enable the TSC_ADC_SS. Before the user can turn the module back on, the user must read the ADCSTAT register to check if the status of the ADC FSM is idle and the current step is the idle step.

11.3.6 DMA Requests

Each FIFO group can be serviced by either a DMA or by the CPU. To generate DMA requests, the user must set the enable bit in the DMAEN_SET Register. Also, the user can program the desired number of words to generate a DMA request using the DMAxREQ register. When the FIFO level reaches or exceeds that value, a DMA request is generated.

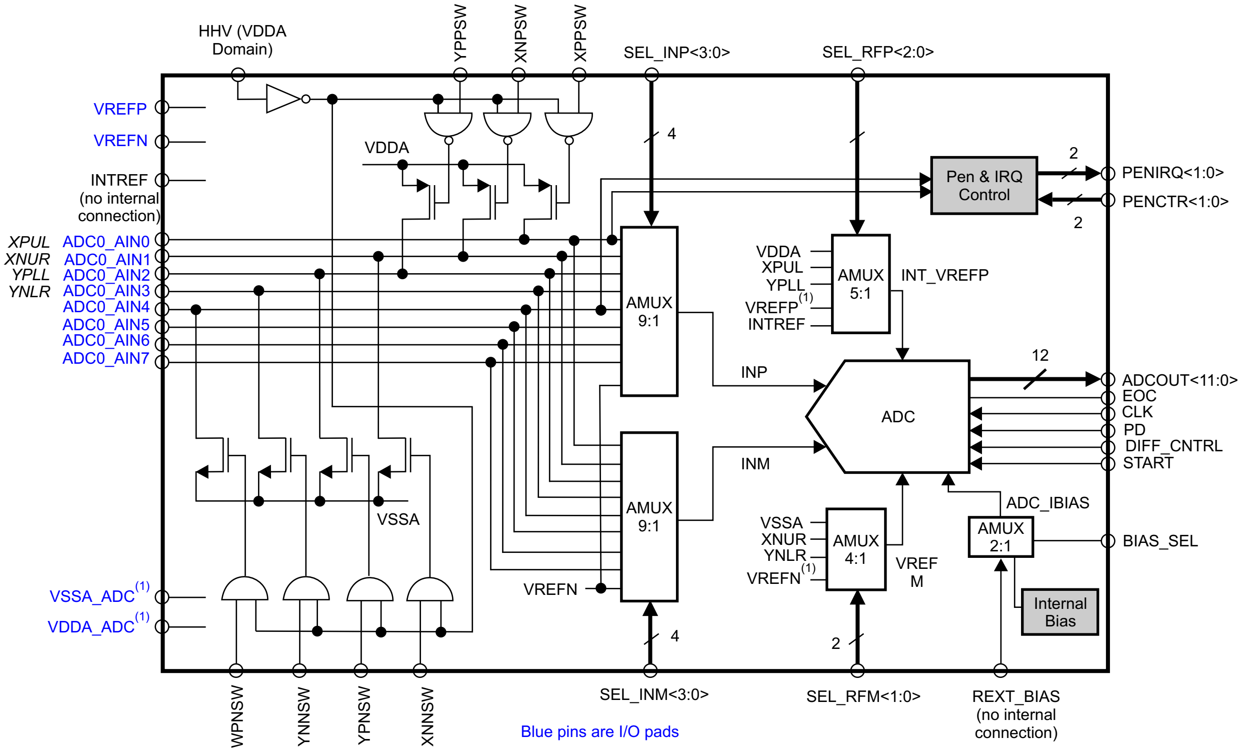
The DMA slave port allows for burst reads in order to effectively move the FIFO data. Internally, the OCP DMA address (MSB) is decoded for either FIFO 0 or FIFO 1. The lower bits of the DMA addresses are ignored since the FIFO pointers are incremented internally.

11.3.7 Analog Front End (AFE) Functional Block Diagram

The AFE features are listed below, and some are controlled by the TSC_ADC_SS:

- 12-bit ADC
- Sampling rate can be as fast as every 15 ADC clock cycles
- Support for internal ADC clock divider logic
- Support for configuring the delay between samples also the sampling time

Figure 11-2. Functional Block Diagram



(1) In the device-specific datasheet:

- VDDA_ADC and VSSA_ADC are referred to as "Internal References"
- VREFP and VREFN are referred to as "External References"

11.4 Operational Modes

The sequencer is completely controlled by software and behaves accordingly to how the **Step Registers** are programmed. A **step** is the general term for sampling a channel input. It is defined by the programmer who decides which input values to send to the AFE as well as how and when to sample a channel input.

The choices for each step can all be programmed using the STEPCONFIG_x registers.

A step requires using these registers:

- STEPEN: Enables or disables the step
- STEPCONFIG_x: Controls the input values to the ADC (the reference voltages, the pull up/down transistor biasing, which input channel to sample, differential control, HW synchronized or SW enabled, averaging, and which FIFO group to save the data)
- STEPDELAY_x: Controls the OpenDelay (the time between driving the AFE inputs until sending the SOC signal to the AFE), and also controls the SampleDelay (the time for the ADC to sample the input signal)

The sequencer supports a total of 16 programmable steps, a touchscreen charge step, and an idle step. Each step requires using the registers listed above. However, the idle step does not have an enable bit, so it will always be enabled, or a delay register. In addition, the ADC does not actually sample a channel during the idle and touchscreen charge steps.

Assuming all the steps are configured as general-purpose mode (no touchscreen), then the steps would be configured as SW enabled. When the TSC_ADC_SS is first enabled, the sequencer will always start in the Idle step and then wait for a STEPEN[n] bit to turn on. After a step is enabled, the sequencer will start with the lowest step (1) and continue until step (16). If a step is not enabled, then sequencer will skip to the next step. If all steps are disabled, then the sequencer will remain in the IDLE state and continue to apply the settings in the IDLECONFIG register.

Assuming a touchscreen-only mode (no general-purpose channels) the steps could be configured as HW synchronized triggered (mapped to the Pen event). The sequencer would wait in the IDLE state until a Pen-down event occurred and then begin the HW step conversions. The charge step, which occurs after the last HW-synchronized step is finished, is designed to charge the capacitance in a touch panel when the appropriate bias transistor is enabled. The purpose of the charge step is to prepare the TSC for the next Pen-down event.

Assuming a mixed mode application (touchscreen and general-purpose channels), the user can configure the steps as either HW-triggered (mapped to a Pen event) or SW-enabled. If the sequencer is in the idle state and a Pen-down event occurs, the HW-synchronized steps are always scheduled first, followed by the charge step, and cannot be preempted by SW steps. If there is no HW event, then the SW-enabled steps are scheduled instead.

If a Pen-down event occurs while the sequencer is in the middle of scheduling the SW steps, the user can program the scheduler to allow preemption. If the HW preemption control bit is enabled in the CTRL register, the sequencer will stop the scheduled SW sequence and schedule the HW steps. After the last HW step and charge step are completed, the sequencer will continue from the next SW step (before the preemption occurred). If the HW preemption is disabled, then the touch event will be ignored until the last software step is completed; if the touch event is removed before the last software step is finished, then the touch event will be missed.

Even if a touchscreen is not present, the user can still configure the steps to be HW-synchronized by mapping to the ext_hw_event signal shown in Figure 11-1. This HW event input signal can be driven at the SOC from a number of external inputs chosen by the ADC0_EVT_CAPT register in the Control Module.

When mapping is set for the ext_hw_event signal, then the TSC_ADC_SS will wait for a rising edge transition (from low to high) before starting. The ext_hw_event signal is captured on the internal L4 OCP clock domain. The ext_hw_event signal should be held for at least two TSC_ADC_SS OCP clocks (L4 frequency).

An END_OF_SEQUENCE interrupt is generated after the last active step is completed before going back to the IDLE state. The END_OF_SEQUENCE interrupt does not mean data is in the FIFO (should use the FIFO interrupts and FIFOCOUNT_x register).

11.4.1 PenCtrl and PenIRQ

The Pen IRQ can only occur if the correct AFE_Pen_Ctrl bits are high in the CTRL register and if the correct ground transistor biasing is set in the STEPCONFIG_x and IDLECONFIG registers.

Setting the AFE_Pen_Ctrl bits in the CTRL register will enable a weak internal pull-up resistor on ADC0_AIN for 4-wire configurations and ADC0_AIN for 5-wire.

If a step is configured as HW-synchronized, the sequencer will override the AFE_Pen_Ctrl bits set by the software (bits 6:5) once it transitions from the Idle step. The sequencer will automatically mask the AFE_Pen_Ctrl bits (override them and turn them off) so that the ADC can get an accurate measurement from the x and y-axes. After the last HW-synchronized step, the sequence will go to the Charge step and the pen override mask is removed and the values set by the software (bits 6:5) will have control. The Pen-down events will be temporarily ignored during the Charge step (HW will mask any potential glitches that may occur)

If the sequencer is not using the HW synchronized approach, (all the steps are configured as software enabled), then it is the software programmer’s responsibility to correctly turn on and off the AFE_Pen_Ctrl bits to receive the correct measurements from the touchscreen. The software must enable the Charge step and ignore any potential glitches.

It is also possible to detect the Pen-down event even if all the STEPEN[n] bits are off. By setting the appropriate AFE_Pen_Ctrl bit to 1, and configuring the IDLECONFIG register to bias the correct transistor to ground, the Pen-down event will generate. The flowchart for the sequencer is shown in Figure 11-3 and an example timing diagram in Figure 11-4.

Figure 11-3. Sequencer FSM

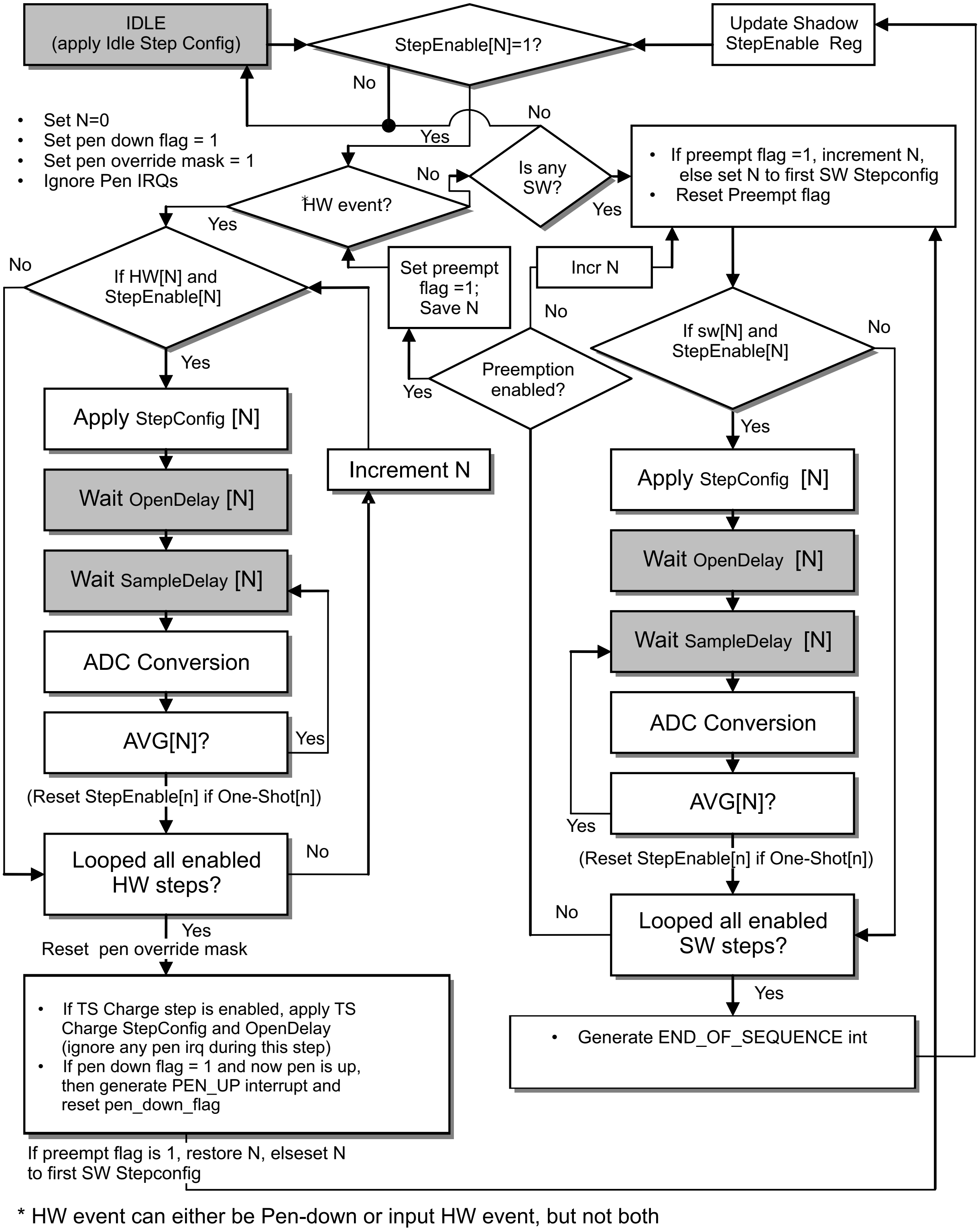
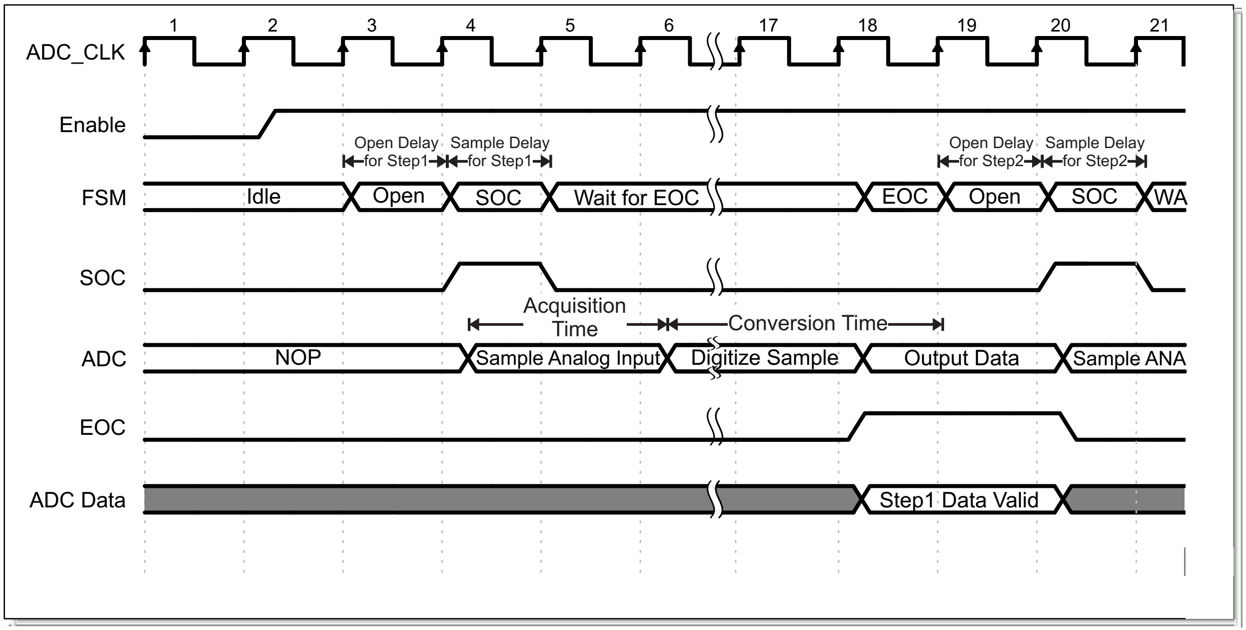


Figure 11-3 does not actually represent clock cycles but instead illustrates how the scheduler will work. However, each shaded box above does represent a FSM state and will use a clock cycle. Using the minimum delay values, the ADC can sample at 15 ADC clocks per sample. Below is an example timing diagram illustrating the states of the sequencer and also the showing when the STEPCONFIG_x and the STEPDELAY_x registers values are applied. The below example assumes the steps are software controlled, and averaging is turned off.

Figure 11-4. Example Timing Diagram for Sequencer



The idle step is always enabled and applied when the FSM is in the IDLE state (that is, either waiting for a HW event or waiting for a step to be enabled). The idle step cannot be disabled.

Once the TSC_ADC_SS is enabled and assuming at least one step is active, the FSM will transition from the idle state and apply the first active STEPCONFIG_x and STEPDELAY_x register settings. It is possible for the Open Delay to be 0, and, as a result, the FSM will immediately skip to the Sample Delay. The ADC will begin sampling once the SoC signal goes high. The ADC will continue to sample for at least 1 clock cycle, which is the Sample Delay minimum, after the SoC. The ADC will then take 12 cycles to digitize the sample followed by 1 cycle to send the EOC for a minimum total of 15 ADC_CLK cycles per sample. Once the EOC has been sent, the FSM will begin the next active step.

This process is repeated and continued (from step 1 to step 16) until the last active step is completed.

11.5 ADC0 Registers

11.5.1 ADC0 Registers

Table 11-4 lists the memory-mapped registers for the ADC0. All register offset addresses not listed in Table 11-4 should be considered as reserved locations and the register contents should not be modified.

Table 11-4. ADC0 Registers

Offset	Acronym	Register Name	Section
0h	ADC0_REVISION		Section 11.5.1.1
10h	ADC0_SYSCONFIG		Section 11.5.1.2
24h	ADC0_IRQSTS_RAW		Section 11.5.1.3
28h	ADC0_IRQSTS		Section 11.5.1.4
2Ch	ADC0_IRQEN_SET		Section 11.5.1.5
30h	ADC0_IRQEN_CLR		Section 11.5.1.6
34h	ADC0_IRQWAKEUP		Section 11.5.1.7
38h	ADC0_DMAEN_SET		Section 11.5.1.8
3Ch	ADC0_DMAEN_CLR		Section 11.5.1.9
40h	ADC0_CTRL		Section 11.5.1.10
44h	ADC0_ADCSTAT		Section 11.5.1.11
48h	ADC0_ADCRANGE		Section 11.5.1.12
4Ch	ADC0_ADC_CLKDIV		Section 11.5.1.13
50h	ADC0_ADC_MISC		Section 11.5.1.14
54h	ADC0_STEPEN		Section 11.5.1.15
58h	ADC0_IDLECONFIG		Section 11.5.1.16
5Ch	ADC0_TS_CHARGE_STEPCONFIG		Section 11.5.1.17
60h	ADC0_TS_CHARGE_DELAY		Section 11.5.1.18
64h	ADC0_STEPCONFIG_0		Section 11.5.1.19
68h	ADC0_STEPDELAY_0		Section 11.5.1.20
6Ch	ADC0_STEPCONFIG_1		Section 11.5.1.19
70h	ADC0_STEPDELAY_1		Section 11.5.1.20
74h	ADC0_STEPCONFIG_2		Section 11.5.1.19
78h	ADC0_STEPDELAY_2		Section 11.5.1.20
7Ch	ADC0_STEPCONFIG_3		Section 11.5.1.19
80h	ADC0_STEPDELAY_3		Section 11.5.1.20
84h	ADC0_STEPCONFIG_4		Section 11.5.1.19
88h	ADC0_STEPDELAY_4		Section 11.5.1.20
8Ch	ADC0_STEPCONFIG_5		Section 11.5.1.19
90h	ADC0_STEPDELAY_5		Section 11.5.1.20
94h	ADC0_STEPCONFIG_6		Section 11.5.1.19
98h	ADC0_STEPDELAY_6		Section 11.5.1.20
9Ch	ADC0_STEPCONFIG_7		Section 11.5.1.19
A0h	ADC0_STEPDELAY_7		Section 11.5.1.20
A4h	ADC0_STEPCONFIG_8		Section 11.5.1.19
A8h	ADC0_STEPDELAY_8		Section 11.5.1.20
ACh	ADC0_STEPCONFIG_9		Section 11.5.1.19
B0h	ADC0_STEPDELAY_9		Section 11.5.1.20
B4h	ADC0_STEPCONFIG_10		Section 11.5.1.19
B8h	ADC0_STEPDELAY_10		Section 11.5.1.20
BCh	ADC0_STEPCONFIG_11		Section 11.5.1.19
C0h	ADC0_STEPDELAY_11		Section 11.5.1.20

Table 11-4. ADC0 Registers (continued)

Offset	Acronym	Register Name	Section
C4h	ADC0_STEPCONFIG_12		Section 11.5.1.19
C8h	ADC0_STEPDELAY_12		Section 11.5.1.20
CCh	ADC0_STEPCONFIG_13		Section 11.5.1.19
D0h	ADC0_STEPDELAY_13		Section 11.5.1.20
D4h	ADC0_STEPCONFIG_14		Section 11.5.1.19
D8h	ADC0_STEPDELAY_14		Section 11.5.1.20
DCh	ADC0_STEPCONFIG_15		Section 11.5.1.19
E0h	ADC0_STEPDELAY_15		Section 11.5.1.20
E4h	ADC0_FIFOCOUNT_0		Section 11.5.1.21
E8h	ADC0_FIFOTHR_0		Section 11.5.1.22
ECh	ADC0_DMAREQ_0		Section 11.5.1.23
F0h	ADC0_FIFOCOUNT_1		Section 11.5.1.21
F4h	ADC0_FIFOTHR_1		Section 11.5.1.22
F8h	ADC0_DMAREQ_1		Section 11.5.1.23
100h	ADC0_FIFO0DATA		Section 11.5.1.24
200h	ADC0_FIFO1DATA		Section 11.5.1.25

11.5.1.1 ADC0_REVISION Register (offset = 0h) [reset = 47300001h]

Register mask: FFFFFFFFh

ADC0_REVISION is shown in Figure 11-5 and described in Table 11-5.

Revision Register

Figure 11-5. ADC0_REVISION Register

31	30	29	28	27	26	25	24
SCHEME		RESERVED		FUNC			
R-1h		R-0h		R-730h			
23	22	21	20	19	18	17	16
FUNC							
R-730h							
15	14	13	12	11	10	9	8
R_RTL					X_MAJOR		
R-0h					R-0h		
7	6	5	4	3	2	1	0
CUSTOM		Y_MINOR					
R-0h		R-1h					

Table 11-5. ADC0_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	Scheme value
29-28	RESERVED	R	0h	Always read as 0. Writes have no affect.
27-16	FUNC	R	730h	Function value
15-11	R_RTL	R	0h	RTL Version value
10-8	X_MAJOR	R	0h	Major Revision value
7-6	CUSTOM	R	0h	Custom Revision value
5-0	Y_MINOR	R	1h	Minor Revision value

11.5.1.2 ADC0_SYSCONFIG Register (offset = 10h) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_SYSCONFIG is shown in Figure 11-6 and described in Table 11-6.

SysConfig Register

Figure 11-6. ADC0_SYSCONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				IDLEMODE		RESERVED	
R/W-0h				R/W-0h		R/W-0h	

Table 11-6. ADC0_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-2	IDLEMODE	R/W	0h	0h (R/W) = Force Idle (always acknowledges) 1h (R/W) = No Idle Mode (never acknowledges) 2h (R/W) = Smart-Idle Mode 3h (R/W) = Smart Idle with Wakeup
1-0	RESERVED	R/W	0h	

11.5.1.3 ADC0_IRQSTS_RAW Register (offset = 24h) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_IRQSTS_RAW is shown in Figure 11-7 and described in Table 11-7.

IRQ status (unmasked)

Figure 11-7. ADC0_IRQSTS_RAW Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED					PEN_IRQ_SYNCHRONIZED	PEN_UP_EVT	OUT_OF_RANGE
R/W-0h					R/W-0h	R/W-0h	R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FIFO1_UNDERFLOW	FIFO1_OVERRUN	FIFO1_THR	FIFO0_UNDERFLOW	FIFO0_OVERRUN	FIFO0_THR	END_OF_SEQUENCE	HW_PEN_EVT_ASYNCHRONOUS
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 11-7. ADC0_IRQSTS_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10	PEN_IRQ_SYNCHRONIZED	R/W	0h	0h (W) = No action 0h (R) = No event pending 1h (W) = Set event (debug) 1h (R) = Event pending
9	PEN_UP_EVT	R/W	0h	0h (W) = No action 0h (R) = No event pending 1h (W) = Set event (debug) 1h (R) = Event pending
8	OUT_OF_RANGE	R/W	0h	0h (W) = No action 0h (R) = No event pending 1h (W) = Set event (debug) 1h (R) = Event pending
7	FIFO1_UNDERFLOW	R/W	0h	0h (W) = No action 0h (R) = No event pending 1h (W) = Set event (debug) 1h (R) = Event pending
6	FIFO1_OVERRUN	R/W	0h	0h (W) = No action 0h (R) = No event pending 1h (W) = Set event (debug) 1h (R) = Event pending
5	FIFO1_THR	R/W	0h	0h (W) = No action 0h (R) = No event pending 1h (W) = Set event (debug) 1h (R) = Event pending

Table 11-7. ADC0_IRQSTS_RAW Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	FIFO0_UNDERFLOW	R/W	0h	0h (W) = No action 0h (R) = No event pending 1h (W) = Set event (debug) 1h (R) = Event pending
3	FIFO0_OVERRUN	R/W	0h	0h (W) = No action 0h (R) = No event pending 1h (W) = Set event (debug) 1h (R) = Event pending
2	FIFO0_THR	R/W	0h	0h (W) = No action 0h (R) = No event pending 1h (W) = Set event (debug) 1h (R) = Event pending
1	END_OF_SEQUENCE	R/W	0h	0h (W) = No action 0h (R) = No event pending 1h (W) = Set event (debug) 1h (R) = Event pending
0	HW_PEN_EVT_ASYNCHRONOUS	R/W	0h	0h (W) = No action 0h (R) = No event pending 1h (W) = Set event (debug) 1h (R) = Event pending

11.5.1.4 ADC0_IRQSTS Register (offset = 28h) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_IRQSTS is shown in Figure 11-8 and described in Table 11-8.

IRQ status (masked)

Figure 11-8. ADC0_IRQSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED					HW_PEN_EVT_SYNCHRONOUS	PEN_UP_EVT	OUT_OF_RANGE
R/W-0h					R/W-0h	R/W-0h	R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FIFO1_UNDERFLOW	FIFO1_OVERRUN	FIFO1_THR	FIFO0_UNDERFLOW	FIFO0_OVERRUN	FIFO0_THR	END_OF_SEQUENCE	HW_PEN_EVT_ASYNCHRONOUS
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 11-8. ADC0_IRQSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10	HW_PEN_EVT_SYNCHRONOUS	R/W	0h	0h (W) = No action 0h (R) = No (enabled) event pending 1h (W) = Clear (raw) event 1h (R) = Event pending
9	PEN_UP_EVT	R/W	0h	0h (W) = No action 0h (R) = No (enabled) event pending 1h (W) = Clear (raw) event 1h (R) = Event pending
8	OUT_OF_RANGE	R/W	0h	0h (W) = No action 0h (R) = No (enabled) event pending 1h (W) = Clear (raw) event 1h (R) = Event pending
7	FIFO1_UNDERFLOW	R/W	0h	0h (W) = No action 0h (R) = No (enabled) event pending 1h (W) = Clear (raw) event 1h (R) = Event pending
6	FIFO1_OVERRUN	R/W	0h	0h (W) = No action 0h (R) = No (enabled) event pending 1h (W) = Clear (raw) event 1h (R) = Event pending
5	FIFO1_THR	R/W	0h	0h (W) = No action 0h (R) = No (enabled) event pending 1h (W) = Clear (raw) event 1h (R) = Event pending

Table 11-8. ADC0_IRQSTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	FIFO0_UNDERFLOW	R/W	0h	0h (W) = No action 0h (R) = No (enabled) event pending 1h (W) = Clear (raw) event 1h (R) = Event pending
3	FIFO0_OVERRUN	R/W	0h	0h (W) = No action 0h (R) = No (enabled) event pending 1h (W) = Clear (raw) event 1h (R) = Event pending
2	FIFO0_THR	R/W	0h	0h (W) = No action 0h (R) = No (enabled) event pending 1h (W) = Clear (raw) event 1h (R) = Event pending
1	END_OF_SEQUENCE	R/W	0h	0h (W) = No action 0h (R) = No (enabled) event pending 1h (W) = Clear (raw) event 1h (R) = Event pending
0	HW_PEN_EVT_ASYNCHRONOUS	R/W	0h	0h (W) = No action 0h (R) = No (enabled) event pending 1h (W) = Clear (raw) event 1h (R) = Event pending

11.5.1.5 ADC0_IRQEN_SET Register (offset = 2Ch) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_IRQEN_SET is shown in Figure 11-9 and described in Table 11-9.

IRQ enable set bits

Figure 11-9. ADC0_IRQEN_SET Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							

23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							

15	14	13	12	11	10	9	8
RESERVED					HW_PEN_EVT_SYNCHRONOUS	PEN_UP_EVT	OUT_OF_RANGE
R/W-0h					R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
FIFO1_UNDERFLOW	FIFO1_OVERRUN	FIFO1_THR	FIFO0_UNDERFLOW	FIFO0_OVERRUN	FIFO0_THR	END_OF_SEQUENCE	HW_PEN_EVT_ASYNCHRONOUS
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 11-9. ADC0_IRQEN_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10	HW_PEN_EVT_SYNCHRONOUS	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Enable interrupt 1h (R) = Interrupt enabled
9	PEN_UP_EVT	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Enable interrupt 1h (R) = Interrupt enabled
8	OUT_OF_RANGE	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Enable interrupt 1h (R) = Interrupt enabled
7	FIFO1_UNDERFLOW	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Enable interrupt 1h (R) = Interrupt enabled
6	FIFO1_OVERRUN	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Enable interrupt 1h (R) = Interrupt enabled
5	FIFO1_THR	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Enable interrupt 1h (R) = Interrupt enabled

Table 11-9. ADC0_IRQEN_SET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	FIFO0_UNDERFLOW	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Enable interrupt 1h (R) = Interrupt enabled
3	FIFO0_OVERRUN	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Enable interrupt 1h (R) = Interrupt enabled
2	FIFO0_THR	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Enable interrupt 1h (R) = Interrupt enabled
1	END_OF_SEQUENCE	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Enable interrupt 1h (R) = Interrupt enabled
0	HW_PEN_EVT_ASYNCHRONOUS	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Enable interrupt 1h (R) = Interrupt enabled

11.5.1.6 ADC0_IRQEN_CLR Register (offset = 30h) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_IRQEN_CLR is shown in Figure 11-10 and described in Table 11-10.

IRQ enable clear bits

Figure 11-10. ADC0_IRQEN_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED					HW_PEN_EVT_SYNCHRONOUS	PEN_UP_EVT	OUT_OF_RANGE
R/W-0h					R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
FIFO1_UNDERFLOW	FIFO1_OVERRUN	FIFO1_THR	FIFO0_UNDERFLOW	FIFO0_OVERRUN	FIFO0_THR	END_OF_SEQUENCE	HW_PEN_EVT_ASYNCHRONOUS
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 11-10. ADC0_IRQEN_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10	HW_PEN_EVT_SYNCHRONOUS	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Disable interrupt 1h (R) = Interrupt enabled
9	PEN_UP_EVT	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Disable interrupt 1h (R) = Interrupt enabled
8	OUT_OF_RANGE	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Disable interrupt 1h (R) = Interrupt enabled
7	FIFO1_UNDERFLOW	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Disable interrupt 1h (R) = Interrupt enabled
6	FIFO1_OVERRUN	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Disable interrupt 1h (R) = Interrupt enabled
5	FIFO1_THR	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Disable interrupt 1h (R) = Interrupt enabled

Table 11-10. ADC0_IRQEN_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	FIFO0_UNDERFLOW	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Disable interrupt 1h (R) = Interrupt enabled
3	FIFO0_OVERRUN	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Disable interrupt 1h (R) = Interrupt enabled
2	FIFO0_THR	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Disable interrupt 1h (R) = Interrupt enabled
1	END_OF_SEQUENCE	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Disable interrupt 1h (R) = Interrupt enabled
0	HW_PEN_EVT_ASYNCHRONOUS	R/W	0h	0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Disable interrupt 1h (R) = Interrupt enabled

11.5.1.7 ADC0_IRQWAKEUP Register (offset = 34h) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_IRQWAKEUP is shown in Figure 11-11 and described in Table 11-11.

IRQ wakeup enable

Figure 11-11. ADC0_IRQWAKEUP Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED							WAKEEN0
R/W-0h							R/W-0h

Table 11-11. ADC0_IRQWAKEUP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	WAKEEN0	R/W	0h	Wakeup generation for HW Pen event. 0h (R/W) = Wakeup disabled 1h (R/W) = Wakeup enabled

11.5.1.8 ADC0_DMAEN_SET Register (offset = 38h) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_DMAEN_SET is shown in Figure 11-12 and described in Table 11-12.

Per-Line DMA set

Figure 11-12. ADC0_DMAEN_SET Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED						EN_1	EN_0
R/W-0h						R/W-0h	R/W-0h

Table 11-12. ADC0_DMAEN_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	EN_1	R/W	0h	Enable DMA request FIFO 1. 0h (W) = No action 0h (R) = DMA line disabled 1h (W) = Enable DMA line 1h (R) = DMA line enabled
0	EN_0	R/W	0h	Enable DMA request FIFO 0. 0h (W) = No action 0h (R) = DMA line disabled 1h (W) = Enable DMA line 1h (R) = DMA line enabled

11.5.1.9 ADC0_DMAEN_CLR Register (offset = 3Ch) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_DMAEN_CLR is shown in Figure 11-13 and described in Table 11-13.

Per-Line DMA clr

Figure 11-13. ADC0_DMAEN_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED						EN_1	EN_0
R/W-0h						R/W-0h	R/W-0h

Table 11-13. ADC0_DMAEN_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	EN_1	R/W	0h	Disable DMA request FIFO 1. 0h (W) = No action 0h (R) = DMA line disabled 1h (W) = Disable DMA line 1h (R) = DMA line enabled
0	EN_0	R/W	0h	Disable DMA request FIFO 0. 0h (W) = No action 0h (R) = DMA line disabled 1h (W) = Disable DMA line 1h (R) = DMA line enabled

11.5.1.10 ADC0_CTRL Register (offset = 40h) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_CTRL is shown in Figure 11-14 and described in Table 11-14.

Control Register

Figure 11-14. ADC0_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
Rreturns0s-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED						HW_PREEMPT	HW_EVT_MAPPING
Rreturns0s-0h						R/W-0h	R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOUCH_SCREEN_EN	AFE_PEN_CTRL		POWER_DOWN	ADC_BIAS_SELECT	STEPCONFIG_WRITEPROTECT_N	STEP_ID_TAG	EN
R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 11-14. ADC0_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	Rreturns0s	0h	
9	HW_PREEMPT	R/W	0h	0h (R/W) = SW steps are not pre-empted by HW events 1h (R/W) = SW steps are pre-empted by HW events
8	HW_EVT_MAPPING	R/W	0h	0h (R/W) = Map HW event to Pen touch irq (from AFE) 1h (R/W) = Map HW event to HW event input
7	TOUCH_SCREEN_EN	R/W	0h	0h (R/W) = Touchscreen transistors disabled 1h (R/W) = Touchscreen transistors enabled
6-5	AFE_PEN_CTRL	R/W	0h	These two bits are sent directly to the AFE Pen Ctrl inputs. Bit 6 controls the Wiper touch (5 wire modes). Bit 5 controls the X+ touch (4 wire modes). User also needs to make sure the ground path is connected properly for pen interrupt to occur (using the StepConfig registers). Refer to section 4 interrupts for more information.
4	POWER_DOWN	R/W	0h	ADC Power Down control. 0h (R/W) = AFE is powered up (default) 1h (R/W) = Write 1 to power down AFE (the tsc_adc_ss enable (bit 0) should also be set to off)
3	ADC_BIAS_SELECT	R/W	0h	Select Bias to AFE. 0 = Internal. 1 = Reserved.
2	STEPCONFIG_WRITEPROTECT_N	R/W	0h	StepConfig_WriteProtect_n is active low. 0h (R/W) = Step configuration registers are protected (not writable) 1h (R/W) = Step configuration registers are not protected (writable)
1	STEP_ID_TAG	R/W	0h	Writing 1 to this bit will store the Step ID number with the captured ADC data in the FIFO. 0h (R/W) = Write zeroes 1h (R/W) = Store the channel ID tag

Table 11-14. ADC0_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	EN	R/W	0h	TSC_ADC_SS module enable bit. After programming all the steps and configuration registers, write a 1 to this bit to turn on TSC_ADC_SS. Writing a 0 will disable the module (after the current conversion). 0h (R/W) = Disable 1h (R/W) = Enable

11.5.1.11 ADC0_ADCSTAT Register (offset = 44h) [reset = 10h]

Register mask: FFFFFFFFh

ADC0_ADCSTAT is shown in Figure 11-15 and described in Table 11-15.

General Status bits for Sequencer Status

Figure 11-15. ADC0_ADCSTAT Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
Rreturns0s-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
Rreturns0s-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PEN_IRQ1	PEN_IRQ0	FSM_BUSY	STEP_ID				
R-0h	R-0h	R-0h	R-10h				

Table 11-15. ADC0_ADCSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	Rreturns0s	0h	-
7	PEN_IRQ1	R	0h	PEN_IRQ[1] status
6	PEN_IRQ0	R	0h	PEN_IRQ[0] status
5	FSM_BUSY	R	0h	Status of OCP FSM and ADC FSM. 0h (R/W) = Idle 1h (R/W) = Busy
4-0	STEP_ID	R	10h	Encoded values: 0h (R/W) = Step 1 1h (R/W) = Step 2 2h (R/W) = Step 3 3h (R/W) = Step 4 4h (R/W) = Step 5 5h (R/W) = Step 6 6h (R/W) = Step 7 7h (R/W) = Step 8 8h (R/W) = Step 9 9h (R/W) = Step 10 Ah (R/W) = Step 11 Bh (R/W) = Step 12 Ch (R/W) = Step 13 Dh (R/W) = Step 14 Eh (R/W) = Step 15 Fh (R/W) = Step 16 10h (R/W) = Idle 11h (R/W) = Charge

11.5.1.12 ADC0_ADCRANGE Register (offset = 48h) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_ADCRANGE is shown in Figure 11-16 and described in Table 11-16.

High and Low Range Threshold for ADC Range Check

Figure 11-16. ADC0_ADCRANGE Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="12">HIGH_RANGE_DATA</td></tr>
<tr><td colspan="4">Rreturns0s-0h</td><td colspan="12">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="12">LOW_RANGE_DATA</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="12">R/W-0h</td></tr>
</table>

Table 11-16. ADC0_ADCRANGE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	Rreturns0s	0h	
27-16	HIGH_RANGE_DATA	R/W	0h	Sampled ADC data is compared to this value. If the sampled data is greater than the value, then an interrupt is generated.
15-12	RESERVED	R/W	0h	Reserved.
11-0	LOW_RANGE_DATA	R/W	0h	Sampled ADC data is compared to this value. If the sampled data is less than the value, then an interrupt is generated.

11.5.1.13 ADC0_ADC_CLKDIV Register (offset = 4Ch) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_ADC_CLKDIV is shown in Figure 11-17 and described in Table 11-17.

ADC clock divider register

Figure 11-17. ADC0_ADC_CLKDIV Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">ADC_CLKDIV</td></tr>
<tr><td colspan="16">Rreturns0s-0h</td><td colspan="16">R/W-0h</td></tr>
</table>

Table 11-17. ADC0_ADC_CLKDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	Rreturns0s	0h	
15-0	ADC_CLKDIV	R/W	0h	The input ADC clock will be divided by this value and sent to the AFE. Program to the value minus 1

11.5.1.14 ADC0_ADC_MISC Register (offset = 50h) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_ADC_MISC is shown in Figure 11-18 and described in Table 11-18.

AFE misc debug

Figure 11-18. ADC0_ADC_MISC Register

31	30	29	28	27	26	25	24
RESERVED							
Rreturns0s-0h							
23	22	21	20	19	18	17	16
RESERVED							
Rreturns0s-0h							
15	14	13	12	11	10	9	8
RESERVED							
Rreturns0s-0h							
7	6	5	4	3	2	1	0
AFE_SPARE_OUTPUT				AFE_SPARE_INPUT			
R-0h				R/W-0h			

Table 11-18. ADC0_ADC_MISC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	Rreturns0s	0h	RESERVED.
7-4	AFE_SPARE_OUTPUT	R	0h	Connected to AFE Spare Output pins. Reserved in normal operation.
3-0	AFE_SPARE_INPUT	R/W	0h	Connected to AFE Spare Input pins. Reserved in normal operation.

11.5.1.15 ADC0_STEPEN Register (offset = 54h) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_STEPEN is shown in Figure 11-19 and described in Table 11-19.

Step Enable

Figure 11-19. ADC0_STEPEN Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">Rreturns0s-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="7">RESERVED</td><td>STEP16</td></tr>
<tr><td colspan="7">Rreturns0s-0h</td><td>R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>STEP15</td><td>STEP14</td><td>STEP13</td><td>STEP12</td><td>STEP11</td><td>STEP10</td><td>STEP9</td><td>STEP8</td></tr>
<tr><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>STEP7</td><td>STEP6</td><td>STEP5</td><td>STEP4</td><td>STEP3</td><td>STEP2</td><td>STEP1</td><td>TS_CHARGE</td></tr>
<tr><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 11-19. ADC0_STEPEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	Rreturns0s	0h	RESERVED.
16	STEP16	R/W	0h	Enable step 16
15	STEP15	R/W	0h	Enable step 15
14	STEP14	R/W	0h	Enable step 14
13	STEP13	R/W	0h	Enable step 13
12	STEP12	R/W	0h	Enable step 12
11	STEP11	R/W	0h	Enable step 11
10	STEP10	R/W	0h	Enable step 10
9	STEP9	R/W	0h	Enable step 9
8	STEP8	R/W	0h	Enable step 8
7	STEP7	R/W	0h	Enable step 7
6	STEP6	R/W	0h	Enable step 6
5	STEP5	R/W	0h	Enable step 5
4	STEP4	R/W	0h	Enable step 4
3	STEP3	R/W	0h	Enable step 3
2	STEP2	R/W	0h	Enable step 2
1	STEP1	R/W	0h	Enable step 1
0	TS_CHARGE	R/W	0h	Enable TS Charge step

11.5.1.16 ADC0_IDLECONFIG Register (offset = 58h) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_IDLECONFIG is shown in Figure 11-20 and described in Table 11-20.

Idle Step configuration

Figure 11-20. ADC0_IDLECONFIG Register

31	30	29	28	27	26	25	24
RESERVED						DIFF_CNTRL	SEL_RFM_SWC
R/W-0h						R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
SEL_RFM_SWC	SEL_INP_SWC				SEL_INM_SWM		
R/W-0h	R/W-0h				R/W-0h		
15	14	13	12	11	10	9	8
SEL_INM_SWM	SEL_RFP_SWC			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
YPPSW_SWC	XNNSW_SWC	XPPSW_SWC	RESERVED				
R/W-0h	R/W-0h	R/W-0h	R/W-0h				

Table 11-20. ADC0_IDLECONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R/W	0h	
25	DIFF_CNTRL	R/W	0h	Differential Control Pin. 0h (R/W) = Single Ended 1h (R/W) = Differential Pair Enable
24-23	SEL_RFM_SWC	R/W	0h	SEL_RFM pins SW configuration. 0h (R/W) = VSSA_ADC 1h (R/W) = XNUR 2h (R/W) = YNLR 3h (R/W) = VREFN
22-19	SEL_INP_SWC	R/W	0h	SEL_INP pins SW configuration. Note values 1xxx (binary) = VREFN. 0h (R/W) = Channel 1 1h (R/W) = Channel 2 2h (R/W) = Channel 3 3h (R/W) = Channel 4 4h (R/W) = Channel 5 5h (R/W) = Channel 6 6h (R/W) = Channel 7 7h (R/W) = Channel 8 8h (R/W) = VREFN

Table 11-20. ADC0_IDLECONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18-15	SEL_INM_SWM	R/W	0h	SEL_INM pins for neg differential. Note all values 1xxx (binary) = VREFN. 0h (R/W) = Channel 1 1h (R/W) = Channel 2 2h (R/W) = Channel 3 3h (R/W) = Channel 4 4h (R/W) = Channel 5 5h (R/W) = Channel 6 6h (R/W) = Channel 7 7h (R/W) = Channel 8 8h (R/W) = VREFN
14-12	SEL_RFP_SWC	R/W	0h	SEL_RFP pins SW configuration. Note all values 1xx (binary) = Reserved. 0h (R/W) = VDDA_ADC 1h (R/W) = XPUL 2h (R/W) = YPLL 3h (R/W) = VREFP
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW_SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW_SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-0	RESERVED	R/W	0h	

11.5.1.17 ADC0_TS_CHARGE_STEPCONFIG Register (offset = 5Ch) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_TS_CHARGE_STEPCONFIG is shown in Figure 11-21 and described in Table 11-21.

TS Charge StepConfiguration

Figure 11-21. ADC0_TS_CHARGE_STEPCONFIG Register

31	30	29	28	27	26	25	24
RESERVED						DIFF_CNTRL	SEL_RFM_SWC
R/W-0h						R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SEL_RFM_SWC	SEL_INP_SWC				SEL_INM_SWM		
R/W-0h	R/W-0h				R/W-0h		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SEL_INM_SWM	SEL_RFP_SWC			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
YPPSW__SWC	XNNSW__SWC	XPPSW_SWC	RESERVED				
R/W-0h	R/W-0h	R/W-0h	R/W-0h				

Table 11-21. ADC0_TS_CHARGE_STEPCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R/W	0h	
25	DIFF_CNTRL	R/W	0h	Differential Control Pin. 0h (R/W) = Single Ended 1h (R/W) = Differential Pair Enable
24-23	SEL_RFM_SWC	R/W	0h	SEL_RFM pins SW configuration. 0h (R/W) = VSSA_ADC 1h (R/W) = XNUR 2h (R/W) = YNLR 3h (R/W) = VREFN
22-19	SEL_INP_SWC	R/W	0h	SEL_INP pins SW configuration. Note all values 1xxx (binary) = VREFN. 0h (R/W) = Channel 1 1h (R/W) = Channel 2 2h (R/W) = Channel 3 3h (R/W) = Channel 4 4h (R/W) = Channel 5 5h (R/W) = Channel 6 6h (R/W) = Channel 7 7h (R/W) = Channel 8 8h (R/W) = VREFN

Table 11-21. ADC0_TS_CHARGE_STEPCONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18-15	SEL_INM_SWM	R/W	0h	SEL_INM pins for neg differential. Note all values 1xxx (binary) = VREFN. 0h (R/W) = Channel 1 1h (R/W) = Channel 2 2h (R/W) = Channel 3 3h (R/W) = Channel 4 4h (R/W) = Channel 5 5h (R/W) = Channel 6 6h (R/W) = Channel 7 7h (R/W) = Channel 8 8h (R/W) = VREFN
14-12	SEL_RFP_SWC	R/W	0h	SEL_RFP pins SW configuration. Note all values 1xx (binary) = INTREF. 0h (R/W) = VDDA_ADC 1h (R/W) = XPUL 2h (R/W) = YPLL 3h (R/W) = VREFP 4h (R/W) = INTREF
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW__SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW__SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-0	RESERVED	R/W	0h	

11.5.1.18 ADC0_TS_CHARGE_DELAY Register (offset = 60h) [reset = 1h]

Register mask: FFFFFFFFh

ADC0_TS_CHARGE_DELAY is shown in Figure 11-22 and described in Table 11-22.

TS Charge Delay Register

Figure 11-22. ADC0_TS_CHARGE_DELAY Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">RESERVED</td><td colspan="18">OPENDELAY</td></tr>
<tr><td colspan="14">R/W-0h</td><td colspan="18">R/W-1h</td></tr>
</table>

Table 11-22. ADC0_TS_CHARGE_DELAY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17-0	OPENDELAY	R/W	1h	Program the # of ADC clock cycles to wait between applying the step configuration registers and going back to the IDLE state. (Value must be greater than 0.)

11.5.1.19 ADC0_STEPCONFIG_0 Register (offset = 64h + [i * 8h]) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_STEPCONFIG_0 is shown in Figure 11-23 and described in Table 11-23.

Step Configuration n

Figure 11-23. ADC0_STEPCONFIG_0 Register

31	30	29	28	27	26	25	24
RESERVED				RANGE_CHECK	FIFO_SELECT	DIFF_CNTRL	SEL_RFM_SWC
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
SEL_RFM_SWC	SEL_INP_SWC				SEL_INM_SWC		
R/W-0h	R/W-0h				R/W-0h		
15	14	13	12	11	10	9	8
SEL_INM_SWC	SEL_RFP_SWC			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
YPPSW_SWC	XNNSW_SWC	XPPSW_SWC	AVERAGING			MODE	
R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	

Table 11-23. ADC0_STEPCONFIG_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27	RANGE_CHECK	R/W	0h	0h (R/W) = Disable out-of-range check 1h (R/W) = Compare ADC data with range check register
26	FIFO_SELECT	R/W	0h	Sampled data will be stored in FIFO. 0h (R/W) = FIFO 0 1h (R/W) = FIFO 1
25	DIFF_CNTRL	R/W	0h	Differential Control Pin
24-23	SEL_RFM_SWC	R/W	0h	SEL_RFM pins SW configuration. 0h (R/W) = VSSA_ADC 1h (R/W) = XNUR 2h (R/W) = YNLR 3h (R/W) = VREFN
22-19	SEL_INP_SWC	R/W	0h	SEL_INP pins SW configuration. Note all values 1xxx (binary) = VREFN. 0h (R/W) = Channel 1 1h (R/W) = Channel 2 2h (R/W) = Channel 3 3h (R/W) = Channel 4 4h (R/W) = Channel 5 5h (R/W) = Channel 6 6h (R/W) = Channel 7 7h (R/W) = Channel 8 8h (R/W) = VREFN

Table 11-23. ADC0_STEPCONFIG_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18-15	SEL_INM_SWC	R/W	0h	SEL_INM pins for neg differential. Note all values 1xxx (binary) = VREFN. 0h (R/W) = Channel 1 1h (R/W) = Channel 2 2h (R/W) = Channel 3 3h (R/W) = Channel 4 4h (R/W) = Channel 5 5h (R/W) = Channel 6 6h (R/W) = Channel 7 7h (R/W) = Channel 8 8h (R/W) = VREFN
14-12	SEL_RFP_SWC	R/W	0h	SEL_RFP pins SW configuration. Note all values 1xx (binary) = INTREF. 0h (R/W) = VDDA_ADC 1h (R/W) = XPUL 2h (R/W) = YPLL 3h (R/W) = VREFP 4h (R/W) = INTREF
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW_SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW_SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-2	AVERAGING	R/W	0h	Number of samplings to average: 0h (R/W) = No average 1h (R/W) = 2 samples average 2h (R/W) = 4 samples average 3h (R/W) = 8 samples average 4h (R/W) = 16 samples average
1-0	MODE	R/W	0h	0h (R/W) = SW enabled, one-shot 1h (R/W) = SW enabled, continuous 2h (R/W) = HW synchronized, one-shot 3h (R/W) = HW synchronized, continuous

11.5.1.20 ADC0_STEPDELAY_0 Register (offset = 68h + [i * 8h]) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_STEPDELAY_0 is shown in Figure 11-24 and described in Table 11-24.

Step Delay Register n

Figure 11-24. ADC0_STEPDELAY_0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">SAMPLEDELAY</td><td colspan="6">RESERVED</td><td colspan="2">OPENDELAY</td></tr>
<tr><td colspan="8">R/W-0h</td><td colspan="6">R/W-0h</td><td colspan="2">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">OPENDELAY</td></tr>
<tr><td colspan="16">R/W-0h</td></tr>
</table>

Table 11-24. ADC0_STEPDELAY_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SAMPLEDELAY	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SOC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	RESERVED	R/W	0h	
17-0	OPENDELAY	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion

11.5.1.21 ADC0_FIFOCOUNT_0 Register (offset = E4h + [i * Ch]) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_FIFOCOUNT_0 is shown in Figure 11-25 and described in Table 11-25.

FIFO[n] word count

Figure 11-25. ADC0_FIFOCOUNT_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
Rreturns0s-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									WORDS_IN_FIFO						
Rreturns0s-0h									R-0h						

Table 11-25. ADC0_FIFOCOUNT_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	Rreturns0s	0h	RESERVED
6-0	WORDS_IN_FIFO	R	0h	Number of words currently in the FIFO[n]

11.5.1.22 ADC0_FIFOTHR_0 Register (offset = E8h + [i * Ch]) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_FIFOTHR_0 is shown in Figure 11-26 and described in Table 11-26.

FIFO[n] Threshold level trigger

Figure 11-26. ADC0_FIFOTHR_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
Rreturns0s-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										FIFO_THR_LEVEL					
Rreturns0s-0h										R/W-0h					

Table 11-26. ADC0_FIFOTHR_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	Rreturns0s	0h	
5-0	FIFO_THR_LEVEL	R/W	0h	Program the desired FIFO[n] data sample level to reach before generating interrupt to CPU (program to value minus 1)

11.5.1.23 ADC0_DMAREQ_0 Register (offset = ECh + [i * Ch]) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_DMAREQ_0 is shown in Figure 11-27 and described in Table 11-27.

FIFO[n] DMA req[n] (request) trigger

Figure 11-27. ADC0_DMAREQ_0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">Rreturns0s-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="10">RESERVED</td><td colspan="6">DMA_REQUEST_LEVEL</td></tr>
<tr><td colspan="10">Rreturns0s-0h</td><td colspan="6">R/W-0h</td></tr>
</table>

Table 11-27. ADC0_DMAREQ_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	Rreturns0s	0h	RESERVED
5-0	DMA_REQUEST_LEVEL	R/W	0h	Number of words in FIFO[n] before generating a DMA request (program to value minus 1)

11.5.1.24 ADC0_FIFO0DATA Register (offset = 100h) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_FIFO0DATA is shown in Figure 11-28 and described in Table 11-28.

ADC_ FIFO0 _READ Data

Figure 11-28. ADC0_FIFO0DATA Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="12">RESERVED</td><td colspan="4">ADCCHNLID</td></tr>
<tr><td colspan="12">R-0h</td><td colspan="4">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="12">ADCDATA</td></tr>
<tr><td colspan="4">Rreturns0s-0h</td><td colspan="12">R-0h</td></tr>
</table>

Table 11-28. ADC0_FIFO0DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	RESERVED.
19-16	ADCCHNLID	R	0h	Optional ID tag of channel that captured the data. If tag option is disabled, these bits will be 0.
15-12	RESERVED	Rreturns0s	0h	
11-0	ADCDATA	R	0h	12 bit sampled ADC converted data value stored in FIFO 0.

11.5.1.25 ADC0_FIFO1DATA Register (offset = 200h) [reset = 0h]

Register mask: FFFFFFFFh

ADC0_FIFO1DATA is shown in Figure 11-29 and described in Table 11-29.

ADC FIFO1_READ Data

Figure 11-29. ADC0_FIFO1DATA Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="12">RESERVED</td><td colspan="4">ADCCHNLID</td></tr>
<tr><td colspan="12">R-0h</td><td colspan="4">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="12">ADCDATA</td></tr>
<tr><td colspan="4">R-0h</td><td colspan="12">R-0h</td></tr>
</table>

Table 11-29. ADC0_FIFO1DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	RESERVED
19-16	ADCCHNLID	R	0h	Optional ID tag of channel that captured the data. If tag option is disabled, these bits will be 0.
15-12	RESERVED	R	0h	RESERVED
11-0	ADCDATA	R	0h	12 bit sampled ADC converted data value stored in FIFO 1.

ADC1: Magnetic Card Reader

This chapter describes the Magnetic Card Reader (ADC1) of the device.

Topic **Page**

12.1 Introduction

The magnetic card reader (MCR) and analog-to-digital converter (ADC) subsystem (MCR_ADC_SS), also known as ADC1, implements several enhancements to the basic general-purpose ADC which allows it to be connected directly to magnetic card heads and operated as a magnet card reader.

ADC1 contains two major components: a digital finite-state machine sequencer (FSM sequencer) and analog front-end (AFE).

The FSM sequencer is set up and controlled by software using a 32-bit OCP interface and supports up to 16 programmable logical steps (a step defines how and when to sample a physical input). The ADC conversion data is stored in either of two FIFOs and can be read by the processor or DMA. Interrupts can be generated when the FIFO data has reached a programmable level. The FSM sequencer supports continuous sampling as well as single-shot modes.

The AFE consists of a single 12-bit successive approximation ADC, 4 programmable gain differential preamplifiers, 4 analog multiplexers, and 8 analog inputs. Analog multiplexers allow the 8 analog inputs to be configured as 4 differential inputs or 8 single-ended inputs when the preamplifiers are bypassed.

NOTE: The magnetic card reader function is not supported on the AM437x product family. ADC1 should only be used as a general-purpose ADC when using an AM437x device.

12.1.1 MagneticCard Reader Features

The main features of the MagneticCard Reader include:

- Programmable FSM sequencer that supports 16 steps:
  - Software register bit for start of sequence
  - Optional start of sequence with hardware events
    - External hardware event connected to the EXT_HW_TRIGGER input
    - ADC data value exceeds a programmable swipe threshold
  - Option to discard ADC data until it exceeds a programmable swipe threshold
    - Supported in software and hardware event modes
  - Continuous or one-shot single sequence
  - Sequence through up to 16 individually enabled steps
  - Programmable analog input selector for each step
  - Programmable Open Delay between steps
  - Programmable Sampling Delay (acquisition period) for each step
  - Programmable averaging of input samples - 16/8/4/2/1
  - Differential or singled ended mode setting for each step
  - Store data in either of two FIFOs
  - Option to encode physical step number with ADC data
  - FIFOs serviced by the processor or DMA
  - Programmable DMA level interrupt (for each FIFO)
  - Programmable FIFO level interrupt (for each FIFO)
  - Stop bit to end conversion
- Support for following interrupts/status, with masking:
  - Interrupt when ADC data is greater than or equal to a programmable swipe threshold
  - Interrupt after a sequence of enabled steps has completed (end_of_sequence)
  - Interrupt when FIFO fills to a programmable threshold
  - Interrupt if sampled data is out of a programmable range
  - Interrupt for FIFO overflow and underflow conditions
  - Status bit to indicate if ADC is busy converting

  - Status bits to indicate which step is currently being executed

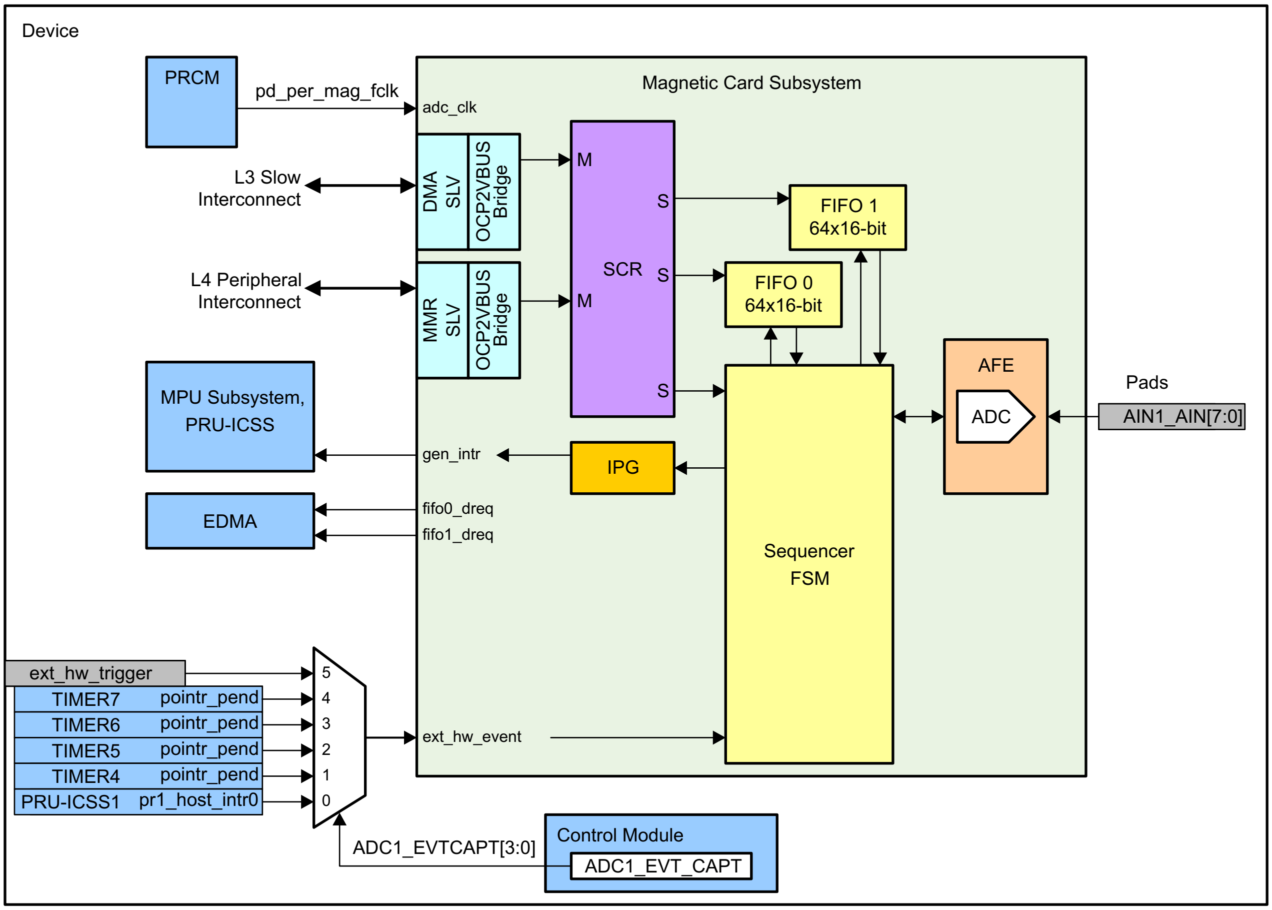
12.1.2 Unsupported Features

The magnetic card reader function is not supported on the AM437x product family. ADC1 should only be used as a general-purpose ADC when using an AM437x device.

12.2 Integration

Figure 12-1 shows how the MCR_ADC (ADC1) module is integrated into the device.

Figure 12-1. MagneticCard Reader Integration



12.2.1 MagneticCard Reader Connectivity Attributes

The general connectivity attributes for the MagneticCard Reader are shown in Table 12-1.

Table 12-1. MagneticCard Reader Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L3S_GCLK (OCP) PD_PER_MAG_FCLK (Functional)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Idle
Interrupt Requests	1 interrupt to MPU Subsystem, PRU-ICSS
DMA Requests	1 event Edge event signals to EDMA
Physical Address	L3 slow slave port (DMA) L4 peripheral slave port (MMR)

12.2.2 MagneticCard Reader Clock and Reset Management

The MagneticCard Reader has two clock domains. The OCP2VBUS bridges, SCR, IPG, FIFOs, and portions of the FSM Sequencer use the ocp_clk. The ADC AFE and portions of the FSM Sequencer use the adc_clk.

Table 12-2. MagneticCard Reader Clock Signals

Clock Signal	Maximum Frequency	Reference Source	Comments
ocp_clock Interface / functional clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l3s_gclk from PRCM
adc_clk ADC clock	19.2, 24, 25, or 26 MHz	CLK_M_OSC	pd_per_mgc_fgclk from PRCM
	192 MHz	PER_CLKOUTM2	

The frequency of ocp_clock must be greater than or equal to six times the frequency of adc_clk.

12.2.3 MagneticCard Reader Pin List

The MagneticCard Reader interface pins are summarized in Table 12-3.

Table 12-3. MagneticCard Reader Pin List

Pin	Type	Description
ADC1_AIN[7:0](1)	I	Analog inputs
ADC1_VREFN(2)	Analog	Negative External Reference Voltage Input
ADC1_VREFP(3)	Analog	Positive External Reference Voltage Input

(1) Unused analog inputs should be open-circuit.
(2) Connect ADC1_VREFN to VSSA_ADC when not using a negative external reference voltage.
(3) Connect ADC1_VREFP to VDDA_ADC1 when not using a positive external reference voltage.

If ADC1 is not used, connect all ADC1 terminals (ADC1_AIN[7:0], ADC1_VREFN, ADC1_VREFP, VDDA_ADC1, and VSSA_ADC) to the same ground as all VSS terminals.

12.3 Functional Description

12.3.1 FSM Sequencer Functional Description

The FSM sequencer supports up to 17 steps. One of the 17 steps is the Idle step which is always enabled and does not perform any analog-to-digital conversions. The other 16 steps always perform analog-to-digital conversions but they can be independently enabled or disabled.

The 16 steps which perform analog-to-digital conversions can individually be enabled and configured to operate as software enabled steps or hardware enabled steps. Each of these steps has their own programmable step configuration and step delay registers that allow users to configure the ADC operation on a per-step basis.

The FSM sequencer remains in the Idle step when it is not executing one of the 16 programmable steps. The Idle step does not have an enable bit in the step enable register or step delay register since this step does not perform any ADC operation.

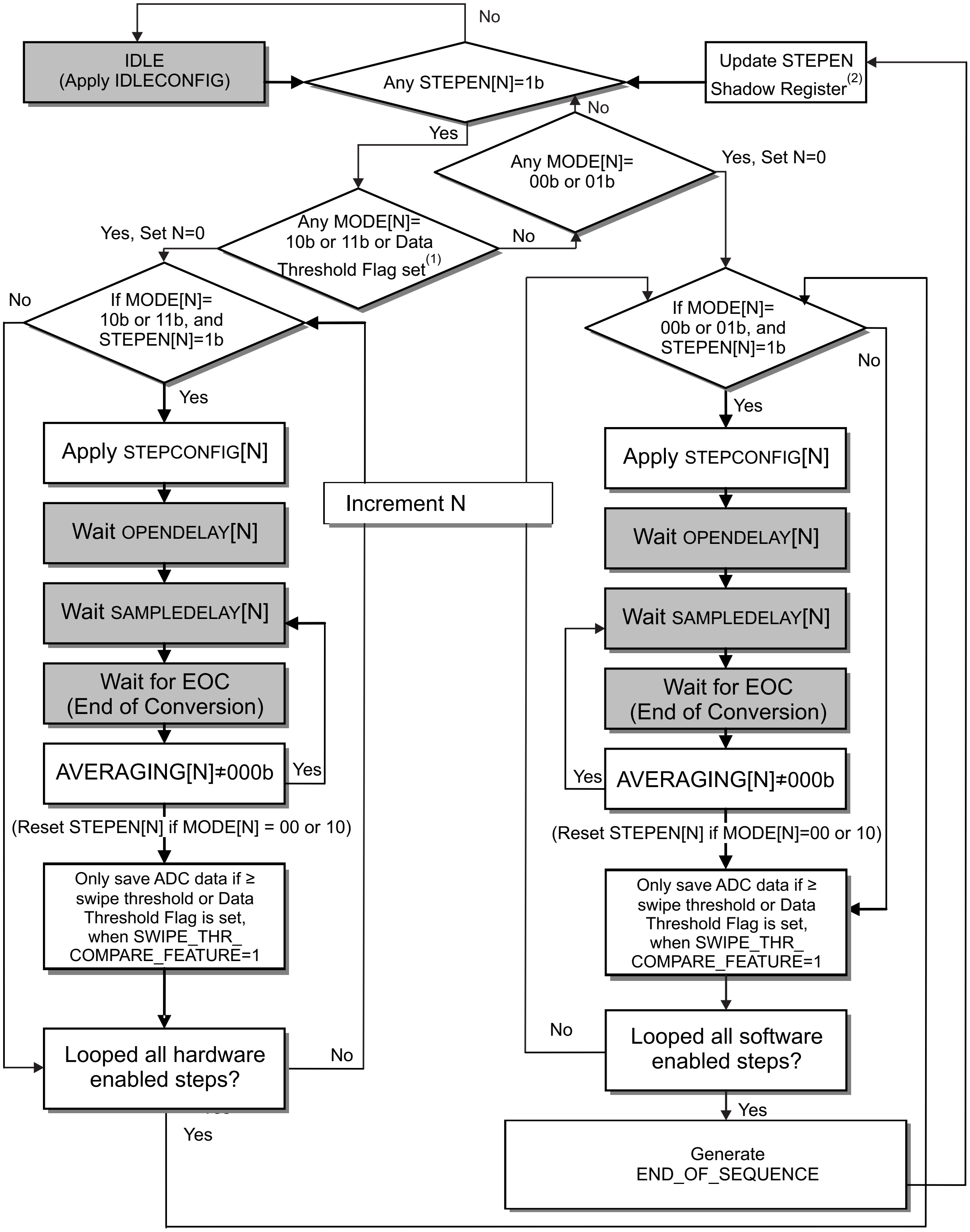
The FSM sequencer enters the Idle step and configures the AFE as defined by the idle configuration register immediately after ADC1 is enabled. It remains in the Idle step waiting for any of the 16 programmable steps to be enabled. When one or more step is enabled, the FSM sequencer will begin executing or skipping each step starting with the lowest and incrementing up through each step until the highest step has been executed or skipped. The FSM sequencer will execute enabled steps and skip disabled steps. Once all of the steps have been executed or skipped, the FSM sequencer will return to the Idle step or immediately begin repeating steps configured for continuous mode.

If the swipe threshold compare feature is enabled, the ADC output data will be compared to one of four swipe threshold registers. If the ADC output data value is lower than the swipe threshold, the ADC data is discarded (not saved in the FIFO), and the FSM sequencer continues to the next step. Once any step has reached or exceeded its swipe threshold, the FSM sequencer will save the ADC output data from the current step and all future steps without comparing ADC output data to the swipe threshold.

An end_of_sequence interrupt can be generated after the last enabled step has been executed. If any software enabled steps are configured for continuous operation and they are still enabled when the FSM sequencer has incremented through all the steps, the FSM sequencer will start the sequence again with the first enabled step. Hardware enabled steps are mapped to a hardware event and will need the hardware event to occur before being scheduled.

Figure 12-2 illustrates how the FSM sequencer works. Each shaded box represents a FSM state. **Note**: Figure 12-2 does not represent the number of adc_clk cycles required by each state since some states implement user programmable delays that may require more than one adc_clk cycle to complete.

Figure 12-2. FSM Sequencer



---

(1) A hardware event can be triggered by an external event or start of swipe interrupt, but the Data Threshold Flag is only set with the start of swipe interrupt and is only cleared when ADC1 is disabled.

(2) The STEPEN shadow register is updated after the FSM sequencer schedules the last step in order to queue up the next step.

12.3.1.1 Step Enable

A step is enabled or disabled via the STEPEN register.

The system software should only write to the STEPEN register when ADC1 is disabled or the FSM sequencer in in the Idle step. Therefore, it may be necessary to disable ADC1 before updating the STEPEN register if the FSM sequencer is executing back-to-back steps where it never enters the Idle step.

12.3.1.2 Step Configuration

12.3.1.2.1 One-Shot (Single) or Continuous Mode

Each step can be configured, via the least significant bit of the MODE bit-field in the respective STEPCONFIGx register, to operate in one-shot or continuous mode. Steps configured for one-shot mode will perform a single conversion before being disabled by the FSM sequencer. Steps configured for continuous mode will automatically be re-enabled by the FSM sequencer after each conversion until disabled by software.

12.3.1.2.2 Software- or Hardware-Enabled Steps

The user can configure each step to begin immediately after the step is enabled by software (SW Enabled), or wait for a hardware event to occur (HW Enabled). One of two sources can be selected for the hardware event.

One option is an internal comparator that compares the value of ADC output data to the value programmed into one of four swipe threshold registers. The hardware event will be triggered if the ADC output data value is greater than or equal to the selected swipe threshold when this option is selected. When using this option, an internal Data Threshold Flag will be set once the hardware event has occurred and this flag tells the FSM sequencer to continuously repeat all HW Enabled steps until ADC1 is disabled.

The other option is the EXT_HW_TRIGGER input signal which can be sourced from a device input terminal. When this option is selected, the hardware event will be triggered after the EXT_HW_TRIGGER input signal transitions from low to high and is held high for a period greater than two cycles of ocp_clock. This hardware event will only schedule one complete sequence, even for continuous mode. If HW Enabled steps need to be executed again, a new hardware event must be generated after the previously triggered sequence has completed.

The EXT_HW_TRIGGER input cannot be used as a hardware event when the FSM sequencer is configured to use the start of swipe interrupt as a hardware event.

Each step can be configured to operate in SW Enabled mode or HW Enabled mode independently via the most significant bit of the MODE bit-field in the respective STEPCONFIGx register. However, the hardware event source is applied globally to all HW Enabled steps and configured via the CTRL register.

12.3.1.2.3 Averaging of Samples

Each step can be configured, via the respective STEPCONFIGx register, to sample an input 1, 2, 4, 8, or 16 times and provide an average data value. If a step is configured to sample more than once, the additional samples are taken back-to-back immediately after the first sample. However, Open Delay is only applied to the first sample while Sample Delay is applied to all samples. Once the last sample has been taken the average data value will be stored in the FIFO unless the swipe threshold compare feature is enabled and the average data value is less than the swipe threshold.

12.3.1.2.4 Preamplifier Gain Control

Four differential preamplifiers with a programmable voltage gain of 12, 14, 16, or 18 were added to ADC1 as one of several magnet card reader enhancements. Each step provides individual gain control for each preamplifier via the respective STEPCONFIGx register.

12.3.1.2.5 Swipe Threshold Compare Feature

The swipe threshold compare feature is another magnet card reader enhancement. When this feature is enabled, via the respective STEPCONFIGx register, the FSM sequencer will automatically discard ADC data until it has exceeded a programmable swipe threshold value.

An internal Data Threshold Flag will be set once this hardware event has occurred and this flag tells the FSM sequencer to continuously repeat all HW Enabled steps until ADC1 is disabled.

12.3.1.2.6 Analog Multiplexer Input Select

The AFE contains 4 analog multiplexers. Two 9-to-1 multiplexers are used to select the source connected to the positive input (INP) and negative input (INM) of the ADC. A 6-to-1 multiplexer is used to select the source connected to the positive reference input (ADC_REFP) of the ADC. A 4-to-1 multiplexer is used to select the source connected to the negative reference input (ADC_REFN) of the ADC.

Each step can be configured, via the respective STEPCONFIGx register, to select the appropriate ADC input sources. For more information, refer to the ADC1 AFE Functional Block Diagram (Figure 12-5) and the STEPCONFIG register field descriptions.

12.3.1.2.7 Differential Control

ADC1 supports differential inputs and each step can be configured, via the respective STEPCONFIGx register, to select single-ended input mode or differential input mode.

Each enabled step should be configured for Differential mode when the AFE preamplifiers are not bypassed. Refer to the CRTL register field descriptions for information related to bypassing the AFE preamplifiers.

12.3.1.2.8 FIFO Select

ADC1 contains two FIFOs and each step can be configured, via the respective STEPCONFIGx register, to store its ADC data into either FIFO.

12.3.1.2.9 Range Check Interrupt Enable

Each step can be configured, via the respective STEPCONFIGx register, to compare the value of ADC data to high and low limits programmed in the ADCRANGE register. The out-of-range interrupt is generated if the value of ADC data is greater than the value of THR_HIGH_RANGE_DATA or less than the value of THR_LOW_RANGE_DATA. Refer to the ADCRANGE register field descriptions for information related to setting the high and low limits that trigger the out-of-range interrupt.

12.3.1.2.10 Swipe Threshold Register Pointer

ADC1 provides four swipe threshold registers and each step can be configured, via the respective STEPCONFIGx register, to use one of these four swipe thresholds when the swipe threshold compare feature is enabled.

12.3.1.3 Open Delay and Sample Delay

The FSM sequencer provides two programmable delays for each step. Open Delay is used to control when the acquisition begins after the step starts and Sample Delay is used to control the acquisition period. Delays for each of the 16 steps can be configured independently via the respective STEPDELAYx register.

12.3.1.3.1 Open Delay

Open Delay defaults to a value of zero which causes the acquisition period to begin as soon as the step starts. The start of the acquisition period can be delayed one adc_clk clock period for each incremental value of Open Delay.

12.3.1.3.2 Sample Delay

Sample Delay defaults to a value of zero which causes the acquisition period to be equal to two adc_clk clock periods. The acquisition period can be extended one adc_clk clock for each incremental value of Sample Delay.

When the AFE preamplifiers are not bypassed, the value of Sample Delay must be configured to provide a minimum acquisition period of 0.6 us which provides enough time for the preamplifiers outputs to settle. When the AFE preamplifiers are bypassed, the value of Sample Delay should be configured to provide enough time for the respective external voltage source to completely charge the AFE input capacitance during the acquisition period.

12.3.1.4 Interrupts

ADC1 has 9 internal events that can trigger an interrupt to the processor. These events are logically or’ed together to produce a single interrupt to the processor. The user can individually enable or disable each interrupt source via the IRQEN_SET and IRQEN_CLR registers. Once an interrupt is generated, the IRQSTS register can be read to determine which interrupt source(s) interrupted the processor. The interrupt source(s) can be cleared by writing a ‘1’ to the corresponding bit of the IRQSTS register.

12.3.1.4.1 End of Sequence

An END_OF_SEQUENCE interrupt is generated after the FSM sequencer completes the last enabled step.

12.3.1.4.2 FIFO Threshold

Each FIFO has the ability to generate an interrupt when the FIFO word count has reached a programmable threshold value. The FIFO0_THR/FIFO1_THR interrupts are generated when the value of FIFO0COUNT/FIFO1COUNT registers equals the word count threshold programmed in the respective FIFO0THR/FIFO1THR registers.

12.3.1.4.3 FIFO Overrun and Underflow

Each FIFO also has the ability to generate overrun and underflow interrupts. To clear a FIFO underflow or FIFO overrun interrupt, the user should write a ‘1’ so the status bit. ADC1 does not recover from these conditions automatically. Therefore, software will need to disable and re-enable ADC1 after this occurs. Before user can turn the module back on, they must first check if the FSM sequencer is in the Idle step by reading the ADC_STATUS_REG register.

12.3.1.4.4 Out of Range

The OUT_OF_RANGE interrupt can be enabled or disabled on each step. If enabled, the ADC data value is compared to the low and high thresholds programmed in the global ADCRANGE register. The OUT_OF_RANGE interrupt is generated if the ADC data value is greater than the value programmed in THR_HIGH_RANGE_DATA or less than the value programmed in THR_LOW_RANGE_DATA.

12.3.1.4.5 Start of Swipe

The START_OF_SWIPE interrupt can be enabled or disabled on each step. If enabled, the ADC data value is compared to one of four swipe threshold registers. The START_OF_SWIPE interrupt is generated if the ADC data value is greater than or equal to the value programmed in the swipe threshold register being pointed to by the respective SWIPE_THR_REG_POINTER. This interrupt can be configured as a hardware event which can be used to trigger HW enabled steps.

12.3.1.5 DMA Requests

Each FIFO can be serviced by the processor or DMA. The user can individually enable or disable each FIFO DMA request via the DMAEN_SET and DMAEN_CLR registers.

A DMA request is generated when the value of FIFO0COUNT/FIFO1COUNT registers equals the word count threshold programmed in the respective DMA0REQ/DMA1REQ registers.

The DMA slave port allows for burst reads in order to effectively move the FIFO data. Refer to the DMA section for more information.

12.3.2 AFE Functional Description

The AFE contains a single 12-bit successive approximation ADC which can be connected to one of eight user selectable analog inputs for each step executed by the FSM sequencer. There is also an option to configure the ADC to operate in differential mode where it samples the differential voltage applied to two of eight user selectable analog inputs. When operating in differential mode, the user also has the option of routing the differential input signal through a programmable gain preamplifier.

The ADC sits idle until the FSM sequencer sends a Start of Conversion (SOC) pulse. Once the FSM sequencer sends a SOC pulse to the ADC, it begins sampling the analog input signal on the rising edge of SOC and continues sampling the analog input signal one cycle of adc_clk after the falling edge of SOC. The SOC pulse width will be the value of Sample Delay plus one adc_clk periods, so the total ADC acquisition time will be the value of Sample Delay plus two adc_clk periods. The ADC captures the analog input signal at the end of the acquisition period and starts conversion, which requires thirteen adc_clk periods to digitize the sampled input. An EOC signal is returned to the FSM sequencer to indicate the digital data is ready.

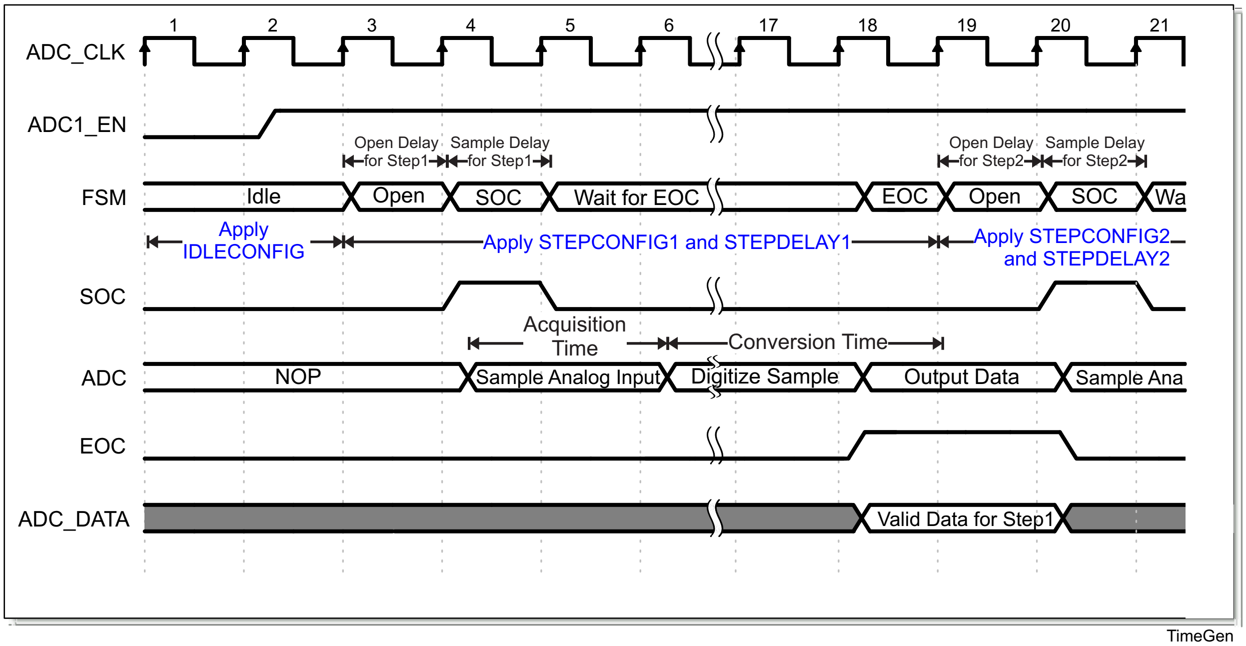
The ADC output is positive binary weighted data ranging from 0 to (2^{12} - 1). When configured for single-ended mode, ADC output data values 0 to (2^{12} - 1) represents a corresponding input voltage that ranges from the ADC negative reference (REFM) voltage to the ADC positive reference (REFP) voltage. When configured for differential mode, ADC output data values 0 to ((2^{12}/2) - 1) represents a corresponding negative differential input voltage that ranges from the ADC REFP voltage to the ADC REFM voltage and output data values (2^{12}/2) to (2^{12} - 1) represents a corresponding positive differential input voltage that ranges from the ADC REFM voltage to the ADC REFP voltage.

Using the minimum values of Open Delay and Sample Delay, the ADC can sample an analog input every 15 adc_clk periods.

Figure 12-3 illustrates the operation of the FSM sequencer and ADC, with indicators showing when the hardware configuration defined by STEPCONFIGx and STEPDELAYx registers are applied.

Figure 12-3 assumes both steps shown are software enabled with averaging turned off, an OPENDELAY value of 1, and SAMPLEDELAY value of 0. The analog input is sampled for the duration of the SOC pulse plus one cycle of the ADC_CLK. If the value of OPENDELAY were 0, it would appear as if the time associated with the Open portion of the FSM waveform would be removed from all the waveforms.

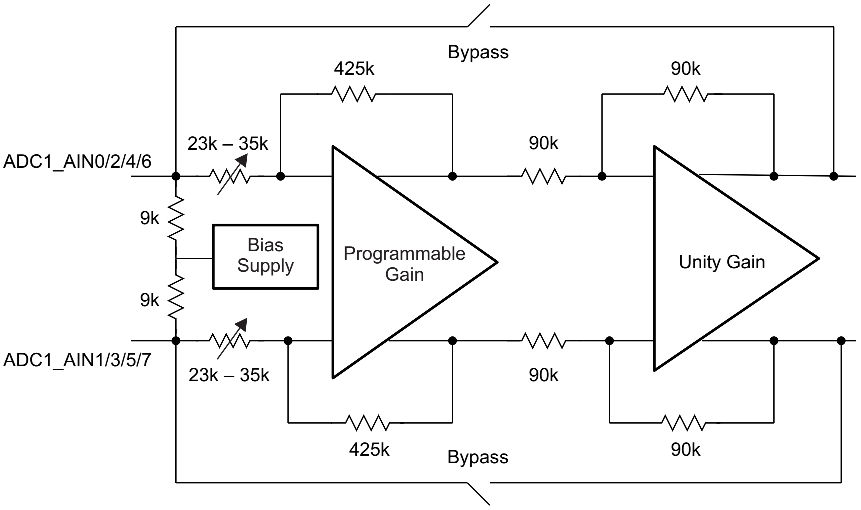
Figure 12-3. Example Timing Diagram for FSM Sequencer



12.3.2.1 AFE Functional Block Diagram

The AFE has 8 analog inputs which can be configured as either singled-ended or differential inputs to the ADC. Each analog input has an internal bias resistor connected to a mid-supply voltage source of VDDA_ADC1 divided by 2 that is shared with the other input associated with the preamplifier. A schematic of the input bias resistors, bias supply, and preamplifier is shown in Figure 12-4.

Figure 12-4. Input Bias Resistors, Bias Supply, and Preamplifier Schematic



NOTE: The resistor values shown are typical values.

The four preamplifiers have individual programmable voltage gain of 12, 14, 16, or 18.

If the preamplifiers are not bypassed, the ADC must be configured to operate in differential mode with one the following four input topologies. ADC1_AIN0 paired with ADC1_AIN1 where ADC1_AIN0 is the positive input and ADC1_AIN1 is the negative input to Preamplifier1. ADC1_AIN2 paired with ADC1_AIN3 where ADC1_AIN2 is the positive input and ADC1_AIN3 is the negative input to Preamplifier2. ADC1_AIN4 paired with ADC1_AIN5 where ADC1_AIN4 is the positive input and ADC1_AIN5 is the negative input to Preamplifier3. ADC1_AIN6 paired with ADC1_AIN7 where ADC1_AIN6 is the positive input and ADC1_AIN7 is the negative input to Preamplifier4. The differential input impedance of each preamplifier is 18 kohms biased to VDDA_ADC1 divided by 2 with a 22k-50k ohm source.

If the preamplifiers are bypassed, the user may configure a step to use any 1 of the 8 inputs when operating in single-ended mode or any 2 of the 8 inputs when operating in differential mode. However, the effect of the bias resistors and bias supply on the attached voltage source(s) should be considered when using ADC1 as a general purpose ADC. Each internal bias supply has a high output impedance when turned off which will be the case when bypassing the preamplifiers. In this use case, the majority of the input leakage current from any of the analog inputs will be based on the external voltage source connected to the other input associated with the preamplifier. For example, the ADC would measure 0.975 volts on ADC1_AIN1 if it were connected to a 1.0 volt, 1 kohm source while ADC1_AIN2 is connected to a 0.5 volt, 1 kohm source. The error caused by the internal bias resistors can be minimized by connecting the analog inputs to low impedance voltage sources or leaving the other input associated with the preamplifier open-circuit.

Figure 12-5. AFE Functional Block Diagram

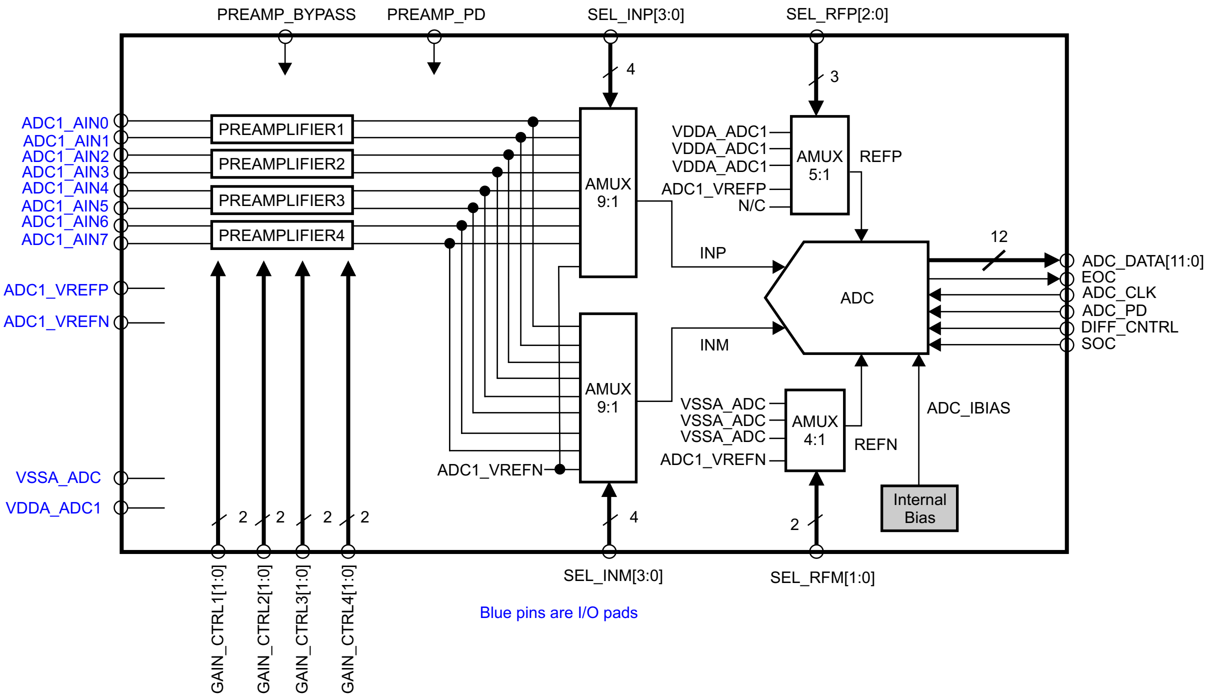


Table 12-4. ADC1 AFE Signal List (see Figure 12-5)

Signal	Type	Function
ADC_CLK	In	ADC clock input.
ADC_DATA[11:0]	Out	Digital data outputs.
ADC_PD	In	ADC power down input, active high.
ADC1_AIN0 (1)	In	Analog input 1.
ADC1_AIN1 (1)	In	Analog input 2.
ADC1_AIN2 (1)	In	Analog input 3.
ADC1_AIN3 (1)	In	Analog input 4.
ADC1_AIN4 (1)	In	Analog input 5.
ADC1_AIN5 (1)	In	Analog input 6.
ADC1_AIN6 (1)	In	Analog input 7.
ADC1_AIN7 (1)	In	Analog input 8.
ADC1_VREFN (1)	In	ADC negative reference input.
ADC1_VREFP (1)	In	ADC positive reference input.
DIFF_CNTRL	In	Differential control input, active high.
EOC	Out	End of Conversion output, active high.
GAIN_CTRL1 [1:0]	In	Gain control inputs for preamplifier 1.
GAIN_CTRL2 [1:0]	In	Gain control inputs for preamplifier 2.
GAIN_CTRL3 [1:0]	In	Gain control inputs for preamplifier 3.
GAIN_CTRL4 [1:0]	In	Gain control inputs for preamplifier 4.
PREAMP_PD	In	Preamplifier power down input, active high.
PREAMP_BYPASS	In	Preamplifier bypass input, active high.
SEL_RFP[2:0]	In	REFP multiplexer control inputs to select the ADC positive reference voltage.
SEL_RFM[1:0]	In	REFM multiplexer control inputs to select the ADC negative reference voltage.
SEL_INP[3:0]	In	INP multiplexer control inputs to select the ADC positive analog input.
SEL_INM[3:0]	In	INM multiplexer control inputs to select the ADC negative analog input.
SOC	In	Start of conversion input, active high.
VDDA_ADC1 (1)	Power	ADC1 Analog Power
VSSA_ADC (1)	Power	ADC0 and ADC1 Analog Ground

(1) Signal connects to device terminals.

12.3.3 FIFOs and DMA

12.3.3.1 FIFOs

The processor can read ADC data directly from the FIFO by using the respective FIFO0DATA or FIFO1DATA register. The internal logic will pop the next data from the FIFO and increment the FIFO read pointers.

The FIFO data will no longer be accessible after ADC1 is disabled because the FIFO pointers are reset. Therefore, it is important to read all FIFO data before disabling ADC1.

12.3.3.2 DMA

ADC1 has a dedicated slave port for the DMA which allows it to perform continuous burst reads when accessing the FIFOs.

Each FIFO has its own DMA request which can be enabled via the DMAEN_SET register and disabled via the DMAEN_CLR register.

The first DMA request is generated after the FIFO leaves the EMPTY state and fills to the level programmed in the DMA_REQUEST_LEVEL bit field of the respective DMA0REQ or DMA1REQ register.

Subsequently, a new DMA request is automatically generated on the next clock cycle after the current DMA access completes if the previous DMA access did not empty the FIFO.

12.3.4 Power Management

ADC1 supports smart-idle mode where it shares slave idle/request handshaking with the PRCM that allows ADC1 to automatically enter an idle state when it is disabled or the FSM sequencer is in the Idle state with all the steps disabled.

Software also has the option to directly turn off power to the ADC by setting the ADC_POWER_DOWN bit in the CTRL register.

The preamplifiers in the AFE can be powered down by setting the PREAMP_PD bit in the CTRL register. The preamplifiers are automatically powered down when they are bypassed by setting the PREAMP_BYPASS bit in the CTRL register.

It is the software’s responsibility to empty the FIFO before allowing ADC1 to enter the idle state. There should be no activity on the DMA slave port or MMR slave ports while ADC1 is in the idle state.

12.3.5 Magnetic Card Operation (Use Cases)

The AFE implemented in ADC1 provides four differential inputs with integrated preamplifiers that can be connected directly to magnetic heads to implement a magnetic card reader.

12.3.5.1 Use Case 1

Continuously sample a single differential analog input while comparing the ADC output data to a programmable swipe threshold value to detect a card swipe, then read one or more tracks of card data.

The FSM sequencer should be configured to enable one software enabled continuous step that samples the appropriate differential input with its swipe compare feature enabled (bit 11 in STEPCONFIGx register). If more than one track is being read from the card after the swipe is detected, the FSM sequencer must also be configured with hardware enabled steps that sample the differential inputs associated with the other tracks.

This configuration will cause the FSM sequencer to continuously sample the input being sampled by the software enabled step and compare the ADC output data to the swipe threshold value that is programmed in one of four threshold registers. The appropriate threshold register is selected with the swipe threshold register pointer (bits 31:30 in STEPCONFIGx register). If the ADC output data corresponding to the analog input is less than swipe threshold, the ADC output data is discarded and the analog input is sampled again.

When the analog input produces an ADC output data value that is equal to or greater than the swipe threshold, the ADC output data will be stored in the FIFO and the start of swipe interrupt will be generated if enabled. The start of swipe interrupt can be configured to automatically enable the hardware enabled steps which sample the other differential inputs.

This example describes how a single track on the magnetic card can be used to detect a card swipe which causes all tracks of the magnetic card to be continuously sampled and resulting ADC output data stored in the FIFO until FSM sequencer is disabled (bit 0 in CTRL register).

The FSM sequencer will perform the following operations to implement this use case:

Sample input associated with software enabled step (if data < swipe threshold, discard data),

Sample input associated with software enabled step until data >= swipe threshold (save data, generate start of swipe interrupt),

Sample inputs associated with hardware enabled steps then sample input associated with software enabled step (save data in the order samples were taken),

Repeat previous sequence until ADC1 is disabled.

12.3.5.2 Use Case 2

Continuously sample all four differential analog inputs while comparing the ADC output data of each sample to its programmable swipe threshold value to detect a card swipe, then read four tracks of card data.

The FSM sequencer should be configured to enable four software enabled continuous steps such that each step samples one of the four differential inputs with each step having its swipe compare feature enabled (bit 11 in STEPCONFIGx register).

This configuration will cause the FSM sequencer to continuously sample all four differential inputs and compare the ADC output data from each step to its respective swipe threshold value that is programmed in one of four threshold registers. The appropriate threshold register is selected with the swipe threshold register pointer (bits 31:30 in STEPCONFIGx register). If the ADC output data corresponding to the analog input is less than swipe threshold, the ADC output data is discarded and the next analog input is sampled.

When any of the four analog inputs produces an ADC output data value that is equal to or greater than its respective swipe threshold, the ADC output data from that step will be stored in the FIFO and the start of swipe interrupt will be generated if enabled. The FSM sequencer will continue executing the same sequence of steps with the next step being the one that follows the step which triggered the swipe interrupt. The ADC output data from each step following the swipe interrupt will be stored in the FIFO.

This example describes how all four tracks on the magnetic card can be used to detect a card swipe which causes all tracks of the magnetic card to be continuously sampled and resulting ADC output data stored in the FIFO until FSM sequencer is disabled (bit 0 in CTRL register).

The FSM sequencer will perform the following operations to implement this use case:

Sample input associated with first software enabled step (if data < swipe threshold, discard data),

Sample input associated with second software enabled step (if data < swipe threshold, discard data),

Sample input associated with third software enabled step (if data < swipe threshold, discard data),

Sample input associated with forth software enabled step (if data < swipe threshold, discard data),

Repeat above sequence of steps until data >= swipe threshold (save data, generate start of swipe interrupt),

Resume the same sequence of steps (save data in the order samples were taken),

Continue until ADC1 is disabled.

12.3.6 Simultaneous Control of ADC0

ADC1 can be configured to control the AFE in ADC0. This feature is enabled by setting the SIMULTANEOUS_CTRL bit in the CTRL register. When enabled, the ADC1 FSM sequencer sources the AFE control signals to both ADC0 AFE and ADC1 AFE. This causes the ADC0 AFE to perform the same operations as ADC1 AFE, but ADC0 AFE will be sampling ADC0 analog inputs while ADC1 AFE is sampling ADC1 analog inputs.

The ADC output data from ADC0 AFE is returned to the ADC1 FSM sequencer which stores this data in FIFO1 while storing ADC output data from ADC1 AFE in FIFO0. The FSM sequencer packs the 12 bits of ADC data and optional 4-bit step ID from both AFEs into a single 32-bit value when the processor or DMA reads FIFO0 when simultaneous control mode is enabled.

The ADC output data from ADC0 AFE is returned to the ADC1 FSM sequencer which stores this data in FIFO1 while storing ADC output data from ADC1 AFE in FIFO0. The FSM sequencer packs the 12 bits of ADC data and optional 4-bit step ID from both AFEs into a single 32-bit value when the processor or DMA reads FIFO0 when simultaneous control mode is enabled.

Figure 12-6 and Table 12-5 describe the contents of the ADC1_FIFO0DATA read by the processor or DMA when ADC1 is operating in simultaneous control mode.

For a description of ADC1_FIFO0DATA when ADC1 is not operating in simultaneous control mode, see Section 12.4.1.55.

Figure 12-6. ADC1_FIFO0DATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC0CHNLID				ADC0DATA											
R-0h				R-0h											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC1CHNLID				ADC1DATA											
R-0h				R-0h											

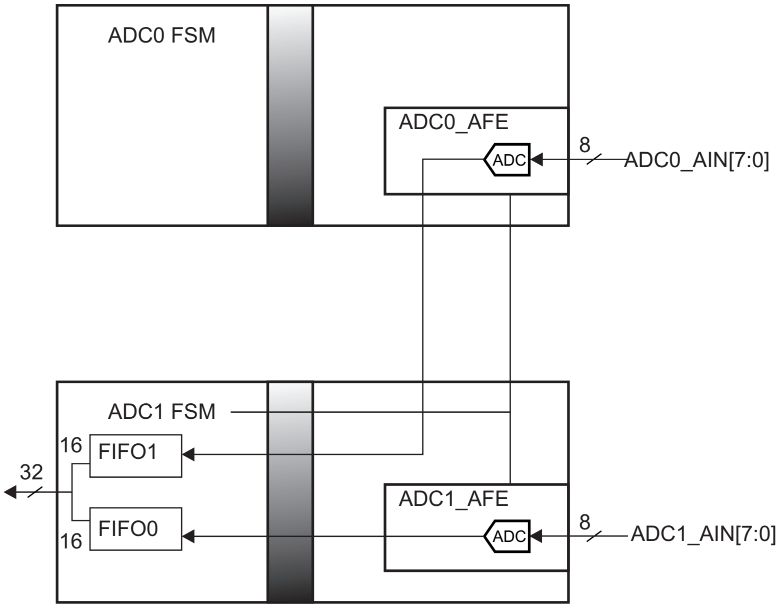
Table 12-5. ADC1_FIFO0DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	ADC0CHNLID	R	0h	ADC0 Step ID Tag of step that captured ADC0 data. These bits will be 0000b if STEP_ID_TAG = 0b.
27-16	ADC0DATA	R	0h	12-bit ADC0 data.
15-12	ADC1CHNLID	R	0h	ADC1 Step ID Tag of step that captured ADC1 data. These bits will be 0000b if STEP_ID_TAG = 0b.
11-0	ADC1DATA	R	0h	12-bit ADC1 data.

All DMA requests and FIFO interrupts are initiated by the ADC1 FSM sequencer based on the conditions of FIFO0 data only. All DMA requests and FIFO interrupts related to FIFO1 should be disabled

Below is a block diagram for the integration of ADC0 and ADC1 when operating in simultaneous mode.

Figure 12-7. Integration of ADC0 and ADC1 in Simultaneous Mode



12.4 Registers

ADC1 only supports 32-bit aligned read/write register accesses.

12.4.1 ADC1 Registers

Table 12-6 lists the memory-mapped registers for the ADC1. All register offset addresses not listed in Table 12-6 should be considered as reserved locations and the register contents should not be modified.

Generated Address Block

Table 12-6. ADC1 Registers

Offset	Acronym	Register Name	Section
0h	ADC1_REVISION	ADC1_REVISION	Section 12.4.1.1
10h	ADC1_SYSCONFIG	ADC1_SYSCONFIG	Section 12.4.1.2
24h	ADC1_IRQSTS_RAW	ADC1_IRQSTATUS_RAW	Section 12.4.1.3
28h	ADC1_IRQSTS	ADC1_IRQSTATUS	Section 12.4.1.4
2Ch	ADC1_IRQEN_SET	ADC1_IRQENABLE_SET	Section 12.4.1.5
30h	ADC1_IRQEN_CLR	ADC1_IRQENABLE_CLR	Section 12.4.1.6
38h	ADC1_DMAEN_SET	ADC1_DMAENABLE_SET	Section 12.4.1.7
3Ch	ADC1_DMAEN_CLR	ADC1_DMAENABLE_CLR	Section 12.4.1.8
40h	ADC1_CTRL		Section 12.4.1.9
44h	ADC1_ADCSTAT	ADC1_ADCSTAT	Section 12.4.1.10
48h	ADC1_ADCRANGE	ADC1_ADCRANGE	Section 12.4.1.11
4Ch	ADC1_CLKDIV	ADC1_ADC_CLKDIV	Section 12.4.1.12
54h	ADC1_STEPEN	ADC1_STEPENABLE	Section 12.4.1.13
58h	ADC1_IDLECONFIG	ADC1_IDLECONFIG	Section 12.4.1.14
5Ch	ADC1_SWIPE_COMPARE_REG1_2	ADC1_SWIPE_COMPARE_REG1_2	Section 12.4.1.15
60h	ADC1_SWIPE_COMPARE_REG3_4	ADC1_SWIPE_COMPARE_REG3_4	Section 12.4.1.16

Table 12-6. ADC1 Registers (continued)

Offset	Acronym	Register Name	Section
64h	ADC1_STEPCONFIG1	ADC1_STEPCONFIG1	Section 12.4.1.17
68h	ADC1_STEPDELAY1	ADC1_STEPDELAY1	Section 12.4.1.18
6Ch	ADC1_STEPCONFIG2	ADC1_STEPCONFIG2	Section 12.4.1.19
70h	ADC1_STEPDELAY2	ADC1_STEPDELAY2	Section 12.4.1.20
74h	ADC1_STEPCONFIG3	ADC1_STEPCONFIG3	Section 12.4.1.21
78h	ADC1_STEPDELAY3	ADC1_STEPDELAY3	Section 12.4.1.22
7Ch	ADC1_STEPCONFIG4	ADC1_STEPCONFIG4	Section 12.4.1.23
80h	ADC1_STEPDELAY4	ADC1_STEPDELAY4	Section 12.4.1.24
84h	ADC1_STEPCONFIG5	ADC1_STEPCONFIG5	Section 12.4.1.25
88h	ADC1_STEPDELAY5	ADC1_STEPDELAY5	Section 12.4.1.26
8Ch	ADC1_STEPCONFIG6	ADC1_STEPCONFIG6	Section 12.4.1.27
90h	ADC1_STEPDELAY6	ADC1_STEPDELAY6	Section 12.4.1.28
94h	ADC1_STEPCONFIG7	ADC1_STEPCONFIG7	Section 12.4.1.29
98h	ADC1_STEPDELAY7	ADC1_STEPDELAY7	Section 12.4.1.30
9Ch	ADC1_STEPCONFIG8	ADC1_STEPCONFIG8	Section 12.4.1.31
A0h	ADC1_STEPDELAY8	ADC1_STEPDELAY8	Section 12.4.1.32
A4h	ADC1_STEPCONFIG9	ADC1_STEPCONFIG9	Section 12.4.1.33
A8h	ADC1_STEPDELAY9	ADC1_STEPDELAY9	Section 12.4.1.34
ACh	ADC1_STEPCONFIG10	ADC1_STEPCONFIG10	Section 12.4.1.35
B0h	ADC1_STEPDELAY10	ADC1_STEPDELAY10	Section 12.4.1.36
B4h	ADC1_STEPCONFIG11	ADC1_STEPCONFIG11	Section 12.4.1.37
B8h	ADC1_STEPDELAY11	ADC1_STEPDELAY11	Section 12.4.1.38
BCh	ADC1_STEPCONFIG12	ADC1_STEPCONFIG12	Section 12.4.1.39
C0h	ADC1_STEPDELAY12	ADC1_STEPDELAY12	Section 12.4.1.40
C4h	ADC1_STEPCONFIG13	ADC1_STEPCONFIG13	Section 12.4.1.41
C8h	ADC1_STEPDELAY13	ADC1_STEPDELAY13	Section 12.4.1.42
CCh	ADC1_STEPCONFIG14	ADC1_STEPCONFIG14	Section 12.4.1.43
D0h	ADC1_STEPDELAY14	ADC1_STEPDELAY14	Section 12.4.1.44
D4h	ADC1_STEPCONFIG15	ADC1_STEPCONFIG15	Section 12.4.1.45
D8h	ADC1_STEPDELAY15	ADC1_STEPDELAY15	Section 12.4.1.46
DCh	ADC1_STEPCONFIG16	ADC1_STEPCONFIG16	Section 12.4.1.47
E0h	ADC1_STEPDELAY16	ADC1_STEPDELAY16	Section 12.4.1.48
E4h	ADC1_FIFO0COUNT	ADC1_FIFO0COUNT	Section 12.4.1.49
E8h	ADC1_FIFO0THR	ADC1_FIFO0THRESHOLD	Section 12.4.1.50
ECh	ADC1_DMA0REQ	ADC1_DMA0REQ	Section 12.4.1.51
F0h	ADC1_FIFO1COUNT	ADC1_FIFO1COUNT	Section 12.4.1.52
F4h	ADC1_FIFO1THR	ADC1_FIFO1THRESHOLD	Section 12.4.1.53
F8h	ADC1_DMA1REQ	ADC1_DMA1REQ	Section 12.4.1.54
100h	ADC1_FIFO0DATA	ADC1_FIFO0DATA	Section 12.4.1.55
200h	ADC1_FIFO1DATA	ADC1_FIFO1DATA	Section 12.4.1.56

12.4.1.1 ADC1_REVISION Register (offset = 0h) [reset = 47600001h]

ADC1_REVISION is shown in Figure 12-8 and described in Table 12-7.

Figure 12-8. ADC1_REVISION Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCHEME		RESERVED		FUNC											
R-1h		R-		R-760h											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R_RTL					X_MAJOR			CUSTOM		Y_MINOR					
R-0h					R-0h			R-0h		R-1h					

Table 12-7. ADC1_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	HL 0.8 scheme
29-28	RESERVED	R		
27-16	FUNC	R	760h	Functional Number
15-11	R_RTL	R	0h	RTL revision. Will vary depending on release.
10-8	X_MAJOR	R	0h	Major revision.
7-6	CUSTOM	R	0h	Custom revision.
5-0	Y_MINOR	R	1h	Minor revision

12.4.1.2 ADC1_SYSCONFIG Register (offset = 10h) [reset = 8h]

ADC1_SYSCONFIG is shown in Figure 12-9 and described in Table 12-8.

Figure 12-9. ADC1_SYSCONFIG Register

Table 12-8. ADC1_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
3-2	IDLEMODE	R/W	2h	00 -> Force Idle (always acknowledges). 01 -> No Idle Mode (never acknowledges). 10 -> Smart-Idle Mode. 11 -> Reserved.
1-0	RESERVED	R		

12.4.1.3 ADC1_IRQSTS_RAW Register (offset = 24h) [reset = 0h]

ADC1_IRQSTS_RAW is shown in Figure 12-10 and described in Table 12-9.

Figure 12-10. ADC1_IRQSTS_RAW Register

Table 12-9. ADC1_IRQSTS_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
9	START_OF_SWIPE	R/W	0h	Write 0: No action. Write 1: Set event (debug). Read 0: No event pending. Read 1: Event pending.
8	OUT_OF_RANGE	R/W	0h	Write 0: No action. Write 1: Set event (debug). Read 0: No event pending. Read 1: Event pending.
7	FIFO1_UNDERFLOW	R/W	0h	Write 0: No action. Write 1: Set event (debug). Read 0: No event pending. Read 1: Event pending.
6	FIFO1_OVERRUN	R/W	0h	Write 0: No action. Write 1: Set event (debug). Read 0: No event pending. Read 1: Event pending.
5	FIFO1_THR	R/W	0h	Write 0: No action. Write 1: Set event (debug). Read 0: No event pending. Read 1: Event pending.
4	FIFO0_UNDERFLOW	R/W	0h	Write 0: No action. Write 1: Set event (debug). Read 0: No event pending. Read 1: Event pending.
3	FIFO0_OVERRUN	R/W	0h	Write 0: No action. Write 1: Set event (debug). Read 0: No event pending. Read 1: Event pending.
2	FIFO0_THR	R/W	0h	Write 0: No action. Write 1: Set event (debug). Read 0: No event pending. Read 1: Event pending.
1	END_OF_SEQUENCE	R/W	0h	Write 0: No action. Write 1: Set event (debug). Read 0: No event pending. Read 1: Event pending.
0	RESERVED	R		

12.4.1.4 ADC1_IRQSTS Register (offset = 28h) [reset = 0h]

ADC1_IRQSTS is shown in Figure 12-11 and described in Table 12-10.

Figure 12-11. ADC1_IRQSTS Register

Table 12-10. ADC1_IRQSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
9	START_OF_SWIPE	R/W	0h	Write 0: No action. Read 0: No (enabled) event pending. Read 1: Event pending. Write 1: Clear (raw) event.
8	OUT_OF_RANGE	R/W	0h	Write 0: No action. Read 0: No (enabled) event pending. Read 1: Event pending. Write 1: Clear (raw) event.
7	FIFO1_UNDERFLOW	R/W	0h	Write 0: No action. Read 0: No (enabled) event pending. Read 1: Event pending. Write 1: Clear (raw) event.
6	FIFO1_OVERRUN	R/W	0h	Write 0: No action. Read 0: No (enabled) event pending. Read 1: Event pending. Write 1: Clear (raw) event.
5	FIFO1_THR	R/W	0h	Write 0: No action. Read 0: No (enabled) event pending. Read 1: Event pending. Write 1: Clear (raw) event.
4	FIFO0_UNDERFLOW	R/W	0h	Write 0: No action. Read 0: No (enabled) event pending. Read 1: Event pending. Write 1: Clear (raw) event.
3	FIFO0_OVERRUN	R/W	0h	Write 0: No action. Read 0: No (enabled) event pending. Read 1: Event pending. Write 1: Clear (raw) event.
2	FIFO0_THR	R/W	0h	Write 0: No action. Read 0: No (enabled) event pending. Read 1: Event pending. Write 1: Clear (raw) event.
1	END_OF_SEQUENCE	R/W	0h	Write 0: No action. Read 0: No (enabled) event pending. Read 1: Event pending. Write 1: Clear (raw) event.
0	RESERVED	R		

12.4.1.5 ADC1_IRQEN_SET Register (offset = 2Ch) [reset = 0h]

ADC1_IRQEN_SET is shown in Figure 12-12 and described in Table 12-11.

Figure 12-12. ADC1_IRQEN_SET Register

Table 12-11. ADC1_IRQEN_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
9	START_OF_SWIPE	R/W	0h	Write 0: No action. Read 0: Interrupt disabled (masked). Read 1: Interrupt enabled. Write 1: Enable interrupt.
8	OUT_OF_RANGE	R/W	0h	Write 0: No action. Read 0: Interrupt disabled (masked). Read 1: Interrupt enabled. Write 1: Enable interrupt.
7	FIFO1_UNDERFLOW	R/W	0h	Write 0: No action. Read 0: Interrupt disabled (masked). Read 1: Interrupt enabled. Write 1: Enable interrupt.
6	FIFO1_OVERRUN	R/W	0h	Write 0: No action. Read 0: Interrupt disabled (masked). Read 1: Interrupt enabled. Write 1: Enable interrupt.
5	FIFO1_THR	R/W	0h	Write 0: No action. Read 0: Interrupt disabled (masked). Read 1: Interrupt enabled. Write 1: Enable interrupt.
4	FIFO0_UNDERFLOW	R/W	0h	Write 0: No action. Read 0: Interrupt disabled (masked). Read 1: Interrupt enabled. Write 1: Enable interrupt.
3	FIFO0_OVERRUN	R/W	0h	Write 0: No action. Read 0: Interrupt disabled (masked). Read 1: Interrupt enabled. Write 1: Enable interrupt.
2	FIFO0_THR	R/W	0h	Write 0: No action. Read 0: Interrupt disabled (masked). Read 1: Interrupt enabled. Write 1: Enable interrupt.
1	END_OF_SEQUENCE	R/W	0h	Write 0: No action. Read 0: Interrupt disabled (masked). Read 1: Interrupt enabled. Write 1: Enable interrupt.
0	RESERVED	R		

12.4.1.6 ADC1_IRQEN_CLR Register (offset = 30h) [reset = 0h]

ADC1_IRQEN_CLR is shown in Figure 12-13 and described in Table 12-12.

Figure 12-13. ADC1_IRQEN_CLR Register

Table 12-12. ADC1_IRQEN_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
9	START_OF_SWIPE	R/W	0h	Write 0: No action. Read 0: Interrupt disabled (masked). Read 1: Interrupt enabled. Write 1: Disable interrupt.
8	OUT_OF_RANGE	R/W	0h	Write 0: No action. Read 0: Interrupt disabled (masked). Read 1: Interrupt enabled. Write 1: Disable interrupt.
7	FIFO1_UNDERFLOW	R/W	0h	Write 0: No action. Read 0: Interrupt disabled (masked). Read 1: Interrupt enabled. Write 1: Disable interrupt.
6	FIFO1_OVERRUN	R/W	0h	Write 0: No action. Read 0: Interrupt disabled (masked). Read 1: Interrupt enabled. Write 1: Disable interrupt.
5	FIFO1_THR	R/W	0h	Write 0: No action. Read 0: Interrupt disabled (masked). Read 1: Interrupt enabled. Write 1: Disable interrupt.
4	FIFO0_UNDERFLOW	R/W	0h	Write 0: No action. Read 0: Interrupt disabled (masked). Read 1: Interrupt enabled. Write 1: Disable interrupt.
3	FIFO0_OVERRUN	R/W	0h	Write 0: No action. Read 0: Interrupt disabled (masked). Read 1: Interrupt enabled. Write 1: Disable interrupt.
2	FIFO0_THR	R/W	0h	Write 0: No action. Read 0: Interrupt disabled (masked). Read 1: Interrupt enabled. Write 1: Disable interrupt.
1	END_OF_SEQUENCE	R/W	0h	Write 0: No action. Read 0: Interrupt disabled (masked). Read 1: Interrupt enabled. Write 1: Disable interrupt.
0	RESERVED	R		

12.4.1.7 ADC1_DMAEN_SET Register (offset = 38h) [reset = 0h]

ADC1_DMAEN_SET is shown in Figure 12-14 and described in Table 12-13.

Figure 12-14. ADC1_DMAEN_SET Register

Table 12-13. ADC1_DMAEN_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
1	EN1	R/W	0h	Enable DMA request FIFO1. Write 0: No action. Read 0: DMA line disabled. Read 1: DMA line enabled. Write 1: Enable DMA line.
0	EN0	R/W	0h	Enable DMA request FIFO0. Write 0: No action. Read 0: DMA line disabled. Read 1: DMA line enabled. Write 1: Enable DMA line.

12.4.1.8 ADC1_DMAEN_CLR Register (offset = 3Ch) [reset = 0h]

ADC1_DMAEN_CLR is shown in Figure 12-15 and described in Table 12-14.

Figure 12-15. ADC1_DMAEN_CLR Register

Table 12-14. ADC1_DMAEN_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
1	EN1	R/W	0h	Disable DMA request FIFO1. Write 0: No action. Read 0: DMA line disabled. Read 1: DMA line enabled. Write 1: Disable DMA line.
0	EN0	R/W	0h	Disable DMA request FIFO0. Write 0 : No action. Read 0 : DMA line disabled. Read 1: DMA line enabled. Write 1: Disable DMA line.

12.4.1.9 ADC1_CTRL Register (offset = 40h) [reset = 0h]

ADC1_CTRL is shown in Figure 12-16 and described in Table 12-15.

Figure 12-16. ADC1_CTRL Register

Table 12-15. ADC1_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
9	RESERVED	R		
8	HW_EVT_MAPPING	R/W	0h	0->Map hardware event to the start of swipe interrupt. 1->Map hardware event to hardware event input (external).
7	RESERVED	R		
6	PREAMP_BYPASS	R/W	0h	Analog input will bypass the PreAmplifier: 0: PreAmp is enabled by default. 1 : Write 1 to bypass PreAmp.
5	PREAMP_PD	R/W	0h	PreAmp Power Down control. 0: PreAmp is powered up (default). 1 : Write 1 to power down PreAmp.
4	ADC_POWER_DOWN	R/W	0h	ADC Power Down control. 0: AFE is powered up (default). 1 : Write 1 to power down AFE.
3	RESERVED	R		
2	SIMULTANEOUS_CTRL	R/W	0h	Enables Simultaneous Control of both AFEs. 0 -> Control only MCR_AFE. 1 -> Control both MCR_AFE and TSC_AFE. See the Programmers/Usage guide section for Simultaneous Mode Features/Limitations.
1	STEP_ID_TAG	R/W	0h	Writing 1 to this bit will store the Step ID number with the captured ADC data in the FIFO. 0: Write zeros. 1: Store the channel ID tag.
0	ADC1_EN	R/W	0h	ADC1 module enable bit. After programming all the steps and configuration registers, write a 1 to this bit to turn on ADC1. Writing a 0 will disable the module (after the current conversion is completed).

12.4.1.10 ADC1_ADCSTAT Register (offset = 44h) [reset = 10h]

ADC1_ADCSTAT is shown in Figure 12-17 and described in Table 12-16.

Figure 12-17. ADC1_ADCSTAT Register

Table 12-16. ADC1_ADCSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
5	FSM_BUSY	R	0h	Status of OCP FSM and ADC FSM. 0 idle. 1 - busy.
4-0	STEP_ID	R	10h	Below are encoded values: 10000 Idle. 10001 Reserved. 00000 Step 1. 00001 Step 2. 00010 Step 3. 00011 Step 4. 00100 Step 5. 00101 Step 6. 00110 Step 7. 00111 Step 8. 01000 Step 9. 01001 Step 10. 01010 Step 11. 01011 Step 12. 01100 Step 13. 01101 Step 14. 01110 Step 15. 01111 Step 16.

12.4.1.11 ADC1_ADCRANGE Register (offset = 48h) [reset = 0h]

ADC1_ADCRANGE is shown in Figure 12-18 and described in Table 12-17.

Figure 12-18. ADC1_ADCRANGE Register

Table 12-17. ADC1_ADCRANGE Register Field Descriptions

Bit	Field	Type	Reset	Description
27-16	THR_HIGH_RANGE_DATA	R/W	0h	Sampled ADC data is compared to this value. If the interrupt is enabled, and if the sampled data is > value then interrupt is generated. Each step can enable or disable this check.
15-12	RESERVED	R		
11-0	THR_LOW_RANGE_DATA	R/W	0h	Sampled ADC data is compared to this value. If the interrupt is enabled, and if the sampled data is < value then interrupt is generated. Each step can enable or disable this check.

12.4.1.12 ADC1_CLKDIV Register (offset = 4Ch) [reset = 0h]

ADC1_CLKDIV is shown in Figure 12-19 and described in Table 12-18.

Figure 12-19. ADC1_CLKDIV Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC_CLKDIV															
R/W-0h															

Table 12-18. ADC1_CLKDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	ADC_CLKDIV	R/W	0h	The input ADC clock will be divided by this value and sent to the AFE. Program to the value minus 1.

12.4.1.13 ADC1_STEPEN Register (offset = 54h) [reset = 0h]

ADC1_STEPEN is shown in Figure 12-20 and described in Table 12-19.

Figure 12-20. ADC1_STEPEN Register

Table 12-19. ADC1_STEPEN Register Field Descriptions

Bit	Field	Type	Reset	Description
16	STEP16	R/W	0h	Enable step 16.
15	STEP15	R/W	0h	Enable step 15.
14	STEP14	R/W	0h	Enable step 14.
13	STEP13	R/W	0h	Enable step 13.
12	STEP12	R/W	0h	Enable step 12.
11	STEP11	R/W	0h	Enable step 11.
10	STEP10	R/W	0h	Enable step 10.
9	STEP9	R/W	0h	Enable step 9.
8	STEP8	R/W	0h	Enable step 8.
7	STEP7	R/W	0h	Enable step 7.
6	STEP6	R/W	0h	Enable step 6.
5	STEP5	R/W	0h	Enable step 5.
4	STEP4	R/W	0h	Enable step 4.
3	STEP3	R/W	0h	Enable step 3.
2	STEP2	R/W	0h	Enable step 2.
1	STEP1	R/W	0h	Enable step 1.
0	RESERVED	R		

12.4.1.14 ADC1_IDLECONFIG Register (offset = 58h) [reset = 0h]

ADC1_IDLECONFIG is shown in Figure 12-21 and described in Table 12-20.

Figure 12-21. ADC1_IDLECONFIG Register

Table 12-20. ADC1_IDLECONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
29-28	GAIN_CTRL4	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
27-26	RESERVED	R		
25	DIFF_CNTRL	R/W	0h	Differential Control Pin. 0 -> Single Ended. 1 -> Differential Pair Enable.
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins software configuration. 00 - VSSA. 01 - VSSA. 10 - VSSA. 11 ADCREFM.
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins software configuration. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
18-15	SEL_INM_SWM_3_0	R/W	0h	SEL_INM pins for neg differential. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins software configuration. 000 - VDDA. 001 - VDDA. 010 - VDDA. 011 - ADCREFP. 1xx - Reserved.
11	RESERVED	R		
10-9	GAIN_CTRL3	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
8-7	GAIN_CTRL2	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
6-5	GAIN_CTRL1	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
4-0	RESERVED	R		

12.4.1.15 ADC1_SWIPE_COMPARE_REG1_2 Register (offset = 5Ch) [reset = 0h]

ADC1_SWIPE_COMPARE_REG1_2 is shown in Figure 12-22 and described in Table 12-21.

Figure 12-22. ADC1_SWIPE_COMPARE_REG1_2 Register

Table 12-21. ADC1_SWIPE_COMPARE_REG1_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-16	THR_DATA1	R/W	0h	Sampled ADC data is compared to this value. If the swipe feature is enabled, and if the sampled data is >= value then data is saved in the FIFO (and also subsequent data is saved for all steps).
15-12	RESERVED	R		
11-0	THR_DATA2	R/W	0h	Sampled ADC data is compared to this value. If the swipe feature is enabled, and if the sampled data is >= value then data is saved in the FIFO (and also subsequent data is saved for all steps).

12.4.1.16 ADC1_SWIPE_COMPARE_REG3_4 Register (offset = 60h) [reset = 0h]

ADC1_SWIPE_COMPARE_REG3_4 is shown in Figure 12-23 and described in Table 12-22.

Figure 12-23. ADC1_SWIPE_COMPARE_REG3_4 Register

Table 12-22. ADC1_SWIPE_COMPARE_REG3_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-16	THR_DATA3	R/W	0h	Sampled ADC data is compared to this value. If the swipe feature is enabled, and if the sampled data is >= value then data is saved in the FIFO (and also subsequent data is saved for all steps).
15-12	RESERVED	R		
11-0	THR_DATA4	R/W	0h	Sampled ADC data is compared to this value. If the swipe feature is enabled, and if the sampled data is >= value then data is saved in the FIFO (and also subsequent data is saved for all steps).

12.4.1.17 ADC1_STEPCONFIG1 Register (offset = 64h) [reset = 0h]

ADC1_STEPCONFIG1 is shown in Figure 12-24 and described in Table 12-23.

Figure 12-24. ADC1_STEPCONFIG1 Register

31	30	29	28	27	26	25	24
SWIPE_THR_REG_POINTER		GAIN_CTRL4		RANGE_CHECK_INTR	FIFO_SELECT	DIFF_CNTRL	SEL_RFM_SWC_1_0
R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM_3_0		
R/W-0h	R/W-0h				R/W-0h		
15	14	13	12	11	10	9	8
SEL_INM_SWM_3_0	SEL_RFP_SWC_2_0			SWIPE_THR_COMPARE_FEATURE	GAIN_CTRL3		GAIN_CTRL2
R/W-0h	R/W-0h			R/W-0h	R/W-0h		R/W-0h
7	6	5	4	3	2	1	0
GAIN_CTRL2	GAIN_CTRL1		AVERAGING			MODE	
R/W-0h	R/W-0h		R/W-0h			R/W-0h	

Table 12-23. ADC1_STEPCONFIG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SWIPE_THR_REG_POINTER	R/W	0h	If the Swipe Data Compare feature is enabled (refer to bit 11) then: 00 Trigger using swipe threshold reg1. 01 Trigger using swipe threshold reg 2. 10 Trigger using swipe threshold reg 3. 11 Trigger using swipe threshold reg 4.
29-28	GAIN_CTRL4	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
27	RANGE_CHECK_INTR	R/W	0h	0 - disable out of range check. 1 - Compare adc data with range check register.
26	FIFO_SELECT	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 0. 1 -> FIFO 1.
25	DIFF_CNTRL	R/W	0h	Differential Control Pin. 0 -> Single Ended. 1 -> Differential Pair Enable.
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins software configuration. 00 - VSSA. 01 - VSSA. 10 - VSSA. 11 ADCREFM.
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins software configuration. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
18-15	SEL_INM_SWM_3_0	R/W	0h	SEL_INM pins for neg differential. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins software configuration. 000 - VDDA. 001 - VDDA. 010 - VDDA. 011 - ADCREFP. 1xx - Reserved.

Table 12-23. ADC1_STEPCONFIG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SWIPE_THR_COMPARE_FEATURE	R/W	0h	Set this bit to 1 to enable the swipe data comparing feature: 0: Feature is off (data is always saved). 1: Trigger feature is on if adc data < swipe data threshold map pointer value then discard data otherwise data is saved (and all future steps are saved) until mag_adc enable (bit 0) is turned off.
10-9	GAIN_CTRL3	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
8-7	GAIN_CTRL2	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
6-5	GAIN_CTRL1	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
4-2	AVERAGING	R/W	0h	Number of samplings to average: 000 -> no average. 001 -> 2 samples average. 010 -> 4 samples average. 011 -> 8 samples average. 100 -> 16 samples average.
1-0	MODE	R/W	0h	00 Software enabled, one-shot. 01 Software enabled, continuous. 10 Hardware synchronized, one-shot. 11 Hardware synchronized, continuous.

12.4.1.18 ADC1_STEPDELAY1 Register (offset = 68h) [reset = 0h]

ADC1_STEPDELAY1 is shown in Figure 12-25 and described in Table 12-24.

Figure 12-25. ADC1_STEPDELAY1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">SAMPLEDELAY</td><td colspan="6">RESERVED</td><td colspan="2">OPENDELAY</td></tr>
<tr><td colspan="8">R/W-0h</td><td colspan="6">R-</td><td colspan="2">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">OPENDELAY</td></tr>
<tr><td colspan="16">R/W-0h</td></tr>
</table>

Table 12-24. ADC1_STEPDELAY1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SAMPLEDELAY	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SoC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	RESERVED	R		
17-0	OPENDELAY	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion.

12.4.1.19 ADC1_STEPCONFIG2 Register (offset = 6Ch) [reset = 0h]

ADC1_STEPCONFIG2 is shown in Figure 12-26 and described in Table 12-25.

Figure 12-26. ADC1_STEPCONFIG2 Register

31	30	29	28	27	26	25	24
SWIPE_THR_REG_POINTER		GAIN_CTRL4		RANGE_CHECK_INTR	FIFO_SELECT	DIFF_CNTRL	SEL_RFM_SWC_1_0
R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM_3_0		
R/W-0h	R/W-0h				R/W-0h		
15	14	13	12	11	10	9	8
SEL_INM_SWM_3_0	SEL_RFP_SWC_2_0			SWIPE_THR_COMPARE_FEATURE	GAIN_CTRL3		GAIN_CTRL2
R/W-0h	R/W-0h			R/W-0h	R/W-0h		R/W-0h
7	6	5	4	3	2	1	0
GAIN_CTRL2	GAIN_CTRL1		AVERAGING			MODE	
R/W-0h	R/W-0h		R/W-0h			R/W-0h	

Table 12-25. ADC1_STEPCONFIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SWIPE_THR_REG_POINTER	R/W	0h	If the Swipe Data Compare feature is enabled (refer to bit 11) then: 00 Trigger using swipe threshold reg1. 01 Trigger using swipe threshold reg 2. 10 Trigger using swipe threshold reg 3. 11 Trigger using swipe threshold reg 4.
29-28	GAIN_CTRL4	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
27	RANGE_CHECK_INTR	R/W	0h	0 - disable out of range check. 1 - Compare adc data with range check register.
26	FIFO_SELECT	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 0. 1 -> FIFO 1.
25	DIFF_CNTRL	R/W	0h	Differential Control Pin. 0 -> Single Ended. 1 -> Differential Pair Enable.
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins software configuration. 00 - VSSA. 01 - VSSA. 10 - VSSA. 11 ADCREFM.
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins software configuration. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
18-15	SEL_INM_SWM_3_0	R/W	0h	SEL_INM pins for neg differential. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins software configuration. 000 - VDDA. 001 - VDDA. 010 - VDDA. 011 - ADCREFP. 1xx - Reserved.

Table 12-25. ADC1_STEPCONFIG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SWIPE_THR_COMPARE_FEATURE	R/W	0h	Set this bit to 1 to enable the swipe data comparing feature: 0: Feature is off (data is always saved). 1: Trigger feature is on if adc data < swipe data threshold map pointer value then discard data otherwise data is saved (and all future steps are saved) until mag_adc enable (bit 0) is turned off.
10-9	GAIN_CTRL3	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
8-7	GAIN_CTRL2	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
6-5	GAIN_CTRL1	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
4-2	AVERAGING	R/W	0h	Number of samplings to average: 000 -> no average. 001 -> 2 samples average. 010 -> 4 samples average. 011 -> 8 samples average. 100 -> 16 samples average.
1-0	MODE	R/W	0h	00 Software enabled, one-shot. 01 Software enabled, continuous. 10 Hardware synchronized, one-shot. 11 Hardware synchronized, continuous.

12.4.1.20 ADC1_STEPDELAY2 Register (offset = 70h) [reset = 0h]

ADC1_STEPDELAY2 is shown in Figure 12-27 and described in Table 12-26.

Figure 12-27. ADC1_STEPDELAY2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAMPLEDELAY								RESERVED						OPENDELAY	
R/W-0h								R-						R/W-0h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OPENDELAY															
R/W-0h															

Table 12-26. ADC1_STEPDELAY2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SAMPLEDELAY	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SoC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	RESERVED	R		
17-0	OPENDELAY	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion.

12.4.1.21 ADC1_STEPCONFIG3 Register (offset = 74h) [reset = 0h]

ADC1_STEPCONFIG3 is shown in Figure 12-28 and described in Table 12-27.

Figure 12-28. ADC1_STEPCONFIG3 Register

31	30	29	28	27	26	25	24
SWIPE_THR_REG_POINTER		GAIN_CTRL4		RANGE_CHECK_INTR	FIFO_SELECT	DIFF_CNTRL	SEL_RFM_SWC_1_0
R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM_3_0		
R/W-0h	R/W-0h				R/W-0h		
15	14	13	12	11	10	9	8
SEL_INM_SWM_3_0	SEL_RFP_SWC_2_0			SWIPE_THR_COMPARE_FEATURE	GAIN_CTRL3		GAIN_CTRL2
R/W-0h	R/W-0h			R/W-0h	R/W-0h		R/W-0h
7	6	5	4	3	2	1	0
GAIN_CTRL2	GAIN_CTRL1		AVERAGING			MODE	
R/W-0h	R/W-0h		R/W-0h			R/W-0h	

Table 12-27. ADC1_STEPCONFIG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SWIPE_THR_REG_POINTER	R/W	0h	If the Swipe Data Compare feature is enabled (refer to bit 11) then: 00 Trigger using swipe threshold reg1. 01 Trigger using swipe threshold reg 2. 10 Trigger using swipe threshold reg 3. 11 Trigger using swipe threshold reg 4.
29-28	GAIN_CTRL4	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
27	RANGE_CHECK_INTR	R/W	0h	0 disable out of range check. 1 - Compare adc data with range check register.
26	FIFO_SELECT	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 0. 1 -> FIFO 1.
25	DIFF_CNTRL	R/W	0h	Differential Control Pin. 0 -> Single Ended. 1 -> Differential Pair Enable.
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins software configuration. 00 - VSSA. 01 - VSSA. 10 - VSSA. 11 ADCREFM.
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins software configuration. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
18-15	SEL_INM_SWM_3_0	R/W	0h	SEL_INM pins for neg differential. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins software configuration. 000 - VDDA. 001 - VDDA. 010 - VDDA. 011 - ADCREFP. 1xx - Reserved.

Table 12-27. ADC1_STEPCONFIG3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SWIPE_THR_COMPARE_FEATURE	R/W	0h	Set this bit to 1 to enable the swipe data comparing feature: 0: Feature is off (data is always saved). 1: Trigger feature is on if adc data < swipe data threshold map pointer value then discard data otherwise data is saved (and all future steps are saved) until mag_adc enable (bit 0) is turned off.
10-9	GAIN_CTRL3	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
8-7	GAIN_CTRL2	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
6-5	GAIN_CTRL1	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
4-2	AVERAGING	R/W	0h	Number of samplings to average: 000 -> no average. 001 -> 2 samples average. 010 -> 4 samples average. 011 -> 8 samples average. 100 -> 16 samples average.
1-0	MODE	R/W	0h	00 Software enabled, one-shot. 01 Software enabled, continuous. 10 Hardware synchronized, one-shot. 11 Hardware synchronized, continuous.

12.4.1.22 ADC1_STEPDELAY3 Register (offset = 78h) [reset = 0h]

ADC1_STEPDELAY3 is shown in Figure 12-29 and described in Table 12-28.

Figure 12-29. ADC1_STEPDELAY3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAMPLEDELAY								RESERVED						OPENDELAY	
R/W-0h								R-						R/W-0h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OPENDELAY															
R/W-0h															

Table 12-28. ADC1_STEPDELAY3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SAMPLEDELAY	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SoC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	RESERVED	R		
17-0	OPENDELAY	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion.

12.4.1.23 ADC1_STEPCONFIG4 Register (offset = 7Ch) [reset = 0h]

ADC1_STEPCONFIG4 is shown in Figure 12-30 and described in Table 12-29.

Figure 12-30. ADC1_STEPCONFIG4 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="2">SWIPE_THR_REG_POINTER</td><td colspan="2">GAIN_CTRL4</td><td>RANGE_CHECK_INTR</td><td>FIFO_SELECT</td><td>DIFF_CNTRL</td><td>SEL_RFM_SWC_1_0</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>SEL_RFM_SWC_1_0</td><td colspan="4">SEL_INP_SWC_3_0</td><td colspan="3">SEL_INM_SWM_3_0</td></tr>
<tr><td>R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="3">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>SEL_INM_SWM_3_0</td><td colspan="3">SEL_RFP_SWC_2_0</td><td>SWIPE_THR_COMPARE_FEATURE</td><td colspan="2">GAIN_CTRL3</td><td>GAIN_CTRL2</td></tr>
<tr><td>R/W-0h</td><td colspan="3">R/W-0h</td><td>R/W-0h</td><td colspan="2">R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>GAIN_CTRL2</td><td colspan="2">GAIN_CTRL1</td><td colspan="3">AVERAGING</td><td colspan="2">MODE</td></tr>
<tr><td>R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="3">R/W-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 12-29. ADC1_STEPCONFIG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SWIPE_THR_REG_POINTER	R/W	0h	If the Swipe Data Compare feature is enabled (refer to bit 11) then: 00 Trigger using swipe threshold reg1. 01 Trigger using swipe threshold reg 2. 10 Trigger using swipe threshold reg 3. 11 Trigger using swipe threshold reg 4.
29-28	GAIN_CTRL4	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
27	RANGE_CHECK_INTR	R/W	0h	0 disable out of range check. 1 - Compare adc data with range check register.
26	FIFO_SELECT	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 0. 1 -> FIFO 1.
25	DIFF_CNTRL	R/W	0h	Differential Control Pin. 0 -> Single Ended. 1 -> Differential Pair Enable.
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins software configuration. 00 - VSSA. 01 - VSSA. 10 - VSSA. 11 ADCREFM.
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins software configuration. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
18-15	SEL_INM_SWM_3_0	R/W	0h	SEL_INM pins for neg differential. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins software configuration. 000 - VDDA. 001 - VDDA. 010 - VDDA. 011 - ADCREFP. 1xx - Reserved.

Table 12-29. ADC1_STEPCONFIG4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SWIPE_THR_COMPARE_FEATURE	R/W	0h	Set this bit to 1 to enable the swipe data comparing feature: 0: Feature is off (data is always saved). 1: Trigger feature is on if adc data < swipe data threshold map pointer value then discard data otherwise data is saved (and all future steps are saved) until mag_adc enable (bit 0) is turned off.
10-9	GAIN_CTRL3	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
8-7	GAIN_CTRL2	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
6-5	GAIN_CTRL1	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
4-2	AVERAGING	R/W	0h	Number of samplings to average: 000 -> no average. 001 -> 2 samples average. 010 -> 4 samples average. 011 -> 8 samples average. 100 -> 16 samples average.
1-0	MODE	R/W	0h	00 Software enabled, one-shot. 01 Software enabled, continuous. 10 Hardware synchronized, one-shot. 11 Hardware synchronized, continuous.

12.4.1.24 ADC1_STEPDELAY4 Register (offset = 80h) [reset = 0h]

ADC1_STEPDELAY4 is shown in Figure 12-31 and described in Table 12-30.

Figure 12-31. ADC1_STEPDELAY4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAMPLEDELAY								RESERVED						OPENDELAY	
R/W-0h								R-						R/W-0h	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPENDELAY															
R/W-0h															

Table 12-30. ADC1_STEPDELAY4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SAMPLEDELAY	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SoC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	RESERVED	R		
17-0	OPENDELAY	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion.

12.4.1.25 ADC1_STEPCONFIG5 Register (offset = 84h) [reset = 0h]

ADC1_STEPCONFIG5 is shown in Figure 12-32 and described in Table 12-31.

Figure 12-32. ADC1_STEPCONFIG5 Register

31	30	29	28	27	26	25	24
SWIPE_THR_REG_POINTER		GAIN_CTRL4		RANGE_CHECK_INTR	FIFO_SELECT	DIFF_CNTRL	SEL_RFM_SWC_1_0
R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM_3_0		
R/W-0h	R/W-0h				R/W-0h		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SEL_INM_SWM_3_0	SEL_RFP_SWC_2_0			SWIPE_THR_COMPARE_FEATURE	GAIN_CTRL3		GAIN_CTRL2
R/W-0h	R/W-0h			R/W-0h	R/W-0h		R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GAIN_CTRL2	GAIN_CTRL1		AVERAGING			MODE	
R/W-0h	R/W-0h		R/W-0h			R/W-0h	

Table 12-31. ADC1_STEPCONFIG5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SWIPE_THR_REG_POINTER	R/W	0h	If the Swipe Data Compare feature is enabled (refer to bit 11) then: 00 Trigger using swipe threshold reg1. 01 Trigger using swipe threshold reg 2. 10 Trigger using swipe threshold reg 3. 11 Trigger using swipe threshold reg 4.
29-28	GAIN_CTRL4	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
27	RANGE_CHECK_INTR	R/W	0h	0 disable out of range check. 1 - Compare adc data with range check register.
26	FIFO_SELECT	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 0. 1 -> FIFO 1.
25	DIFF_CNTRL	R/W	0h	Differential Control Pin. 0 -> Single Ended. 1 -> Differential Pair Enable.
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins software configuration. 00 - VSSA. 01 - VSSA. 10 - VSSA. 11 ADCREFM.
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins software configuration. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
18-15	SEL_INM_SWM_3_0	R/W	0h	SEL_INM pins for neg differential. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins software configuration. 000 - VDDA. 001 - VDDA. 010 - VDDA. 011 - ADCREFP. 1xx - Reserved.

Table 12-31. ADC1_STEPCONFIG5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SWIPE_THR_COMPARE_FEATURE	R/W	0h	Set this bit to 1 to enable the swipe data comparing feature: 0: Feature is off (data is always saved). 1: Trigger feature is on if adc data < swipe data threshold map pointer value then discard data otherwise data is saved (and all future steps are saved) until mag_adc enable (bit 0) is turned off.
10-9	GAIN_CTRL3	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
8-7	GAIN_CTRL2	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
6-5	GAIN_CTRL1	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
4-2	AVERAGING	R/W	0h	Number of samplings to average: 000 -> no average. 001 -> 2 samples average. 010 -> 4 samples average. 011 -> 8 samples average. 100 -> 16 samples average.
1-0	MODE	R/W	0h	00 Software enabled, one-shot. 01 Software enabled, continuous. 10 Hardware synchronized, one-shot. 11 Hardware synchronized, continuous.

12.4.1.26 ADC1_STEPDELAY5 Register (offset = 88h) [reset = 0h]

ADC1_STEPDELAY5 is shown in Figure 12-33 and described in Table 12-32.

Figure 12-33. ADC1_STEPDELAY5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAMPLEDELAY								RESERVED						OPENDELAY	
R/W-0h								R-						R/W-0h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OPENDELAY															
R/W-0h															

Table 12-32. ADC1_STEPDELAY5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SAMPLEDELAY	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SoC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	RESERVED	R		
17-0	OPENDELAY	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion.

12.4.1.27 ADC1_STEPCONFIG6 Register (offset = 8Ch) [reset = 0h]

ADC1_STEPCONFIG6 is shown in Figure 12-34 and described in Table 12-33.

Figure 12-34. ADC1_STEPCONFIG6 Register

31	30	29	28	27	26	25	24
SWIPE_THR_REG_POINTER		GAIN_CTRL4		RANGE_CHECK_INTR	FIFO_SELECT	DIFF_CNTRL	SEL_RFM_SWC_1_0
R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM_3_0		
R/W-0h	R/W-0h				R/W-0h		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SEL_INM_SWM_3_0	SEL_RFP_SWC_2_0			SWIPE_THR_COMPARE_FEATURE	GAIN_CTRL3		GAIN_CTRL2
R/W-0h	R/W-0h			R/W-0h	R/W-0h		R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GAIN_CTRL2	GAIN_CTRL1		AVERAGING			MODE	
R/W-0h	R/W-0h		R/W-0h			R/W-0h	

Table 12-33. ADC1_STEPCONFIG6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SWIPE_THR_REG_POINTER	R/W	0h	If the Swipe Data Compare feature is enabled (refer to bit 11) then: 00 Trigger using swipe threshold reg1. 01 Trigger using swipe threshold reg 2. 10 Trigger using swipe threshold reg 3. 11 Trigger using swipe threshold reg 4.
29-28	GAIN_CTRL4	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
27	RANGE_CHECK_INTR	R/W	0h	0 disable out of range check. 1 - Compare adc data with range check register.
26	FIFO_SELECT	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 0. 1 -> FIFO 1.
25	DIFF_CNTRL	R/W	0h	Differential Control Pin. 0 -> Single Ended. 1 -> Differential Pair Enable.
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins software configuration. 00 - VSSA. 01 - VSSA. 10 - VSSA. 11 ADCREFM.
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins software configuration. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
18-15	SEL_INM_SWM_3_0	R/W	0h	SEL_INM pins for neg differential. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins software configuration. 000 - VDDA. 001 - VDDA. 010 - VDDA. 011 - ADCREFP. 1xx - Reserved.

Table 12-33. ADC1_STEPCONFIG6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SWIPE_THR_COMPARE_FEATURE	R/W	0h	Set this bit to 1 to enable the swipe data comparing feature: 0: Feature is off (data is always saved). 1: Trigger feature is on if adc data < swipe data threshold map pointer value then discard data otherwise data is saved (and all future steps are saved) until mag_adc enable (bit 0) is turned off.
10-9	GAIN_CTRL3	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
8-7	GAIN_CTRL2	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
6-5	GAIN_CTRL1	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
4-2	AVERAGING	R/W	0h	Number of samplings to average: 000 -> no average. 001 -> 2 samples average. 010 -> 4 samples average. 011 -> 8 samples average. 100 -> 16 samples average.
1-0	MODE	R/W	0h	00 Software enabled, one-shot. 01 Software enabled, continuous. 10 Hardware synchronized, one-shot. 11 Hardware synchronized, continuous.

12.4.1.28 ADC1_STEPDELAY6 Register (offset = 90h) [reset = 0h]

ADC1_STEPDELAY6 is shown in Figure 12-35 and described in Table 12-34.

Figure 12-35. ADC1_STEPDELAY6 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">SAMPLEDELAY</td><td colspan="6">RESERVED</td><td colspan="2">OPENDELAY</td></tr>
<tr><td colspan="8">R/W-0h</td><td colspan="6">R-</td><td colspan="2">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">OPENDELAY</td></tr>
<tr><td colspan="16">R/W-0h</td></tr>
</table>

Table 12-34. ADC1_STEPDELAY6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SAMPLEDELAY	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SoC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	RESERVED	R		
17-0	OPENDELAY	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion.

12.4.1.29 ADC1_STEPCONFIG7 Register (offset = 94h) [reset = 0h]

ADC1_STEPCONFIG7 is shown in Figure 12-36 and described in Table 12-35.

Figure 12-36. ADC1_STEPCONFIG7 Register

31	30	29	28	27	26	25	24
SWIPE_THR_REG_POINTER		GAIN_CTRL4		RANGE_CHECK_INTR	FIFO_SELECT	DIFF_CNTRL	SEL_RFM_SWC_1_0
R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM_3_0		
R/W-0h	R/W-0h				R/W-0h		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SEL_INM_SWM_3_0	SEL_RFP_SWC_2_0			SWIPE_THR_COMPARE_FEATURE	GAIN_CTRL3		GAIN_CTRL2
R/W-0h	R/W-0h			R/W-0h	R/W-0h		R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GAIN_CTRL2	GAIN_CTRL1		AVERAGING			MODE	
R/W-0h	R/W-0h		R/W-0h			R/W-0h	

Table 12-35. ADC1_STEPCONFIG7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SWIPE_THR_REG_POINTER	R/W	0h	If the Swipe Data Compare feature is enabled (refer to bit 11) then: 00 Trigger using swipe threshold reg1. 01 Trigger using swipe threshold reg 2. 10 Trigger using swipe threshold reg 3. 11 Trigger using swipe threshold reg 4.
29-28	GAIN_CTRL4	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
27	RANGE_CHECK_INTR	R/W	0h	0 disable out of range check. 1 - Compare adc data with range check register.
26	FIFO_SELECT	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 0. 1 -> FIFO 1.
25	DIFF_CNTRL	R/W	0h	Differential Control Pin. 0 -> Single Ended. 1 -> Differential Pair Enable.
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins software configuration. 00 - VSSA. 01 - VSSA. 10 - VSSA. 11 ADCREFM.
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins software configuration. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
18-15	SEL_INM_SWM_3_0	R/W	0h	SEL_INM pins for neg differential. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins software configuration. 000 - VDDA. 001 - VDDA. 010 - VDDA. 011 - ADCREFP. 1xx - Reserved.

Table 12-35. ADC1_STEPCONFIG7 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SWIPE_THR_COMPARE_FEATURE	R/W	0h	Set this bit to 1 to enable the swipe data comparing feature: 0: Feature is off (data is always saved). 1: Trigger feature is on if adc data < swipe data threshold map pointer value then discard data otherwise data is saved (and all future steps are saved) until mag_adc enable (bit 0) is turned off.
10-9	GAIN_CTRL3	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
8-7	GAIN_CTRL2	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
6-5	GAIN_CTRL1	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
4-2	AVERAGING	R/W	0h	Number of samplings to average: 000 -> no average. 001 -> 2 samples average. 010 -> 4 samples average. 011 -> 8 samples average. 100 -> 16 samples average.
1-0	MODE	R/W	0h	00 Software enabled, one-shot. 01 Software enabled, continuous. 10 Hardware synchronized, one-shot. 11 Hardware synchronized, continuous.

12.4.1.30 ADC1_STEPDELAY7 Register (offset = 98h) [reset = 0h]

ADC1_STEPDELAY7 is shown in Figure 12-37 and described in Table 12-36.

Figure 12-37. ADC1_STEPDELAY7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAMPLEDELAY								RESERVED						OPENDELAY	
R/W-0h								R-						R/W-0h	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPENDELAY															
R/W-0h															

Table 12-36. ADC1_STEPDELAY7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SAMPLEDELAY	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SoC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	RESERVED	R		
17-0	OPENDELAY	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion.

12.4.1.31 ADC1_STEPCONFIG8 Register (offset = 9Ch) [reset = 0h]

ADC1_STEPCONFIG8 is shown in Figure 12-38 and described in Table 12-37.

Figure 12-38. ADC1_STEPCONFIG8 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="2">SWIPE_THR_REG_POINTER</td><td colspan="2">GAIN_CTRL4</td><td>RANGE_CHECK_INTR</td><td>FIFO_SELECT</td><td>DIFF_CNTRL</td><td>SEL_RFM_SWC_1_0</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>SEL_RFM_SWC_1_0</td><td colspan="4">SEL_INP_SWC_3_0</td><td colspan="3">SEL_INM_SWM_3_0</td></tr>
<tr><td>R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="3">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>SEL_INM_SWM_3_0</td><td colspan="3">SEL_RFP_SWC_2_0</td><td>SWIPE_THR_COMPARE_FEATURE</td><td colspan="2">GAIN_CTRL3</td><td>GAIN_CTRL2</td></tr>
<tr><td>R/W-0h</td><td colspan="3">R/W-0h</td><td>R/W-0h</td><td colspan="2">R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>GAIN_CTRL2</td><td colspan="2">GAIN_CTRL1</td><td colspan="3">AVERAGING</td><td colspan="2">MODE</td></tr>
<tr><td>R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="3">R/W-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 12-37. ADC1_STEPCONFIG8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SWIPE_THR_REG_POINTER	R/W	0h	If the Swipe Data Compare feature is enabled (refer to bit 11) then: 00 Trigger using swipe threshold reg1. 01 Trigger using swipe threshold reg 2. 10 Trigger using swipe threshold reg 3. 11 Trigger using swipe threshold reg 4.
29-28	GAIN_CTRL4	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
27	RANGE_CHECK_INTR	R/W	0h	0 disable out of range check. 1 - Compare adc data with range check register.
26	FIFO_SELECT	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 0. 1 -> FIFO 1.
25	DIFF_CNTRL	R/W	0h	Differential Control Pin. 0 -> Single Ended. 1 -> Differential Pair Enable.
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins software configuration. 00 - VSSA. 01 - VSSA. 10 - VSSA. 11 ADCREFM.
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins software configuration. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
18-15	SEL_INM_SWM_3_0	R/W	0h	SEL_INM pins for neg differential. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins software configuration. 000 - VDDA. 001 - VDDA. 010 - VDDA. 011 - ADCREFP. 1xx - Reserved.

Table 12-37. ADC1_STEPCONFIG8 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SWIPE_THR_COMPARE_FEATURE	R/W	0h	Set this bit to 1 to enable the swipe data comparing feature: 0: Feature is off (data is always saved). 1: Trigger feature is on if adc data < swipe data threshold map pointer value then discard data otherwise data is saved (and all future steps are saved) until mag_adc enable (bit 0) is turned off.
10-9	GAIN_CTRL3	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
8-7	GAIN_CTRL2	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
6-5	GAIN_CTRL1	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
4-2	AVERAGING	R/W	0h	Number of samplings to average: 000 -> no average. 001 -> 2 samples average. 010 -> 4 samples average. 011 -> 8 samples average. 100 -> 16 samples average.
1-0	MODE	R/W	0h	00 Software enabled, one-shot. 01 Software enabled, continuous. 10 Hardware synchronized, one-shot. 11 Hardware synchronized, continuous.

12.4.1.32 ADC1_STEPDELAY8 Register (offset = A0h) [reset = 0h]

ADC1_STEPDELAY8 is shown in Figure 12-39 and described in Table 12-38.

Figure 12-39. ADC1_STEPDELAY8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAMPLEDELAY								RESERVED						OPENDELAY	
R/W-0h								R-						R/W-0h	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPENDELAY															
R/W-0h															

Table 12-38. ADC1_STEPDELAY8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SAMPLEDELAY	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SoC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	RESERVED	R		
17-0	OPENDELAY	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion.

12.4.1.33 ADC1_STEPCONFIG9 Register (offset = A4h) [reset = 0h]

ADC1_STEPCONFIG9 is shown in Figure 12-40 and described in Table 12-39.

Figure 12-40. ADC1_STEPCONFIG9 Register

31	30	29	28	27	26	25	24
SWIPE_THR_REG_POINTER		GAIN_CTRL4		RANGE_CHECK_INTR	FIFO_SELECT	DIFF_CNTRL	SEL_RFM_SWC_1_0
R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM_3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM_3_0	SEL_RFP_SWC_2_0			SWIPE_THR_COMPARE_FEATURE	GAIN_CTRL3		GAIN_CTRL2
R/W-0h	R/W-0h			R/W-0h	R/W-0h		R/W-0h

7	6	5	4	3	2	1	0
GAIN_CTRL2	GAIN_CTRL1		AVERAGING			MODE	
R/W-0h	R/W-0h		R/W-0h			R/W-0h	

Table 12-39. ADC1_STEPCONFIG9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SWIPE_THR_REG_POINTER	R/W	0h	If the Swipe Data Compare feature is enabled (refer to bit 11) then: 00 Trigger using swipe threshold reg1. 01 Trigger using swipe threshold reg 2. 10 Trigger using swipe threshold reg 3. 11 Trigger using swipe threshold reg 4.
29-28	GAIN_CTRL4	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
27	RANGE_CHECK_INTR	R/W	0h	0 disable out of range check. 1 - Compare adc data with range check register.
26	FIFO_SELECT	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 0. 1 -> FIFO 1.
25	DIFF_CNTRL	R/W	0h	Differential Control Pin. 0 -> Single Ended. 1 -> Differential Pair Enable.
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins software configuration. 00 - VSSA. 01 - VSSA. 10 - VSSA. 11 ADCREFM.
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins software configuration. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
18-15	SEL_INM_SWM_3_0	R/W	0h	SEL_INM pins for neg differential. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins software configuration. 000 - VDDA. 001 - VDDA. 010 - VDDA. 011 - ADCREFP. 1xx - Reserved.

Table 12-39. ADC1_STEPCONFIG9 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SWIPE_THR_COMPARE_FEATURE	R/W	0h	Set this bit to 1 to enable the swipe data comparing feature: 0: Feature is off (data is always saved). 1: Trigger feature is on if adc data < swipe data threshold map pointer value then discard data otherwise data is saved (and all future steps are saved) until mag_adc enable (bit 0) is turned off.
10-9	GAIN_CTRL3	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
8-7	GAIN_CTRL2	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
6-5	GAIN_CTRL1	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
4-2	AVERAGING	R/W	0h	Number of samplings to average: 000 -> no average. 001 -> 2 samples average. 010 -> 4 samples average. 011 -> 8 samples average. 100 -> 16 samples average.
1-0	MODE	R/W	0h	00 Software enabled, one-shot. 01 Software enabled, continuous. 10 Hardware synchronized, one-shot. 11 Hardware synchronized, continuous.

12.4.1.34 ADC1_STEPDELAY9 Register (offset = A8h) [reset = 0h]

ADC1_STEPDELAY9 is shown in Figure 12-41 and described in Table 12-40.

Figure 12-41. ADC1_STEPDELAY9 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">SAMPLEDELAY</td><td colspan="6">RESERVED</td><td colspan="2">OPENDELAY</td></tr>
<tr><td colspan="8">R/W-0h</td><td colspan="6">R-</td><td colspan="2">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">OPENDELAY</td></tr>
<tr><td colspan="16">R/W-0h</td></tr>
</table>

Table 12-40. ADC1_STEPDELAY9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SAMPLEDELAY	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SoC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	RESERVED	R		
17-0	OPENDELAY	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion.

12.4.1.35 ADC1_STEPCONFIG10 Register (offset = ACh) [reset = 0h]

ADC1_STEPCONFIG10 is shown in Figure 12-42 and described in Table 12-41.

Figure 12-42. ADC1_STEPCONFIG10 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="2">SWIPE_THR_REG_POINTER</td><td colspan="2">GAIN_CTRL4</td><td>RANGE_CHECK_INTR</td><td>FIFO_SELECT</td><td>DIFF_CNTRL</td><td>SEL_RFM_SWC_1_0</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>SEL_RFM_SWC_1_0</td><td colspan="4">SEL_INP_SWC_3_0</td><td colspan="3">SEL_INM_SWM_3_0</td></tr>
<tr><td>R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="3">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>SEL_INM_SWM_3_0</td><td colspan="3">SEL_RFP_SWC_2_0</td><td>SWIPE_THR_COMPARE_FEATURE</td><td colspan="2">GAIN_CTRL3</td><td>GAIN_CTRL2</td></tr>
<tr><td>R/W-0h</td><td colspan="3">R/W-0h</td><td>R/W-0h</td><td colspan="2">R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>GAIN_CTRL2</td><td colspan="2">GAIN_CTRL1</td><td colspan="3">AVERAGING</td><td colspan="2">MODE</td></tr>
<tr><td>R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="3">R/W-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 12-41. ADC1_STEPCONFIG10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SWIPE_THR_REG_POINTER	R/W	0h	If the Swipe Data Compare feature is enabled (refer to bit 11) then: 00 Trigger using swipe threshold reg1. 01 Trigger using swipe threshold reg 2. 10 Trigger using swipe threshold reg 3. 11 Trigger using swipe threshold reg 4.
29-28	GAIN_CTRL4	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
27	RANGE_CHECK_INTR	R/W	0h	0 disable out of range check. 1 - Compare adc data with range check register.
26	FIFO_SELECT	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 0. 1 -> FIFO 1.
25	DIFF_CNTRL	R/W	0h	Differential Control Pin. 0 -> Single Ended. 1 -> Differential Pair Enable.
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins software configuration. 00 - VSSA. 01 - VSSA. 10 - VSSA. 11 ADCREFM.
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins software configuration. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
18-15	SEL_INM_SWM_3_0	R/W	0h	SEL_INM pins for neg differential. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins software configuration. 000 - VDDA. 001 - VDDA. 010 - VDDA. 011 - ADCREFP. 1xx - Reserved.

Table 12-41. ADC1_STEPCONFIG10 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SWIPE_THR_COMPARE_FEATURE	R/W	0h	Set this bit to 1 to enable the swipe data comparing feature: 0: Feature is off (data is always saved). 1: Trigger feature is on if adc data < swipe data threshold map pointer value then discard data otherwise data is saved (and all future steps are saved) until mag_adc enable (bit 0) is turned off.
10-9	GAIN_CTRL3	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
8-7	GAIN_CTRL2	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
6-5	GAIN_CTRL1	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
4-2	AVERAGING	R/W	0h	Number of samplings to average: 000 -> no average. 001 -> 2 samples average. 010 -> 4 samples average. 011 -> 8 samples average. 100 -> 16 samples average.
1-0	MODE	R/W	0h	00 Software enabled, one-shot. 01 Software enabled, continuous. 10 Hardware synchronized, one-shot. 11 Hardware synchronized, continuous.

12.4.1.36 ADC1_STEPDELAY10 Register (offset = B0h) [reset = 0h]

ADC1_STEPDELAY10 is shown in Figure 12-43 and described in Table 12-42.

Figure 12-43. ADC1_STEPDELAY10 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">SAMPLEDELAY</td><td colspan="6">RESERVED</td><td colspan="2">OPENDELAY</td></tr>
<tr><td colspan="8">R/W-0h</td><td colspan="6">R-</td><td colspan="2">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">OPENDELAY</td></tr>
<tr><td colspan="16">R/W-0h</td></tr>
</table>

Table 12-42. ADC1_STEPDELAY10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SAMPLEDELAY	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SoC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	RESERVED	R		
17-0	OPENDELAY	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion.

12.4.1.37 ADC1_STEPCONFIG11 Register (offset = B4h) [reset = 0h]

ADC1_STEPCONFIG11 is shown in Figure 12-44 and described in Table 12-43.

Figure 12-44. ADC1_STEPCONFIG11 Register

31	30	29	28	27	26	25	24
SWIPE_THR_REG_POINTER		GAIN_CTRL4		RANGE_CHECK_INTR	FIFO_SELECT	DIFF_CNTRL	SEL_RFM_SWC_1_0
R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM_3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM_3_0	SEL_RFP_SWC_2_0			SWIPE_THR_COMPARE_FEATURE	GAIN_CTRL3		GAIN_CTRL2
R/W-0h	R/W-0h			R/W-0h	R/W-0h		R/W-0h

7	6	5	4	3	2	1	0
GAIN_CTRL2	GAIN_CTRL1		AVERAGING			MODE	
R/W-0h	R/W-0h		R/W-0h			R/W-0h	

Table 12-43. ADC1_STEPCONFIG11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SWIPE_THR_REG_POINTER	R/W	0h	If the Swipe Data Compare feature is enabled (refer to bit 11) then: 00 Trigger using swipe threshold reg1. 01 Trigger using swipe threshold reg 2. 10 Trigger using swipe threshold reg 3. 11 Trigger using swipe threshold reg 4.
29-28	GAIN_CTRL4	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
27	RANGE_CHECK_INTR	R/W	0h	0 disable out of range check. 1 - Compare adc data with range check register.
26	FIFO_SELECT	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 0. 1 -> FIFO 1.
25	DIFF_CNTRL	R/W	0h	Differential Control Pin. 0 -> Single Ended. 1 -> Differential Pair Enable.
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins software configuration. 00 - VSSA. 01 - VSSA. 10 - VSSA. 11 ADCREFM.
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins software configuration. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
18-15	SEL_INM_SWM_3_0	R/W	0h	SEL_INM pins for neg differential. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins software configuration. 000 - VDDA. 001 - VDDA. 010 - VDDA. 011 - ADCREFP. 1xx - Reserved.

Table 12-43. ADC1_STEPCONFIG11 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SWIPE_THR_COMPARE_FEATURE	R/W	0h	Set this bit to 1 to enable the swipe data comparing feature: 0: Feature is off (data is always saved). 1: Trigger feature is on if adc data < swipe data threshold map pointer value then discard data otherwise data is saved (and all future steps are saved) until mag_adc enable (bit 0) is turned off.
10-9	GAIN_CTRL3	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
8-7	GAIN_CTRL2	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
6-5	GAIN_CTRL1	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
4-2	AVERAGING	R/W	0h	Number of samplings to average: 000 -> no average. 001 -> 2 samples average. 010 -> 4 samples average. 011 -> 8 samples average. 100 -> 16 samples average.
1-0	MODE	R/W	0h	00 Software enabled, one-shot. 01 Software enabled, continuous. 10 Hardware synchronized, one-shot. 11 Hardware synchronized, continuous.

12.4.1.38 ADC1_STEPDELAY11 Register (offset = B8h) [reset = 0h]

ADC1_STEPDELAY11 is shown in Figure 12-45 and described in Table 12-44.

Figure 12-45. ADC1_STEPDELAY11 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAMPLEDELAY								RESERVED						OPENDELAY	
R/W-0h								R-						R/W-0h	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OPENDELAY															
R/W-0h															

Table 12-44. ADC1_STEPDELAY11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SAMPLEDELAY	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SoC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	RESERVED	R		
17-0	OPENDELAY	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion.

12.4.1.39 ADC1_STEPCONFIG12 Register (offset = BCh) [reset = 0h]

ADC1_STEPCONFIG12 is shown in Figure 12-46 and described in Table 12-45.

Figure 12-46. ADC1_STEPCONFIG12 Register

31	30	29	28	27	26	25	24
SWIPE_THR_REG_POINTER		GAIN_CTRL4		RANGE_CHECK_INTR	FIFO_SELECT	DIFF_CNTRL	SEL_RFM_SWC_1_0
R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM_3_0		
R/W-0h	R/W-0h				R/W-0h		
15	14	13	12	11	10	9	8
SEL_INM_SWM_3_0	SEL_RFP_SWC_2_0			SWIPE_THR_COMPARE_FEATURE	GAIN_CTRL3		GAIN_CTRL2
R/W-0h	R/W-0h			R/W-0h	R/W-0h		R/W-0h
7	6	5	4	3	2	1	0
GAIN_CTRL2	GAIN_CTRL1		AVERAGING			MODE	
R/W-0h	R/W-0h		R/W-0h			R/W-0h	

Table 12-45. ADC1_STEPCONFIG12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SWIPE_THR_REG_POINTER	R/W	0h	If the Swipe Data Compare feature is enabled (refer to bit 11) then: 00 Trigger using swipe threshold reg1. 01 Trigger using swipe threshold reg 2. 10 Trigger using swipe threshold reg 3. 11 Trigger using swipe threshold reg 4.
29-28	GAIN_CTRL4	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
27	RANGE_CHECK_INTR	R/W	0h	0 disable out of range check. 1 - Compare adc data with range check register.
26	FIFO_SELECT	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 0. 1 -> FIFO 1.
25	DIFF_CNTRL	R/W	0h	Differential Control Pin. 0 -> Single Ended. 1 -> Differential Pair Enable.
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins software configuration. 00 - VSSA. 01 - VSSA. 10 - VSSA. 11 ADCREFM.
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins software configuration. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
18-15	SEL_INM_SWM_3_0	R/W	0h	SEL_INM pins for neg differential. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins software configuration. 000 - VDDA. 001 - VDDA. 010 - VDDA. 011 - ADCREFP. 1xx - Reserved.

Table 12-45. ADC1_STEPCONFIG12 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SWIPE_THR_COMPARE_FEATURE	R/W	0h	Set this bit to 1 to enable the swipe data comparing feature: 0: Feature is off (data is always saved). 1: Trigger feature is on if adc data < swipe data threshold map pointer value then discard data otherwise data is saved (and all future steps are saved) until mag_adc enable (bit 0) is turned off.
10-9	GAIN_CTRL3	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
8-7	GAIN_CTRL2	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
6-5	GAIN_CTRL1	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
4-2	AVERAGING	R/W	0h	Number of samplings to average: 000 -> no average. 001 -> 2 samples average. 010 -> 4 samples average. 011 -> 8 samples average. 100 -> 16 samples average.
1-0	MODE	R/W	0h	00 Software enabled, one-shot. 01 Software enabled, continuous. 10 Hardware synchronized, one-shot. 11 Hardware synchronized, continuous.

12.4.1.40 ADC1_STEPDELAY12 Register (offset = C0h) [reset = 0h]

ADC1_STEPDELAY12 is shown in Figure 12-47 and described in Table 12-46.

Figure 12-47. ADC1_STEPDELAY12 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">SAMPLEDELAY</td><td colspan="6">RESERVED</td><td colspan="2">OPENDELAY</td></tr>
<tr><td colspan="8">R/W-0h</td><td colspan="6">R-</td><td colspan="2">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">OPENDELAY</td></tr>
<tr><td colspan="16">R/W-0h</td></tr>
</table>

Table 12-46. ADC1_STEPDELAY12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SAMPLEDELAY	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SoC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	RESERVED	R		
17-0	OPENDELAY	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion.

12.4.1.41 ADC1_STEPCONFIG13 Register (offset = C4h) [reset = 0h]

ADC1_STEPCONFIG13 is shown in Figure 12-48 and described in Table 12-47.

Figure 12-48. ADC1_STEPCONFIG13 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="2">SWIPE_THR_REG_POINTER</td><td colspan="2">GAIN_CTRL4</td><td>RANGE_CHECK_INTR</td><td>FIFO_SELECT</td><td>DIFF_CNTRL</td><td>SEL_RFM_SWC_1_0</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>SEL_RFM_SWC_1_0</td><td colspan="4">SEL_INP_SWC_3_0</td><td colspan="3">SEL_INM_SWM_3_0</td></tr>
<tr><td>R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="3">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>SEL_INM_SWM_3_0</td><td colspan="3">SEL_RFP_SWC_2_0</td><td>SWIPE_THR_COMPARE_FEATURE</td><td colspan="2">GAIN_CTRL3</td><td>GAIN_CTRL2</td></tr>
<tr><td>R/W-0h</td><td colspan="3">R/W-0h</td><td>R/W-0h</td><td colspan="2">R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>GAIN_CTRL2</td><td colspan="2">GAIN_CTRL1</td><td colspan="3">AVERAGING</td><td colspan="2">MODE</td></tr>
<tr><td>R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="3">R/W-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 12-47. ADC1_STEPCONFIG13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SWIPE_THR_REG_POINTER	R/W	0h	If the Swipe Data Compare feature is enabled (refer to bit 11) then: 00 Trigger using swipe threshold reg1. 01 Trigger using swipe threshold reg 2. 10 Trigger using swipe threshold reg 3. 11 Trigger using swipe threshold reg 4.
29-28	GAIN_CTRL4	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
27	RANGE_CHECK_INTR	R/W	0h	0 disable out of range check. 1 - Compare adc data with range check register.
26	FIFO_SELECT	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 0. 1 -> FIFO 1.
25	DIFF_CNTRL	R/W	0h	Differential Control Pin. 0 -> Single Ended. 1 -> Differential Pair Enable.
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins software configuration. 00 - VSSA. 01 - VSSA. 10 - VSSA. 11 ADCREFM.
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins software configuration. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
18-15	SEL_INM_SWM_3_0	R/W	0h	SEL_INM pins for neg differential. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins software configuration. 000 - VDDA. 001 - VDDA. 010 - VDDA. 011 - ADCREFP. 1xx - Reserved.

Table 12-47. ADC1_STEPCONFIG13 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SWIPE_THR_COMPARE_FEATURE	R/W	0h	Set this bit to 1 to enable the swipe data comparing feature: 0: Feature is off (data is always saved). 1: Trigger feature is on if adc data < swipe data threshold map pointer value then discard data otherwise data is saved (and all future steps are saved) until mag_adc enable (bit 0) is turned off.
10-9	GAIN_CTRL3	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
8-7	GAIN_CTRL2	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
6-5	GAIN_CTRL1	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
4-2	AVERAGING	R/W	0h	Number of samplings to average: 000 -> no average. 001 -> 2 samples average. 010 -> 4 samples average. 011 -> 8 samples average. 100 -> 16 samples average.
1-0	MODE	R/W	0h	00 Software enabled, one-shot. 01 Software enabled, continuous. 10 Hardware synchronized, one-shot. 11 Hardware synchronized, continuous.

12.4.1.42 ADC1_STEPDELAY13 Register (offset = C8h) [reset = 0h]

ADC1_STEPDELAY13 is shown in Figure 12-49 and described in Table 12-48.

Figure 12-49. ADC1_STEPDELAY13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAMPLEDELAY								RESERVED						OPENDELAY	
R/W-0h								R-						R/W-0h	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPENDELAY															
R/W-0h															

Table 12-48. ADC1_STEPDELAY13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SAMPLEDELAY	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SoC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	RESERVED	R		
17-0	OPENDELAY	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion.

12.4.1.43 ADC1_STEPCONFIG14 Register (offset = CCh) [reset = 0h]

ADC1_STEPCONFIG14 is shown in Figure 12-50 and described in Table 12-49.

Figure 12-50. ADC1_STEPCONFIG14 Register

31	30	29	28	27	26	25	24
SWIPE_THR_REG_POINTER		GAIN_CTRL4		RANGE_CHECK_INTR	FIFO_SELECT	DIFF_CNTRL	SEL_RFM_SWC_1_0
R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM_3_0		
R/W-0h	R/W-0h				R/W-0h		
15	14	13	12	11	10	9	8
SEL_INM_SWM_3_0	SEL_RFP_SWC_2_0			SWIPE_THR_COMPARE_FEATURE	GAIN_CTRL3		GAIN_CTRL2
R/W-0h	R/W-0h			R/W-0h	R/W-0h		R/W-0h
7	6	5	4	3	2	1	0
GAIN_CTRL2	GAIN_CTRL1		AVERAGING			MODE	
R/W-0h	R/W-0h		R/W-0h			R/W-0h	

Table 12-49. ADC1_STEPCONFIG14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SWIPE_THR_REG_POINTER	R/W	0h	If the Swipe Data Compare feature is enabled (refer to bit 11) then: 00 Trigger using swipe threshold reg1. 01 Trigger using swipe threshold reg 2. 10 Trigger using swipe threshold reg 3. 11 Trigger using swipe threshold reg 4.
29-28	GAIN_CTRL4	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
27	RANGE_CHECK_INTR	R/W	0h	0 disable out of range check. 1 - Compare adc data with range check register.
26	FIFO_SELECT	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 0. 1 -> FIFO 1.
25	DIFF_CNTRL	R/W	0h	Differential Control Pin. 0 -> Single Ended. 1 -> Differential Pair Enable.
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins software configuration. 00 - VSSA. 01 - VSSA. 10 - VSSA. 11 ADCREFM.
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins software configuration. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
18-15	SEL_INM_SWM_3_0	R/W	0h	SEL_INM pins for neg differential. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins software configuration. 000 - VDDA. 001 - VDDA. 010 - VDDA. 011 - ADCREFP. 1xx - Reserved.

Table 12-49. ADC1_STEPCONFIG14 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SWIPE_THR_COMPARE_FEATURE	R/W	0h	Set this bit to 1 to enable the swipe data comparing feature: 0: Feature is off (data is always saved). 1: Trigger feature is on if adc data < swipe data threshold map pointer value then discard data otherwise data is saved (and all future steps are saved) until mag_adc enable (bit 0) is turned off.
10-9	GAIN_CTRL3	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
8-7	GAIN_CTRL2	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
6-5	GAIN_CTRL1	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
4-2	AVERAGING	R/W	0h	Number of samplings to average: 000 -> no average. 001 -> 2 samples average. 010 -> 4 samples average. 011 -> 8 samples average. 100 -> 16 samples average.
1-0	MODE	R/W	0h	00 Software enabled, one-shot. 01 Software enabled, continuous. 10 Hardware synchronized, one-shot. 11 Hardware synchronized, continuous.

12.4.1.44 ADC1_STEPDELAY14 Register (offset = D0h) [reset = 0h]

ADC1_STEPDELAY14 is shown in Figure 12-51 and described in Table 12-50.

Figure 12-51. ADC1_STEPDELAY14 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">SAMPLEDELAY</td><td colspan="6">RESERVED</td><td colspan="2">OPENDELAY</td></tr>
<tr><td colspan="8">R/W-0h</td><td colspan="6">R-</td><td colspan="2">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">OPENDELAY</td></tr>
<tr><td colspan="16">R/W-0h</td></tr>
</table>

Table 12-50. ADC1_STEPDELAY14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SAMPLEDELAY	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SoC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	RESERVED	R		
17-0	OPENDELAY	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion.

12.4.1.45 ADC1_STEPCONFIG15 Register (offset = D4h) [reset = 0h]

ADC1_STEPCONFIG15 is shown in Figure 12-52 and described in Table 12-51.

Figure 12-52. ADC1_STEPCONFIG15 Register

31	30	29	28	27	26	25	24
SWIPE_THR_REG_POINTER		GAIN_CTRL4		RANGE_CHECK_INTR	FIFO_SELECT	DIFF_CNTRL	SEL_RFM_SWC_1_0
R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM_3_0		
R/W-0h	R/W-0h				R/W-0h		
15	14	13	12	11	10	9	8
SEL_INM_SWM_3_0	SEL_RFP_SWC_2_0			SWIPE_THR_COMPARE_FEATURE	GAIN_CTRL3		GAIN_CTRL2
R/W-0h	R/W-0h			R/W-0h	R/W-0h		R/W-0h
7	6	5	4	3	2	1	0
GAIN_CTRL2	GAIN_CTRL1		AVERAGING			MODE	
R/W-0h	R/W-0h		R/W-0h			R/W-0h	

Table 12-51. ADC1_STEPCONFIG15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SWIPE_THR_REG_POINTER	R/W	0h	If the Swipe Data Compare feature is enabled (refer to bit 11) then: 00 Trigger using swipe threshold reg1. 01 Trigger using swipe threshold reg 2. 10 Trigger using swipe threshold reg 3. 11 Trigger using swipe threshold reg 4.
29-28	GAIN_CTRL4	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
27	RANGE_CHECK_INTR	R/W	0h	0 disable out of range check. 1 - Compare adc data with range check register.
26	FIFO_SELECT	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 0. 1 -> FIFO 1.
25	DIFF_CNTRL	R/W	0h	Differential Control Pin. 0 -> Single Ended. 1 -> Differential Pair Enable.
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins software configuration. 00 - VSSA. 01 - VSSA. 10 - VSSA. 11 ADCREFM.
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins software configuration. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
18-15	SEL_INM_SWM_3_0	R/W	0h	SEL_INM pins for neg differential. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins software configuration. 000 - VDDA. 001 - VDDA. 010 - VDDA. 011 - ADCREFP. 1xx - Reserved.

Table 12-51. ADC1_STEPCONFIG15 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SWIPE_THR_COMPARE_FEATURE	R/W	0h	Set this bit to 1 to enable the swipe data comparing feature: 0: Feature is off (data is always saved). 1: Trigger feature is on if adc data < swipe data threshold map pointer value then discard data otherwise data is saved (and all future steps are saved) until mag_adc enable (bit 0) is turned off.
10-9	GAIN_CTRL3	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
8-7	GAIN_CTRL2	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
6-5	GAIN_CTRL1	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
4-2	AVERAGING	R/W	0h	Number of samplings to average: 000 -> no average. 001 -> 2 samples average. 010 -> 4 samples average. 011 -> 8 samples average. 100 -> 16 samples average.
1-0	MODE	R/W	0h	00 Software enabled, one-shot. 01 Software enabled, continuous. 10 Hardware synchronized, one-shot. 11 Hardware synchronized, continuous.

12.4.1.46 ADC1_STEPDELAY15 Register (offset = D8h) [reset = 0h]

ADC1_STEPDELAY15 is shown in Figure 12-53 and described in Table 12-52.

Figure 12-53. ADC1_STEPDELAY15 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">SAMPLEDELAY</td><td colspan="6">RESERVED</td><td colspan="2">OPENDELAY</td></tr>
<tr><td colspan="8">R/W-0h</td><td colspan="6">R-</td><td colspan="2">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">OPENDELAY</td></tr>
<tr><td colspan="16">R/W-0h</td></tr>
</table>

Table 12-52. ADC1_STEPDELAY15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SAMPLEDELAY	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SoC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	RESERVED	R		
17-0	OPENDELAY	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion.

12.4.1.47 ADC1_STEPCONFIG16 Register (offset = DCh) [reset = 0h]

ADC1_STEPCONFIG16 is shown in Figure 12-54 and described in Table 12-53.

Figure 12-54. ADC1_STEPCONFIG16 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="2">SWIPE_THR_REG_POINTER</td><td colspan="2">GAIN_CTRL4</td><td>RANGE_CHECK_INTR</td><td>FIFO_SELECT</td><td>DIFF_CNTRL</td><td>SEL_RFM_SWC_1_0</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>SEL_RFM_SWC_1_0</td><td colspan="4">SEL_INP_SWC_3_0</td><td colspan="3">SEL_INM_SWM_3_0</td></tr>
<tr><td>R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="3">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>SEL_INM_SWM_3_0</td><td colspan="3">SEL_RFP_SWC_2_0</td><td>SWIPE_THR_COMPARE_FEATURE</td><td colspan="2">GAIN_CTRL3</td><td>GAIN_CTRL2</td></tr>
<tr><td>R/W-0h</td><td colspan="3">R/W-0h</td><td>R/W-0h</td><td colspan="2">R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>GAIN_CTRL2</td><td colspan="2">GAIN_CTRL1</td><td colspan="3">AVERAGING</td><td colspan="2">MODE</td></tr>
<tr><td>R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="3">R/W-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 12-53. ADC1_STEPCONFIG16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SWIPE_THR_REG_POINTER	R/W	0h	If the Swipe Data Compare feature is enabled (refer to bit 11) then: 00 Trigger using swipe threshold reg1. 01 Trigger using swipe threshold reg 2. 10 Trigger using swipe threshold reg 3. 11 Trigger using swipe threshold reg 4.
29-28	GAIN_CTRL4	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
27	RANGE_CHECK_INTR	R/W	0h	0 disable out of range check. 1 - Compare adc data with range check register.
26	FIFO_SELECT	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 0. 1 -> FIFO 1.
25	DIFF_CNTRL	R/W	0h	Differential Control Pin. 0 -> Single Ended. 1 -> Differential Pair Enable.
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins software configuration. 00 - VSSA. 01 - VSSA. 10 - VSSA. 11 ADCREFM.
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins software configuration. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
18-15	SEL_INM_SWM_3_0	R/W	0h	SEL_INM pins for neg differential. 0000 -> Channel 1. 0111 -> Channel 8. 1xxx -> ADCREFM.
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins software configuration. 000 - VDDA. 001 - VDDA. 010 - VDDA. 011 - ADCREFP. 1xx - Reserved.

Table 12-53. ADC1_STEPCONFIG16 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SWIPE_THR_COMPARE_FEATURE	R/W	0h	Set this bit to 1 to enable the swipe data comparing feature: 0: Feature is off (data is always saved). 1: Trigger feature is on if adc data < swipe data threshold map pointer value then discard data otherwise data is saved (and all future steps are saved) until mag_adc enable (bit 0) is turned off.
10-9	GAIN_CTRL3	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
8-7	GAIN_CTRL2	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
6-5	GAIN_CTRL1	R/W	0h	00: Gain of 12. 01: Gain of 14. 10: Gain of 16. 11: Gain of 18.
4-2	AVERAGING	R/W	0h	Number of samplings to average: 000 -> no average. 001 -> 2 samples average. 010 -> 4 samples average. 011 -> 8 samples average. 100 -> 16 samples average.
1-0	MODE	R/W	0h	00 Software enabled, one-shot. 01 Software enabled, continuous. 10 Hardware synchronized, one-shot. 11 Hardware synchronized, continuous.

12.4.1.48 ADC1_STEPDELAY16 Register (offset = E0h) [reset = 0h]

ADC1_STEPDELAY16 is shown in Figure 12-55 and described in Table 12-54.

Figure 12-55. ADC1_STEPDELAY16 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAMPLEDELAY								RESERVED						OPENDELAY	
R/W-0h								R-						R/W-0h	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPENDELAY															
R/W-0h															

Table 12-54. ADC1_STEPDELAY16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SAMPLEDELAY	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SoC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	RESERVED	R		
17-0	OPENDELAY	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion.

12.4.1.49 ADC1_FIFO0COUNT Register (offset = E4h) [reset = 0h]

ADC1_FIFO0COUNT is shown in Figure 12-56 and described in Table 12-55.

Figure 12-56. ADC1_FIFO0COUNT Register

Table 12-55. ADC1_FIFO0COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
6-0	WORDS_IN_FIFO0	R	0h	Number of words currently in the FIFO0.

12.4.1.50 ADC1_FIFO0THR Register (offset = E8h) [reset = 0h]

ADC1_FIFO0THR is shown in Figure 12-57 and described in Table 12-56.

Figure 12-57. ADC1_FIFO0THR Register

Table 12-56. ADC1_FIFO0THR Register Field Descriptions

Bit	Field	Type	Reset	Description
5-0	FIFO0_THR_LEVEL	R/W	0h	Program the desired FIFO0 data sample level to reach before generating interrupt to CPU (program to value minus 1).

12.4.1.51 ADC1_DMA0REQ Register (offset = ECh) [reset = 0h]

ADC1_DMA0REQ is shown in Figure 12-58 and described in Table 12-57.

Figure 12-58. ADC1_DMA0REQ Register

Table 12-57. ADC1_DMA0REQ Register Field Descriptions

Bit	Field	Type	Reset	Description
5-0	DMA_REQUEST_LEVEL	R/W	0h	Number of words in FIFO0 before generating a DMA request (program to value minus 1).

12.4.1.52 ADC1_FIFO1COUNT Register (offset = F0h) [reset = 0h]

ADC1_FIFO1COUNT is shown in Figure 12-59 and described in Table 12-58.

Figure 12-59. ADC1_FIFO1COUNT Register

Table 12-58. ADC1_FIFO1COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
6-0	WORDS_IN_FIFO1	R	0h	Number of words currently in the FIFO1.

12.4.1.53 ADC1_FIFO1THR Register (offset = F4h) [reset = 0h]

ADC1_FIFO1THR is shown in Figure 12-60 and described in Table 12-59.

Figure 12-60. ADC1_FIFO1THR Register

Table 12-59. ADC1_FIFO1THR Register Field Descriptions

Bit	Field	Type	Reset	Description
5-0	FIFO1_THR_LEVEL	R/W	0h	Program the desired FIFO1 data sample level to reach before generating interrupt to CPU (program to value minus 1).

12.4.1.54 ADC1_DMA1REQ Register (offset = F8h) [reset = 0h]

ADC1_DMA1REQ is shown in Figure 12-61 and described in Table 12-60.

Figure 12-61. ADC1_DMA1REQ Register

Table 12-60. ADC1_DMA1REQ Register Field Descriptions

Bit	Field	Type	Reset	Description
5-0	DMA_REQUEST_LEVEL	R/W	0h	Number of words in FIFO1 before generating a DMA request (program to value minus 1).

12.4.1.55 ADC1_FIFO0DATA Register (offset = 100h) [reset = 0h]

ADC1_FIFO0DATA is shown in Figure 12-62 and described in Table 12-61.

Figure 12-62. ADC1_FIFO0DATA Register

Table 12-61. ADC1_FIFO0DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
19-16	ADCCHNLID	R	0h	Optional ID tag of channel that captured the data. If tag option is disabled, these bits will be 0.
15-12	RESERVED	R		
11-0	ADCDATA	R	0h	12 bit sampled ADC converted data value stored in FIFO0.

12.4.1.56 ADC1_FIFO1DATA Register (offset = 200h) [reset = 0h]

ADC1_FIFO1DATA is shown in Figure 12-63 and described in Table 12-62.

Figure 12-63. ADC1_FIFO1DATA Register

Table 12-62. ADC1_FIFO1DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
19-16	ADCCHNLID	R	0h	Optional ID tag of channel that captured the data. If tag option is disabled, these bits will be 0.
15-12	RESERVED	R		
11-0	ADCDATA	R	0h	12 bit sampled ADC converted data value stored in FIFO1.

Display Subsystem (DSS)

This chapter describes the display subsystem (DSS) of the device.

NOTE: The Display Subsystem (DSS) is not supported for the AMIC120.

Topic **Page**

13.1 Introduction

13.1.1 DSS Features

The general features of the DSS module include:

- Display Controller
  - Display Modes
    - Programmable pixel memory formats (Palletized: 1, 2, 4, 8-bit per pixel; RGB 16, and 24-bit per pixel; and YUV 4:2:2)
    - Programmable display size (up to 2048 x 2048)
    - 256 x 24-bit entries palette in RGB
    - Programmable pixel rate (up to 100 MHz)
  - Display Support
    - Four types of displays are supported: Passive and Active colors, Passive and Active monochromes.
    - 4-/8-bit Monochrome Passive panel interface support (15 grayscale levels supported using dithering block)
    - RGB 8-bit Color Passive panel interface support (3,375 colors supported for color panel using dithering block).
    - RGB 12/16/18/24-bit Active panel interface support (replicated or dithered encoded pixel values).
    - Remote Frame Buffer (embedded in the LCD panel) support through the RFBI module
    - Partial refresh of the remote frame buffer through the RFBI module
    - Partial display
    - Multiple cycles output format on 8/9/12/16bit interface (TDM)
  - Signal Processing
    - Overlay and Windowing support for one Graphics layer (RGB or CLUT) and two Video layers (YUV 4:2:2, RGB16 and RGB24)
    - RGB 24-bit support on the display interface, optionally dithered to RGB 18-bit pixel output + 6-Bit Frame rate Control (spatial and temporal)
    - Transparency color key (source and destination)
    - Synchronized buffer update
    - Gamma Curve Support
    - Multiple-buffer support
    - Cropping Support
    - Merge capability of the DMA FIFO for use by a single pipeline in case of DVFS
    - Color Phase rotation
  - Interrupt line and DMA line trigger signals
    - The LCD DMA is a secure transaction initiator on the L3 interconnect.
- Remote Frame Buffer Interface
  - Access to RFB's “direct MPU interface”
    - Sending commands to the RFB panel.
    - Sending data to the RFB panel, received from the Display controller or from the MPU (through the L4 OCP slave port)
    - Reading data/status from the RFB to the OCP slave port.
  - RFB interface
    - 8/9/12/16-bit parallel interface (up to QVGA@30fps at nominal voltage)
    - Two programmable configurations for two devices connected to the RFBI module.

    - Tearing Effect control logic (Horizontal Synchronization (HSync) and Vertical Synchronization (VSync) embedded in a single signal (TE) or using two signals (HS+VS)).
  - Data formats
    - Programmable pixel memory formats (12-, 16-, 18- and 24-bit-per-pixel modes in RGB format)
    - Programmable output formats on one/multiple cycles per pixel (data from Display controller and from L4) (TDM)
-

13.1.2 Unsupported Features

The display subsystem does not support the following features.

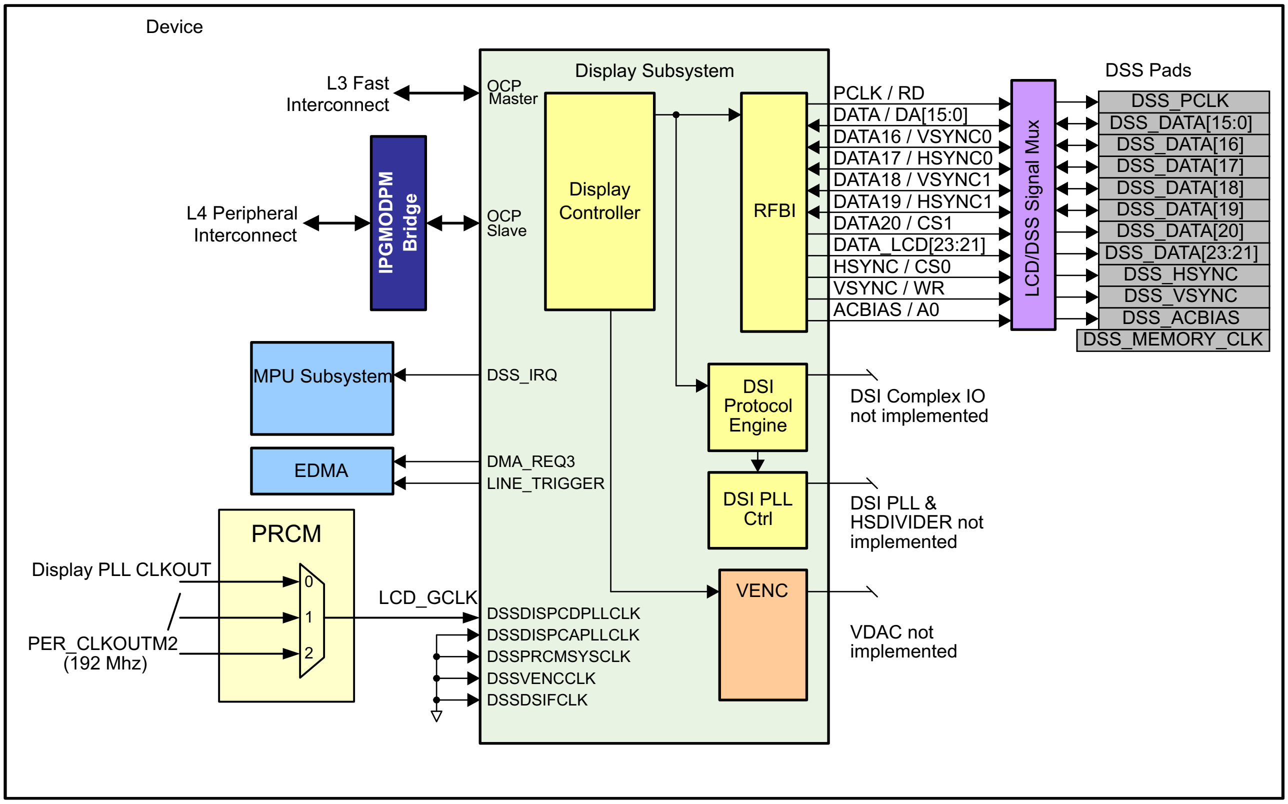
Table 13-1. Unsupported Display Subsystem Features

Feature	Reason
FlatLink Serial Display	SDI Complex I/Os not implemented
MIPI Serial Displays	DSI Complex I/Os not implemented
NTSC/PAL Displays	Video DAC not implemented
Rotation/Mirroring	No SDRAM rotation engine implemented on device

13.2 Integration

This device includes a Display Subsystem (DSS) that reads display data from external memory and drives several different types of LCD displays. The Display Subsystem integration is shown in Figure 13-1.

Figure 13-1. DSS Integration



13.2.1 DSS Connectivity Attributes

The general connectivity attributes for the DSS module are shown in Table 13-2.

Table 13-2. DSS Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L3_GCLK (OCP master clock) PD_PER_L4LS_GCLK (OCP slave clock) PD_PER_LCD_GCLK (Functional clock)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Standby Smart Idle
Interrupt Requests	1 interrupt to MPU Subsystem (DSSINT)
DMA Requests	2 DMA requests to EDMA
Physical Address	L4 peripheral slave port

13.2.2 DSS Clock and Reset Management

The DSS module uses the OCP interface and multiple functional clocks. The L4 slave interface runs at half the frequency of the L3 Master interface. The functional clocks come from the device's display PLL rather than a dedicated DSI_PLL. When the display PLL is in bypass mode, its output is sourced by either CORE_CLKOUTM6 or PER_CLKOUTM2.

Table 13-3. DSS Clock Signals

Clock Signal	Maximum Frequency	Reference Source	Comments
DSSL3OCPIFCLK Master interface	200 MHz	CORE_CLKOUTM4	pd_per_l3_gclk from PRCM
DSSL3OCPIFENAB L3 phase clock enable used to create L4 clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk from PRCM
DSSPRCMSYSCLK Functional clock	N/A	N/A	DSI PLL source clock not needed since DSI PLL is not present
DSSDISPCAPLLCLK L3 phase clock enable used to create L4 clock	N/A	N/A	DSSDISPCDPLLCLK is default display controller clock source
DSSDISPCDPLLCLK(1)	200 MHz	Display PLL CLKOUT or PER_CLKOUTM2	pd_per_lcd_gclk from PRCM
DSSVENCCLK Video encoder functional clock	N/A	N/A	VENC not supported
DSSDSIFCLK DSI functional clock	N/A	N/A	DSI not supported

(1) The Display PLL CLKOUT can be used as a reference clock for other peripherals. If these other peripherals require the Display PLL CLKOUT to be configured for >200MHz, then the DSS must use the PER_CLKOUTM2 (192 MHz) and the pixel clock will be limited to a divisor of 192 MHz.

13.2.3 DSS Pin List

The display subsystem external interface pins are shown in Table 13-4. Pin function depends on the operating mode (bypass or RFBI). The DSSLCDDIR[2:0] control signals are used to control the pad OEN of the associated signals as shown.

Table 13-4. DSS Pin List

Pin Name	Pin Function	Type	Description	DSSLCDDIR Control Signal
dss_pclk	DISPC_PCLK / RFBI_RD	O	Pixel Clock / RFBI Read Enable	1
dss_data[15:0]	DISPC_DATA_LCD / RFBI_DA[15:0]	I/O/Z	Pixel Data Bus / RFBI Data	0
dss_data16	DISPC_DATA_LCD16 / RFBI_TE_VSYNC0	I/O/Z	Pixel Data Bus / RFBI Tearing Effect or Vertical Sync 0	2
dss_data17	DISPC_DATA_LCD17 / RFBI_HSYNC0	I/O/Z	Pixel Data Bus / RFBI Horizontal Sync 0	2
dss_data18	DISPC_DATA_LCD18 / RFBI_TE_VSYNC1	I/O/Z	Pixel Data Bus / RFBI Tearing Effect Vertical Sync 1	2
dss_data19	DISPC_DATA_LCD19 / RFBI_HSYNC1	I/O/Z	Pixel Data Bus / RFBI Horizontal Sync 1	2
dss_data20	DISPC_DATA_LCD20 / RFBI_CS1	O	Pixel Data Bus / RFBI Chip Select 0	1
dss_data[23:21]	DISPC_DATA_LCD[23:21]	O	Pixel Data Bus	1
dss_hsync	DISPC_HSYNC / RFBI_CS0	O	Horizontal Sync / RFBI Chip Select 0	1
dss_vsync	DISPC_VSYNC / RFBI_WR	O	Vertical Sync / RFBI Write Enable	1
dss_ac_bias_en	DISPC_ACBIAS / RFBI_A0	O	AC Bias Enable / RFBI Command/Data indicator	1

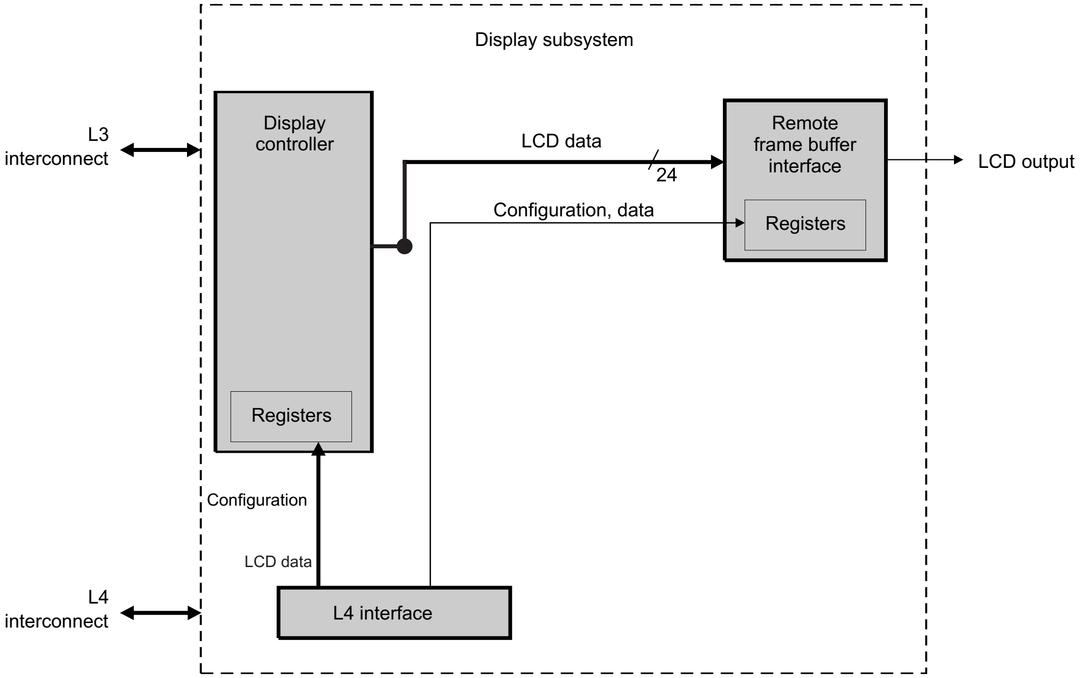
13.3 Functional Description

This section describes the functions of the LCD display support by describing the display controller and RFBI modules.

13.3.1 Block Diagram

Figure 13-2 is a schematic of the display subsystem.

Figure 13-2. Display Subsystem Full Schematic



13.3.2 Display Subsystem Environment

This section describes the two main functions handled by the display subsystem:

- LCD support
- TV display support

13.3.2.1 LCD Support

LCD panels can be connected to the display subsystem of the device using parallel and/or serial interfaces.

Table 13-5 provides more details on the supported interfaces to LCD panels, and the respective pad and signal configurations.

Table 13-5. LCD Interface Signals and Configurations

Pads and Signals Configuration					
Pad names at device level boundary.	**Basic signal multiplexing on pads**		**Additional signal multiplexing on pads**		**Sequential**
	Parallel Interface, Bypass Mode.	**Parallel Interface, RFBI mode.**	**Parallel Interface, Bypass Mode.**	**Parallel Interface, RFBI mode.**	**"1. Parallel Interface, RFBI-0 mode 16-bit 2. Parallel Interface, RFBI-1 mode 16-bit"**
dss_hsync	DISPC_HSYNC	RFBI_CS0			RFBI (CS0)
dss_vsync	DISPC_VSYNC	RFBI_WR			RFBI
dss_pclk	DISPC_PCLK	RFBI_RD			
dss_acbias	DISPC_ACBIAS	RFBI_A0			
dss_data0	DISPC_DATA_LCD0	RFBI_DA0			RFBI_DA [0-5]
dss_data1	DISPC_DATA_LCD1	RFBI_DA1			
dss_data2	DISPC_DATA_LCD2	RFBI_DA2			
dss_data3	DISPC_DATA_LCD3	RFBI_DA3			
dss_data4	DISPC_DATA_LCD4	RFBI_DA4			
dss_data5	DISPC_DATA_LCD5	RFBI_DA5			
dss_data6	DISPC_DATA_LCD6	RFBI_DA6			RFBI_DA [6-15]
dss_data7	DISPC_DATA_LCD7	RFBI_DA7			
dss_data8	DISPC_DATA_LCD8	RFBI_DA8			
dss_data9	DISPC_DATA_LCD9	RFBI_DA9			
dss_data10	DISPC_DATA_LCD10	RFBI_DA10			
dss_data11	DISPC_DATA_LCD11	RFBI_DA11			
dss_data12	DISPC_DATA_LCD12	RFBI_DA12			
dss_data13	DISPC_DATA_LCD13	RFBI_DA13			
dss_data14	DISPC_DATA_LCD14	RFBI_DA14			RFBI_DA [6-15]
dss_data15	DISPC_DATA_LCD15	RFBI_DA15			
gpmc_ad15	DISPC_DATA_LCD16	RFBI_TE_VSYNC0			"RFBI(sync0)"
gpmc_ad14	DISPC_DATA_LCD17	RFBI_HSYNC0			
gpmc_ad13	DISPC_DATA_LCD18	RFBI_TE_VSYNC1			"RFBI(sync1)"
gpmc_ad12	DISPC_DATA_LCD19	RFBI_HSYNC1			
gpmc_ad11	DISPC_DATA_LCD20	RFBI_CS1			RFBI (CS1)
gpmc_ad10	DISPC_DATA_LCD21				
gpmc_ad9	DISPC_DATA_LCD22				
gpmc_ad8	DISPC_DATA_LCD23				
cam1_data9			DISPC_DATA_LCD16		
cam0_data9			DISPC_DATA_LCD17		

Table 13-5. LCD Interface Signals and Configurations (continued)

Pads and Signals Configuration					
Pad names at device level boundary.	**Basic signal multiplexing on pads**		**Additional signal multiplexing on pads**		**Sequential**
	Parallel Interface, Bypass Mode.	**Parallel Interface, RFBI mode.**	**Parallel Interface, Bypass Mode.**	**Parallel Interface, RFBI mode.**	**"1. Parallel Interface, RFBI-0 mode 16-bit 2. Parallel Interface, RFBI-1 mode 16-bit"**
cam0_data8			DISPC_DATA_LCD18		
cam0_pclk			DISPC_DATA_LCD19		
cam0_wen			DISPC_DATA_LCD20		
cam0_field			DISPC_DATA_LCD21		
cam0_vd			DISPC_DATA_LCD22		
cam0_hd			DISPC_DATA_LCD23		

For more information on signals multiplexing, see the device-specific datasheet.

The parallel interface connectivity is detailed in Section 13.3.2.1.1, *Parallel Interface*.

13.3.2.1.1 Parallel Interface

In parallel interface, the paths of the display subsystem modules are the display controller and the RFBI.

The display controller provides the required control signals to interface the memory frame buffer (SDRAM or SRAM) directly to the external displays. The display controller is connected to the memory through the L3 interconnect and has its own DMA (with embedded FIFOs) to read data from the system memory. The L3 interconnect is the master port, while the L4 interconnect is the slave port of the display subsystem.

The display controller has two I/O pad modes at the module level:

- RFBI mode (RFBI enabled), which implements the MIPI DBI 2.0 protocol
- Bypass mode (RFBI disabled), which implements the MIPI DPI 1.0 protocol

The DSS.DISPC_CTRL[16:15] GPOUT[1:0] bits control selection of the display subsystem modules (see Table 13-6).

Table 13-6. I/O Pad Mode Selection

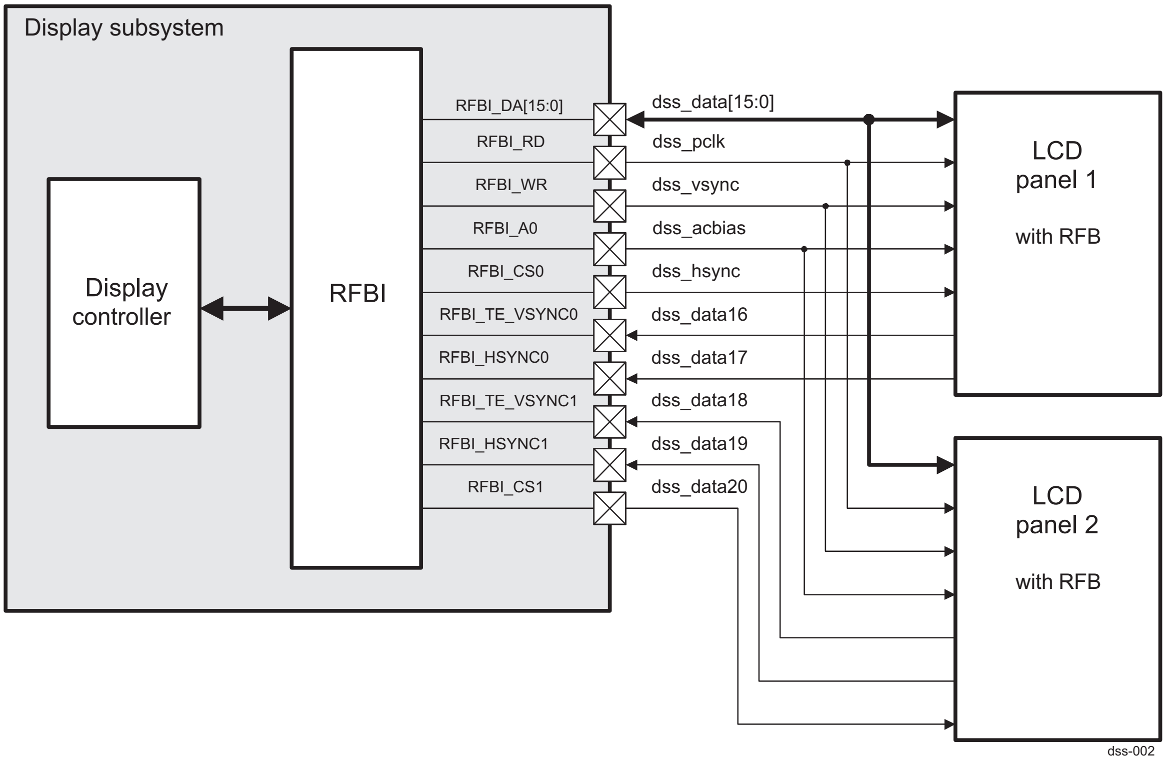
DSS.DISPC_CTRL[16] GPOUT1	DSS.DISPC_CTRL[15] GPOUT0	Mode
0	0	Reset
0	1	RFBI mode
1	0	Invalid
1	1	Bypass mode

The RFB of the LCD panel is connected directly to the RFBI module of the device. The RFBI controls the reads/writes from/to the RFB. The RFBI receives the output from the DISPC (which takes data from the memory) and generates the signals to control the LCD panel. Through the RFBI, the MPU can send commands or parameter/display data to the LCD panel and directly set the DISPC registers to read/write the data from/to the memory in the LCD panel. The RFBI can manage up to two LCD panels when the serial interface is not used.

13.3.2.1.1.1 Parallel Interface in RFBI Mode (MIPI DBI Protocol)

Figure 13-3 shows the LCD support parallel interface in RFBI mode (example for 16-bit data interface).

Figure 13-3. LCD Support Parallel Interface (RFBI Mode)



NOTE: Configure the DSS.RFBI_CTRL[3:2] CONFIG_SELECT bit field to drive signals for LCD 1 only, LCD 2 only, or both LCD 1 and LCD 2.

Table 13-7 describes the interface signals to/from the LCD panel in RFBI mode.

Table 13-7. LCD Interface Signals (RFBI Mode)

Signal Name	Type[1]	Description
RFBI_DA[15:0]	I/O	RFBI I/O data
RFBI_RD	O	Read access signal
RFBI_WR	O	Write access signal
RFBI_A0	O	Command/data selection signal
RFBI_CS0	O	Chip-select (CS) signal for LCD 1
RFBI_CS1	O	CS signal for LCD 2
RFBI_TE_VSYNC0	I	Tearing effect (TE) synchronization signal (TE or VSYNC for LCD panel 1)
RFBI_HSYNC0	I	HSYNC from LCD panel 1
RFBI_TE_VSYNC1	I	TE synchronization signal (TE or VSYNC for LCD panel 2)
RFBI_HSYNC1	I	HSYNC from LCD panel 2

[1] I = Input, O = Output, I/O = Input/Output

- RFBI_DA[15:0]: The pixel data comprises the RFBI pixel data (bits 15:0). A write/read command must be sent to the LCD panel to send/read the data.

  Before any data access, the application must send commands and parameters when it is necessary to configure an LCD panel. The data is used as input in read operations during production test and also to read the status of the registers in the LCD panel and pixels from the embedded frame buffer in the

LCD panel module. RFBI_DA is multiplexed at the chip-level boundary with dss_data [15:0].

- RFBI_RD: This is the read-enable signal used to indicate when a read from the embedded memory in the LCD panel is ongoing. The RFBI registers describe the behavior of the read signal (off/on/cycle time). The polarity of the read-enable signal is programmable. This signal is multiplexed at the chip-level boundary with dss_pclk. The read is used to get status/data information from the LCD panel.
- RFBI_WR: The write-enable signal is used to indicate when a write is ongoing. The RFBI registers describe the behavior of the write signal (off/on/cycle time). The polarity of the write-enable signal is programmable. This signal is multiplexed at the chip-level boundary with dss_vsync.
- RFBI_A0: The signal is asserted to indicate its status: Command or data. The polarity is programmable and the status of the signal depends on the RFBI registers written by the application (CMD/READ/STATUS/PARAM/PIXEL). The register in use by the hardware defines the status of RFBI_A0. The order of the writes/reads to the RFBI registers CMD/READ/STATUS/PARAM/PIXEL defines the transitions of A0. This signal is multiplexed at the chip-level boundary with dss_acbias.
- RFBI_CSx: The signal is the chip-select (CSx) asserted to indicate which LCD panel is selected and must be ready to receive/transmit commands and data. When RE or WE is on, CSx must not be changed (x = 0 for LCD panel 1; x = 1 for LCD panel 2). CS0 is multiplexed at the chip-level boundary with dss_hsync, and CS1 is multiplexed at the chip-level boundary with dss_data[20].
- RFBI_TE_VSYNCx: Based on the trigger mode selected, the signal is the TE pulse signal or the LCD panel VSYNC (vertical synchronization) pulse signal. RFBI_TE_VSYNCx is used by the TE logic as the synchronization signal to send the pixel to the LCD panel.

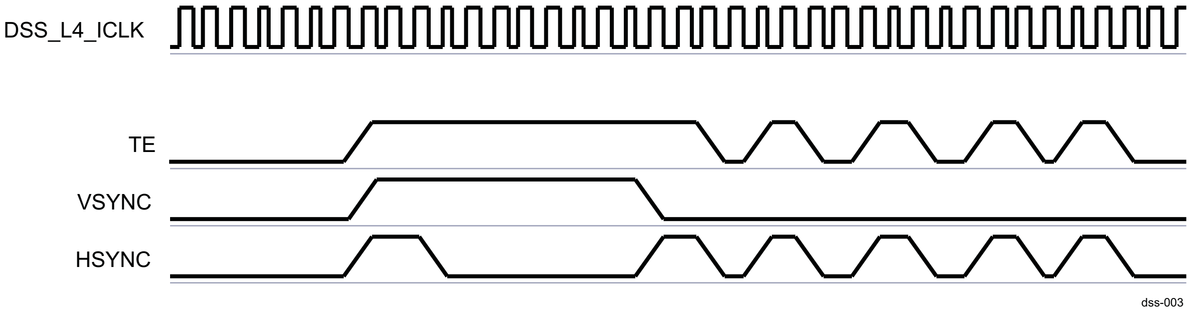
  To select the trigger mode, configure the DSS.RFBI_CONFIG_i[3:2] TRIGGER_MODE bit field (0x0: Internal trigger mode with the DSS.RFBI_CTRL[4] ITE bit, 0x1: External trigger mode with the TE signal RFBI_TE_VSYNCx, 0x2: External trigger mode with the RFBI_TE_VSYNCx, and RFBI_HSYNCx signals with the programmable line counter).

  These signals are multiplexed at the chip-level boundary with dss_data[16] (RFBI_TE_VSYNC0) and dss_data[18] (RFBI_TE_VSYNC1) (LCD panel 1: x = 0; LCD panel 2: x = 1).
- RFBI_HSYNCx: The HSYNC pulse signals indicate to the RFBI module when horizontal synchronization occurs. The polarity of the HSYNC signals is programmable. The minimum pulse width of the signal is two L4 cycles. RFBI_HSYNC is used by the TE logic as a synchronization signal to send the pixel to the LCD panel. These signals are multiplexed at the chip-level boundary with dss_data[17] (RFBI_HSYNC0) and dss_data[19] (RFBI_HSYNC1) (LCD panel 1: x = 0; LCD panel 2: x = 1).

13.3.2.1.1.1.1 Description of the TE Pulse Signal

The externally-generated TE synchronization signal is a logical OR or AND operation between the HSYNC and VSYNC signals (see Figure 13-4). The logical operation (OR or AND) depends on the HSYNC and VSYNC signals polarity. The VSYNC signal indicates to the RFBI module when vertical synchronization occurs; the HSYNC signal indicates to the RFBI module when horizontal synchronization occurs.

Figure 13-4. External Generation of TE Signal Based on Logical OR Operation Between HSYNC and VSYNC (Active-High)



The RFBI module detects the VSYNC and HSYNC pulses embedded in the received signal. VSYNC is detected based on the minimum pulse width defined by the DSS.RFBI_VSYNC_WIDTH register.

HSYNC is detected based on the minimum pulse width defined by the DSS.RFBI_HSYNC_WIDTH register.

The signal is generated from external logic based on the VSYNC/HSYNC of the LCD panel. The automatic trigger can be programmed based on the RFBI_TE signal or use a bit field in the RFBI registers to start data capture.

The polarity of the TE signal is programmable. The HSYNC and VSYNC pulses embedded in the TE signal have the same polarity, which is active high for an ORed signal and active low for an ANDed signal. The minimum pulse width of the signal is two L4 cycles. Hardware resets the line counter when the VSYNC occurs and increments it at every HSYNC. Transfer to the LCD panel begins when the line counter reaches the programmable line number.

13.3.2.1.1.2 Parallel Interface in Bypass Mode (MIPI DPI Protocol)

When bypass mode is enabled, the display controller must be set to use it.

Figure 13-5 shows the LCD support parallel interface in bypass mode.

Figure 13-5. LCD Support Parallel Interface (Bypass Mode)

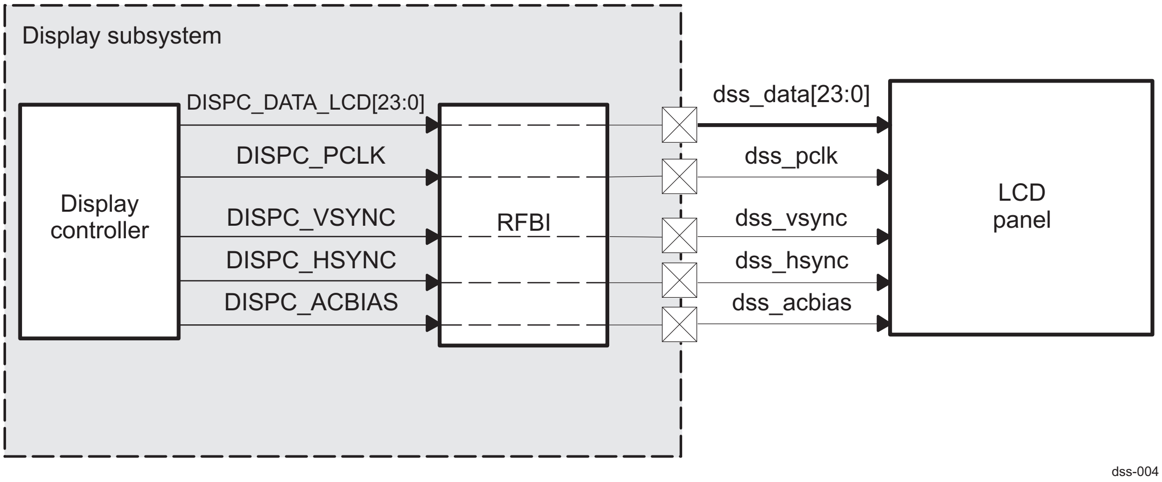


Table 13-8 describes the interface signals to/from the LCD panel in bypass mode.

Table 13-8. LCD Interface Signals (Bypass Mode)

Signal Name	Type[1]	Description
DISPC_DATA_LCD[23:0]	O	LCD data from the display controller module
DISPC_PCLK	O	Pixel CLK from the display controller module
DISPC_VSYNC	O	VSYNC from the display controller module
DISPC_HSYNC	O	HSYNC from the display controller module
DISPC_ACBIAS	O	ACBIAS from the display controller module

[1] I = Input, O = Output, I/O = Input/Output

- DISPC_DATA_LCD[23:0]: The panel pixel data comes directly from the display controller module. DISPC_DATA_LCD is connected at the chip-level boundary with dss_data[23:0].
- DISPC_PCLK: This signal is the pixel clock that comes directly from the display controller. This signal is multiplexed at the chip-level boundary with dss_pclk.
- DISPC_VSYNC: Uses the vertical synchronization signal from the display controller. The LCD frame clock (VSYNC) toggles after all the lines in a frame are transmitted to the LCD panel and a programmable number of line clock cycles has elapsed both at the beginning and at the end of each frame. This signal is multiplexed with dss_vsync at the chip-level boundary.
- DISPC_HSYNC: Uses the horizontal synchronization signal from the display controller. The LCD line clock (HSYNC) toggles after all pixels in a line are transmitted to the LCD panel and a programmable

number of pixel clock wait-states elapse, both at the beginning and at the end of each line. This signal is multiplexed on the chip-level boundary with dss_hsync.

- DISPC_ACBIAS: Uses the ac-bias signal from the display controller.
  - In passive matrix technology, the ac-bias signal is configured to transition each time a programmable number of line clocks occurs. To prevent a dc charge within the screen pixels, the power and ground supplies of the panel are periodically switched. The DISPC signals the panel to switch the polarity by toggling the ac-bias pin.
  - In active matrix technology, the ac-bias signal acts as an output-enable signal to indicate when data must be latched using the pixel clock. This signal is multiplexed on the chip-level boundary with dss_acbias.

13.3.2.1.1.3 LCD Output and Data Format for the Parallel Interface

This section describes the pixel data bus and shows timing diagrams of transactions and synchronizations in both RFBI and bypass modes.

Figure 13-6 through Figure 13-12 show the pixel data bus for bypass mode, depending on the use of 4-, 8-, 12-, 16-, 18-, or 24-pixel data output pins. In RFBI mode, the pixel data bus is reformatted in accordance with the input and output data bus width.

Table 13-9 lists the number of displayed pixels per pixel clock cycle based on the type of display panel.

Table 13-9. Number of Displayed Pixels per Pixel Clock Cycle Based on Display Type

Display Panel	Number of Displayed Pixels per Pixel Clock Cycle
Monochrome 4-bit	4
Monochrome 8-bit	8
Passive matrix color	8/3
Active matrix	1

- Passive matrix technology, Monochrome mode

  Monochrome displays use either a 4-bit or 8-bit interface. Each bit represents one pixel (on or off), which means that either 4 or 8 pixels are sent to the LCD at each pixel clock.

  Figure 13-6 and Figure 13-7 show 4- and 8-bit monochrome displays, respectively.

Figure 13-6. LCD Pixel Data Monochrome4 Passive Matrix

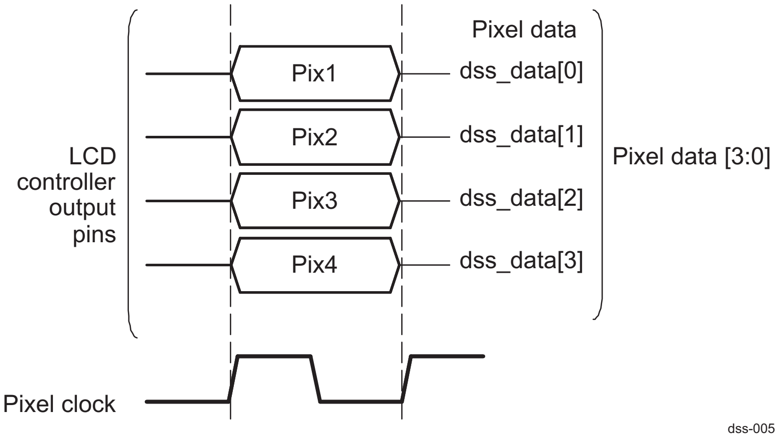
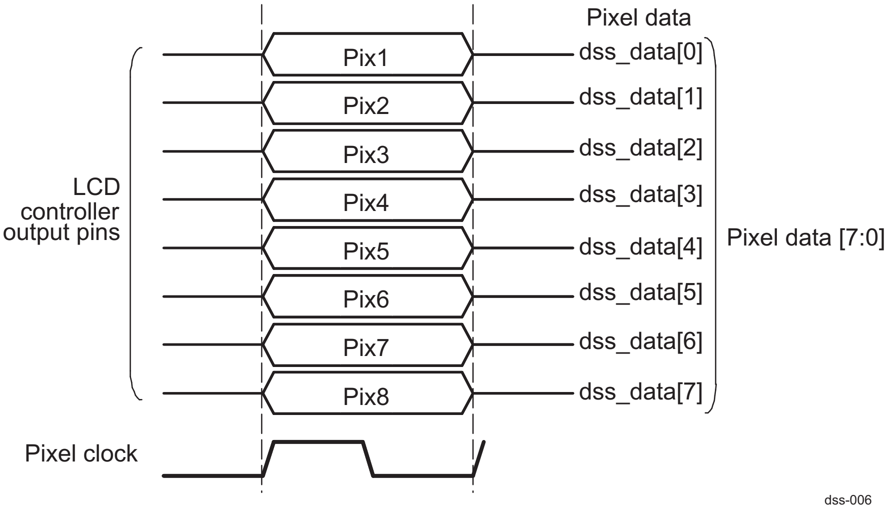


Figure 13-7. LCD Pixel Data Monochrome8 Passive Matrix



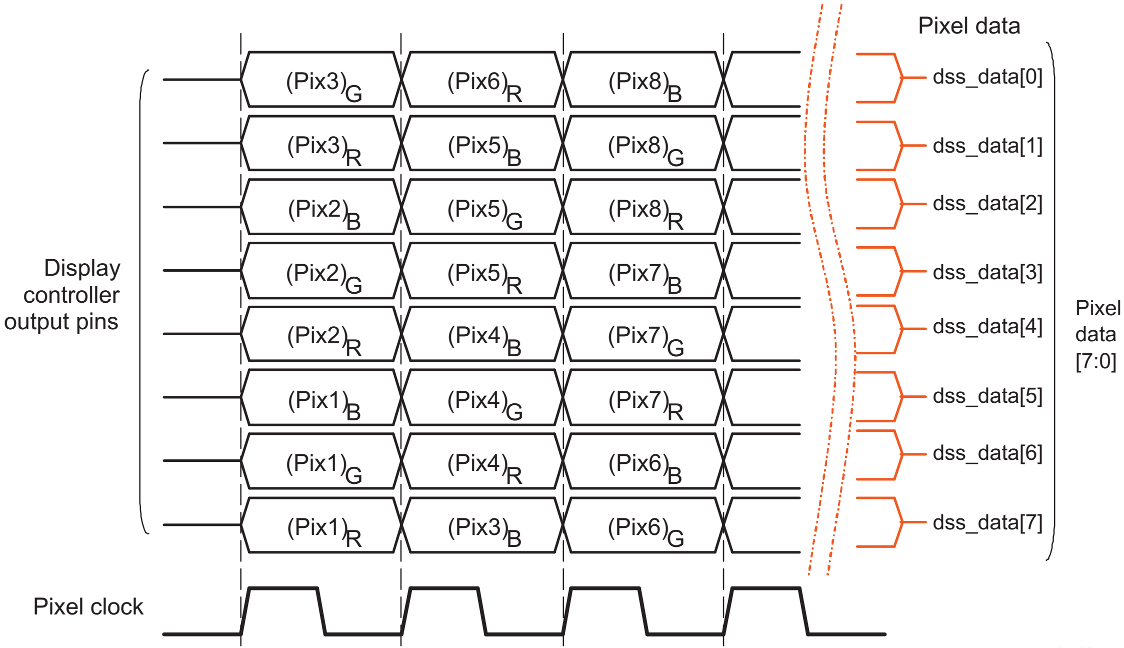
dss-006

- Passive matrix technology, color mode

  Color passive displays use 8-bit data input lines. In a given pixel clock cycle, each line represents one color component (red, green, or blue).

  Figure 13-8 shows an 8-bit color passive matrix display.

Figure 13-8. LCD Pixel Data Color Passive Matrix



dss-007

- Active matrix technology

  Active matrix displays bypass the STN dithering logic block and the output FIFO. Each line represents one pixel.

  Figure 13-9 through Figure 13-12 show 12-, 16-, 18-, and 24-active matrix displays, respectively.

Figure 13-9. LCD Pixel Data Color12 Active Matrix

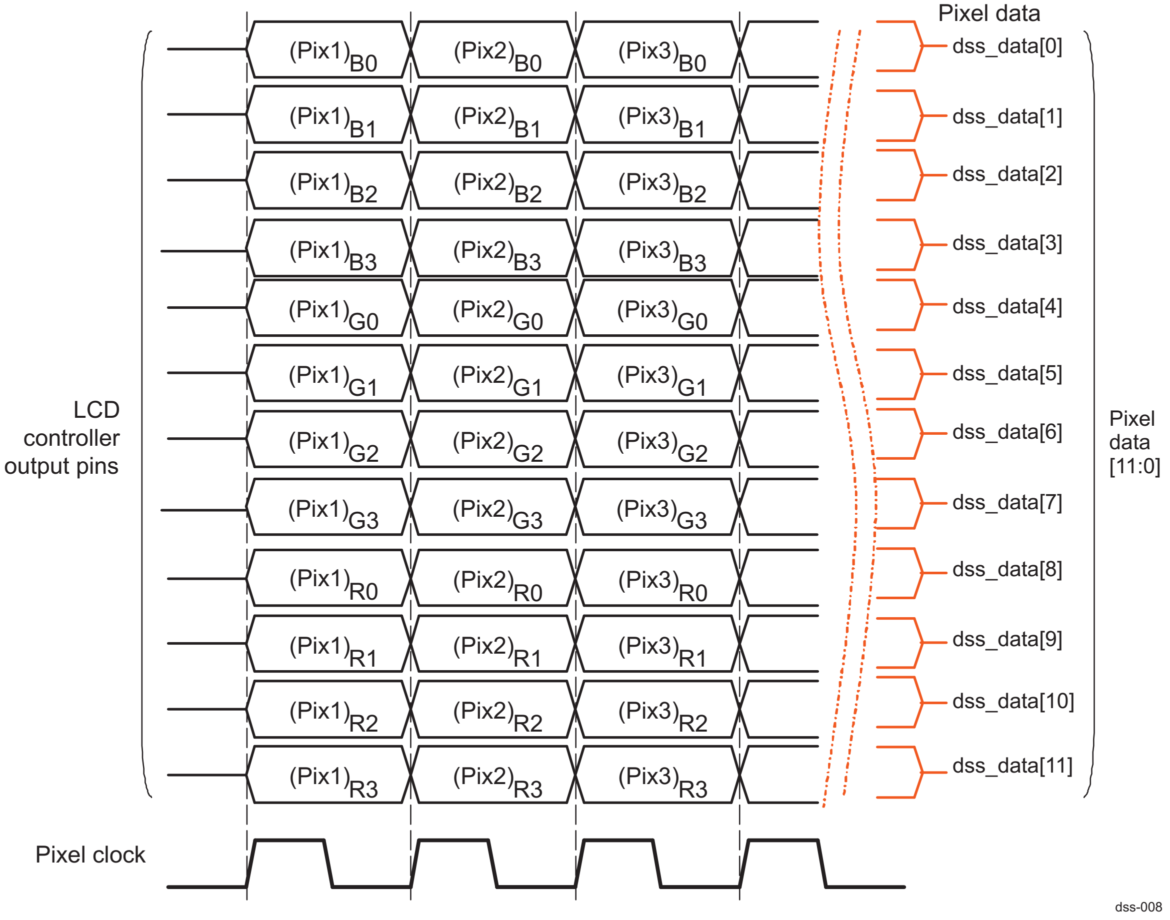


Figure 13-10. LCD Pixel Data Color16 Active Matrix

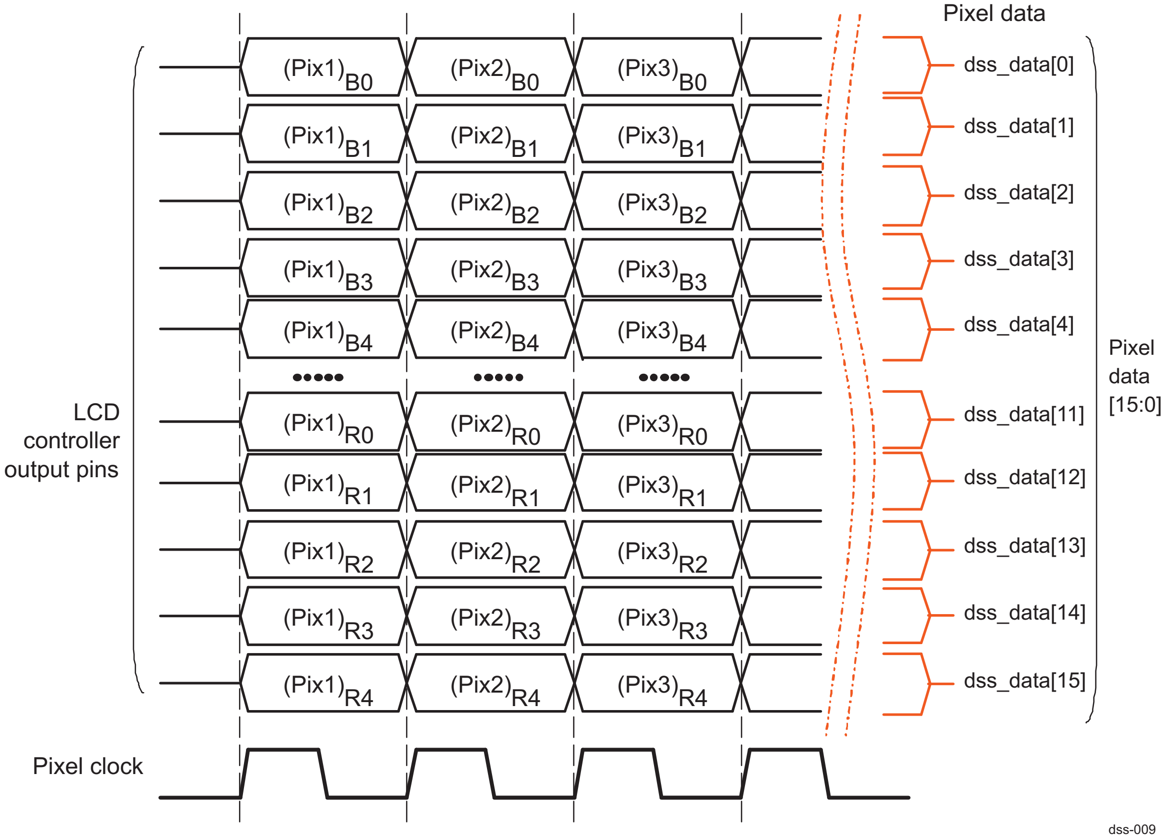


Figure 13-11. LCD Pixel Data Color18 Active Matrix

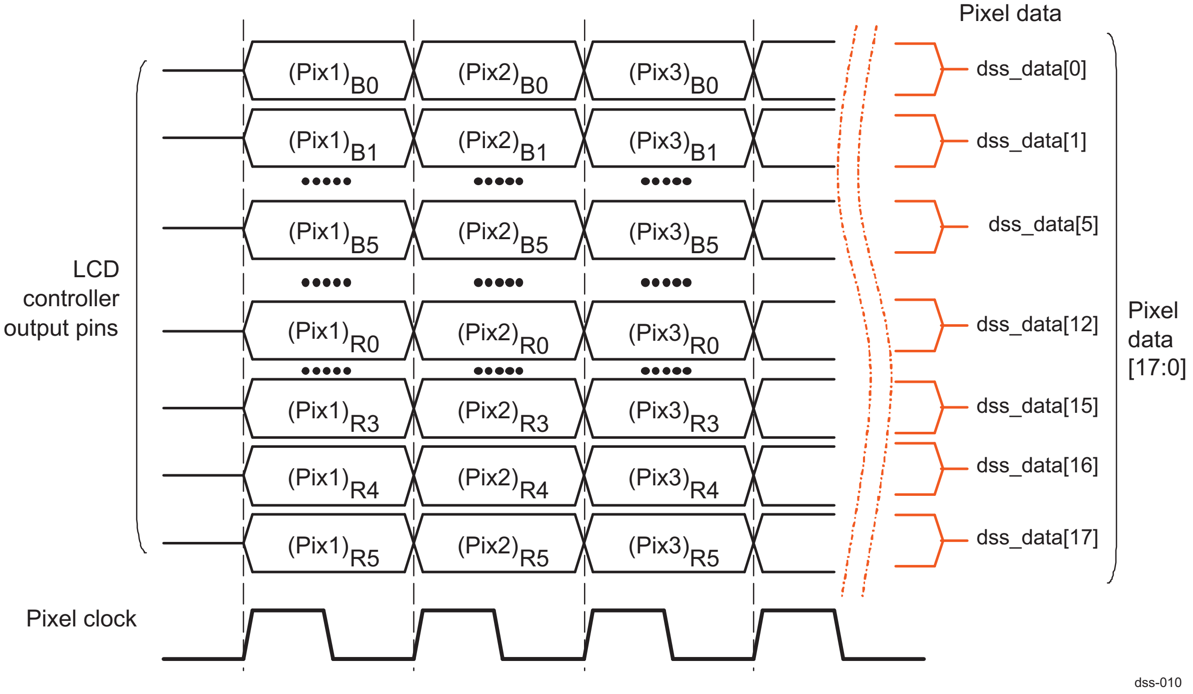
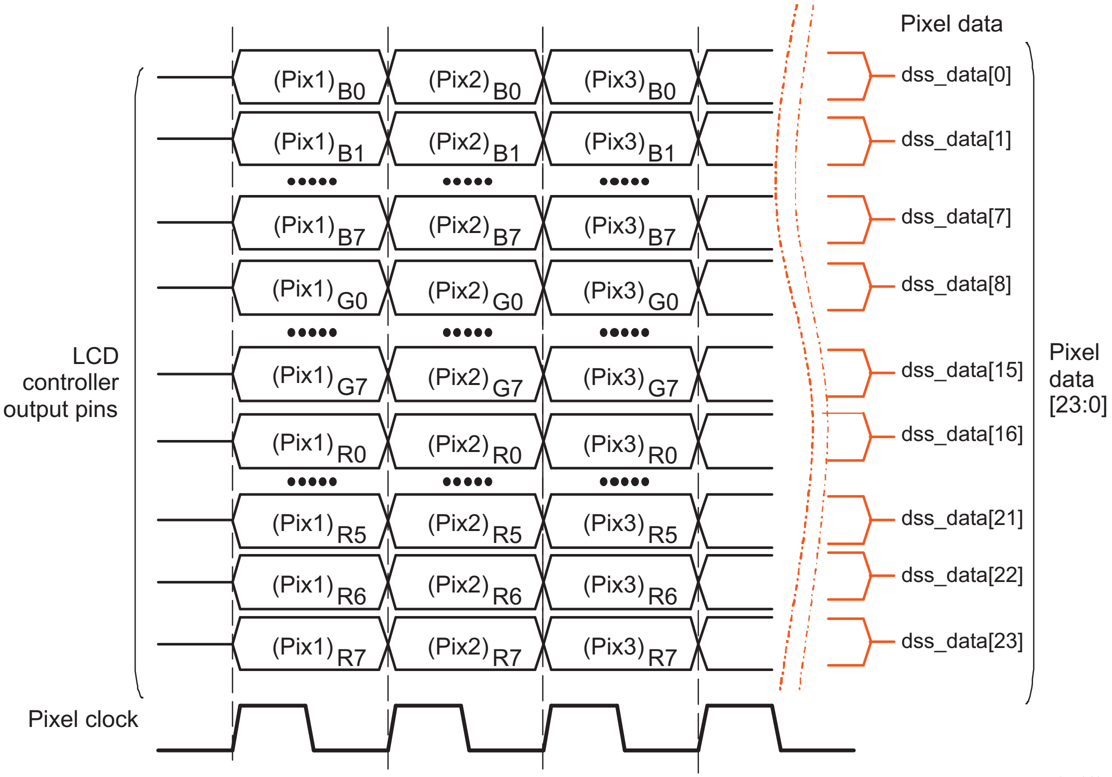


Figure 13-12. LCD Pixel Data Color24 Active Matrix

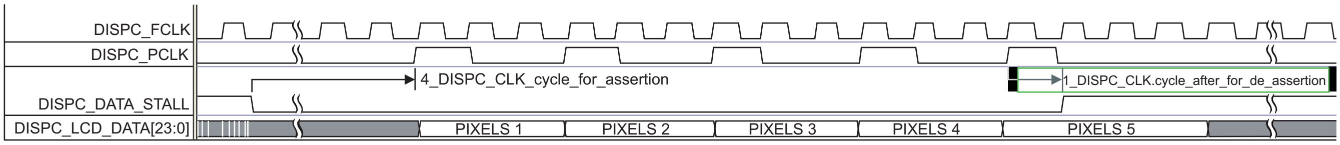


13.3.2.1.1.4 Transaction Timing Diagrams

- Timing diagrams in flow control mode
    - Stall signal

      The stall signal is used in RFBI mode. In the case of RFBI mode, it is used to indicate when the display controller must stop sending data over the LCD output interface. The RFBI module asserts the stall signal to stop data output by the display controller. It is deasserted to indicate when new data must be outputted by the display controller.

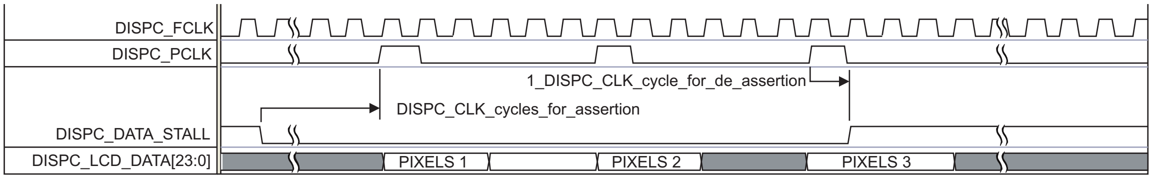
Figure 13-13. RFBI Data Stall Signal Diagram



To avoid underflow of the DMA FIFO, the FIFO handcheck feature can be enabled by setting the DSS.DISPC_CFG[16] FIFOHANDCHECK bit to 1. The fullness of the FIFOs associated with the pipelines used for the LCD output is checked when the STALL signal is inactive before providing data to the pipeline. This prevents emptying the FIFO when the RFBI module requests data and there is not enough data in the display controller DMA FIFO. This feature must be enabled only when the STALL mode is used (DSS.DISPC_CTRL[11] STALL_MODE bit set to 1).

When the FIFO handcheck feature is activated, the pixel transfer to the RFBI module during STALL inactivity period can be stopped (no DISPC_PCLK pulse) and restarted when there is enough data in the FIFO. The FIFO handcheck ensures that underflow cannot occur for the pipelines associated with the LCD output in RFBI mode. Figure 13-14 details the RFBI data stall with FIFO handcheck mode activated.

Figure 13-14. RFBI Data Stall Signal Diagram With Handcheck



- RFBI timing diagrams

Table 13-10 lists the programmable timing fields. Figure 13-15 through Figure 13-17 show timing diagrams of read/write transactions to the LCD panel for the RFBI mode.

Table 13-10. Programmable Timing Fields in RFBI Mode

Timing Name	Register Field	Description
CSOnTime	DSS.RFBI_ONOFF_TIMEi[3:0] CS_ONTIME (with I = 0 or 1)	CS assertion time from start access time
CSOffTime	DSS.RFBI_ONOFF_TIMEi[9:4] CS_OFFTIME (with I = 0 or 1)	CS deassertion time from start access time
WeCycleTime	DSS.RFBI_CYCLE_TIMEi[5:0] WE_CYCLE_TIME (with I = 0 or 1)	The time when A0 becomes valid until write cycle completion
WEOnTime	DSS_RFBI_ONOFF_TIMEi[13:10] WE_ONTIME (with I = 0 or 1)	WE assertion delay time from start access time
WEOffTime	DSS_RFBI_ONOFF_TIMEi[19:14] WE_OFFTIME (with I = 0 or 1)	WE deassertion delay time from start access time
RECycleTime	DSS.RFBI_CYCLE_TIMEi[11:6] RECYCLE_TIME (with I = 0 or 1)	The time when A0 becomes valid until read cycle completion
REOnTime	DSS_RFBI_ONOFF_TIMEi[23:20] RE_ONTIME (with I = 0 or 1)	RE assertion delay time from start access time
REOffTime	DSS.RFBI_ONOFF_TIMEi[29:24] RE_OFFTIME (with I = 0 or 1)	RE assertion delay time from start access time
CSPulseWidth	DSS.RFBI_CYCLE_TIMEi[17:12] CS_PULSE_WIDTH (with I = 0 or 1)	The time when write cycle time or read cycle time completes

Figure 13-15. Command Data Write

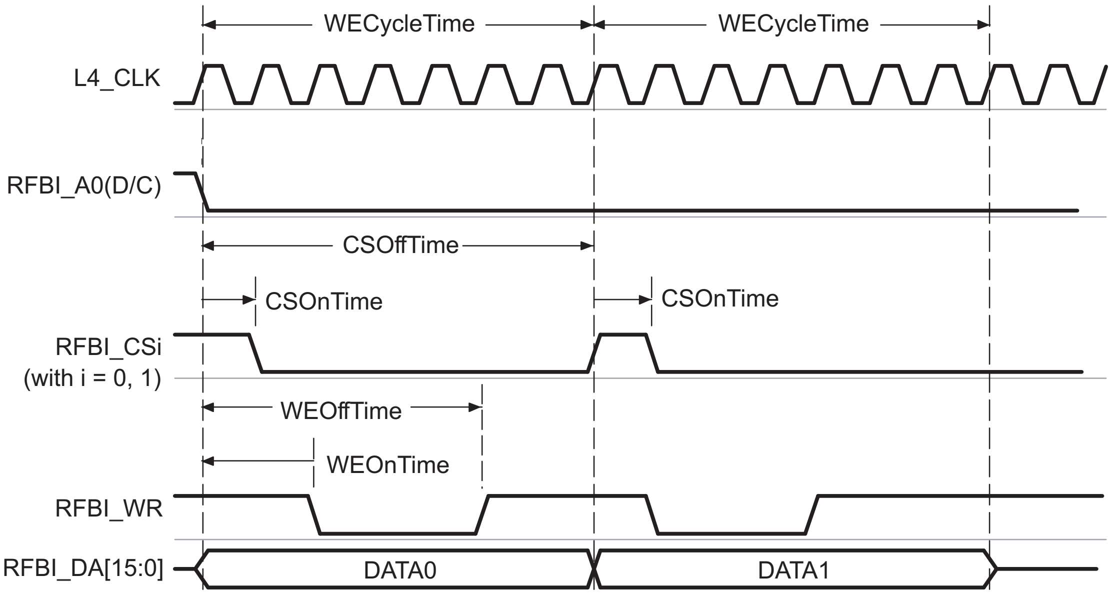


Figure 13-16. Display Data Read

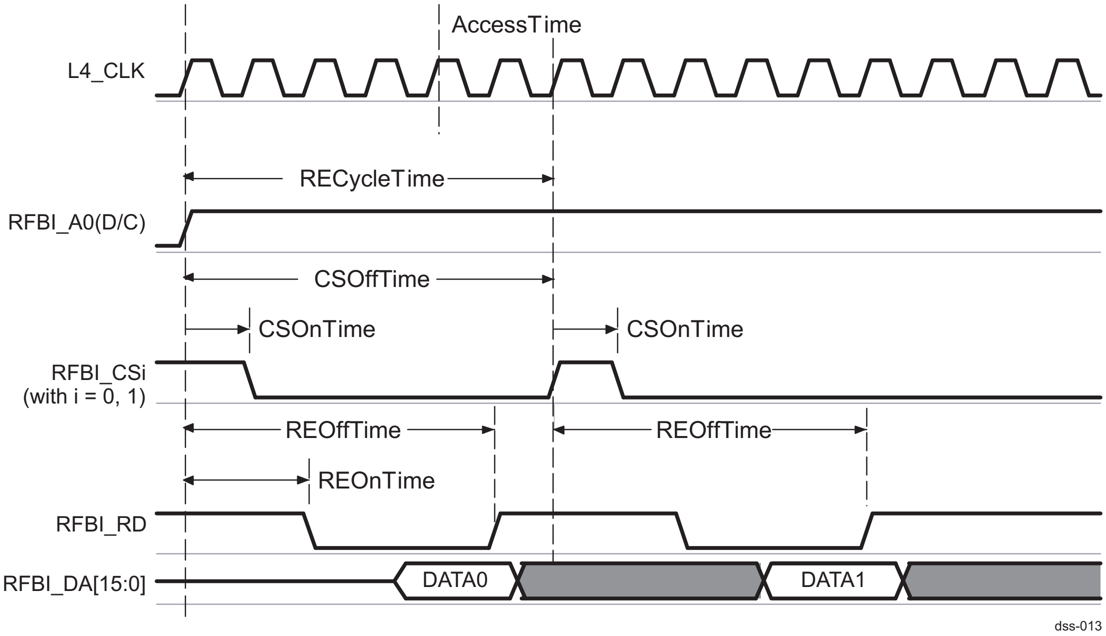
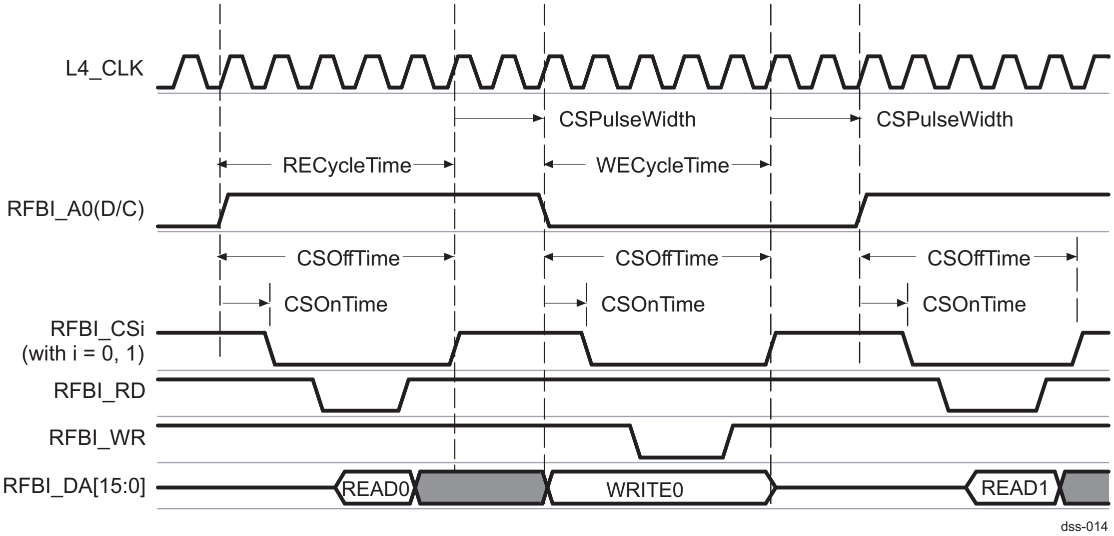


Figure 13-17. Read to Write and Write to Read



- Timing diagrams in bypass mode

  Figure 13-18 through Figure 13-33 show timing diagrams of synchronization signals and pixel clock in bypass mode for both passive matrix and active matrix panels. The display controller directly drives these signals, which are related to the programmable fields described in Table 13-11.

Table 13-11. Programmable Fields in Bypass Mode

Name	Register	Description
PPL	DSS.DISPC_SIZE_LCD[10:0] PPL bit field value + 1	Pixels per line (PPL)
LPP	DSS.DISPC_SIZE_LCD[26:16] LPP bit field value + 1	Lines per panel
HBP	DSS.DISPC_TIMING_H[31:20] HBP bit field value + 1	Horizontal back porch
HFP	DSS.DISPC_TIMING_H[19:8] HFP bit field value + 1	Horizontal front porch
HSW	DSS.DISPC_TIMING_H[7:0] HSW bit field value + 1	Horizontal synchronization pulse width

Table 13-11. Programmable Fields in Bypass Mode (continued)

Name	Register	Description
VBP	DSS.DISPC_TIMING_V[31:20] VBP bit field value	Vertical back porch
VFP	DSS.DISPC_TIMING_V[19:8] VFP bit field value	Vertical front porch
VSW	DSS.DISPC_TIMING_V[7:0] VSW bit field value + 1	Vertical synchronization pulse width
ONOFF	DSS.DISPC_POL_FREQ[17] ONOFF bit	DISPC_HSYNC and DISPC_VSYNC pixel clock control
RF	DSS.DISPC_POL_FREQ[16] RF bit	DISPC_HSYNC and DISPC_VSYNC pixel clock edge control
IEO	DSS.DISPC_POL_FREQ[15] IEO bit	Invert DISPC_ACBIAS
IPC	DSS.DISPC_POL_FREQ[14] IPC bit	Invert DISPC_PCLK
IHS	DSS.DISPC_POL_FREQ[13] IHS bit	Invert DISPC_HSYNC
IVS	DSS.DISPC_POL_FREQ[12] IVS bit	Invert DISPC_VSYNC

- Active matrix timing configuration 1
  - DSS.DISPC_POL_FREQ[17] ONOFF bit = 0
  - DSS.DISPC_POL_FREQ[16] RF bit = 0

    The DISPC_HSYNC and DISPC_VSYNC signals are driven on the opposite edge of DISPC_PCLK from the pixel data.
  - DSS.DISPC_POL_FREQ[15] IEO = 0

    The DISPC_ACBIAS signal is active high.
  - DSS.DISPC_POL_FREQ[14] IPC = 0

    The pixel data are driven on the rising edge of DISPC_PCLK.
  - DSS.DISPC_POL_FREQ[13] IHS = 0

    The DISPC_HSYNC signal is active high.
  - DSS.DISPC_POL_FREQ[12] IVS = 0

    The DISPC_VSYNC signal is active high.

Figure 13-18. Active Matrix Timing Diagram of Configuration 1 (Start of Frame)

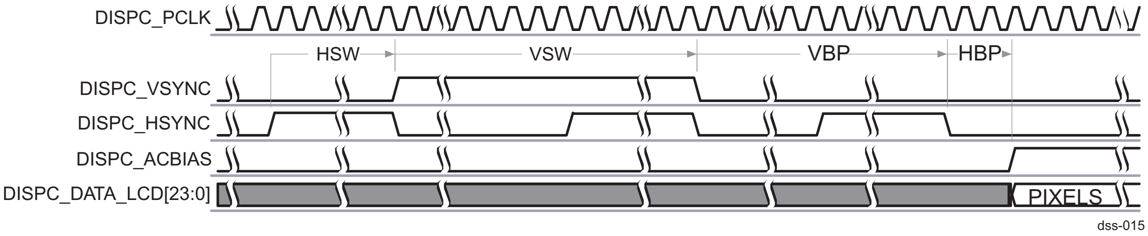


Figure 13-19. Active Matrix Timing Diagram of Configuration 1 (Between Lines)

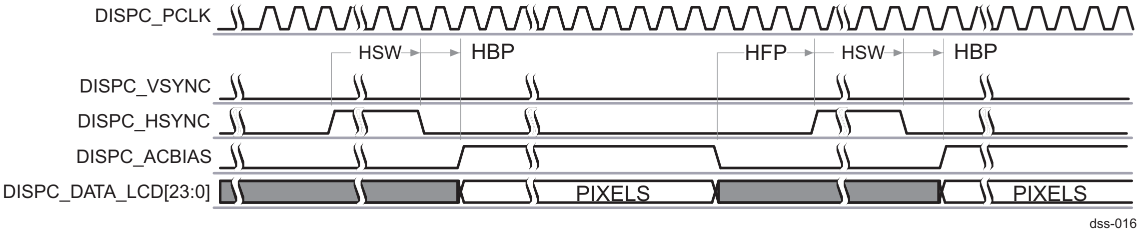


Figure 13-20. Active Matrix Timing Diagram of Configuration 1 (Between Frames)

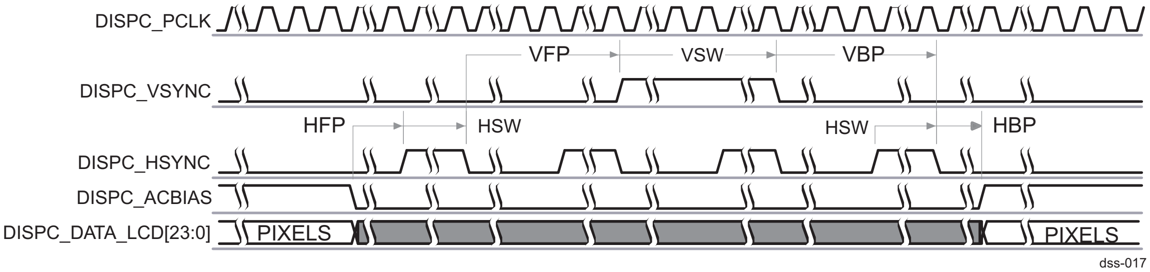
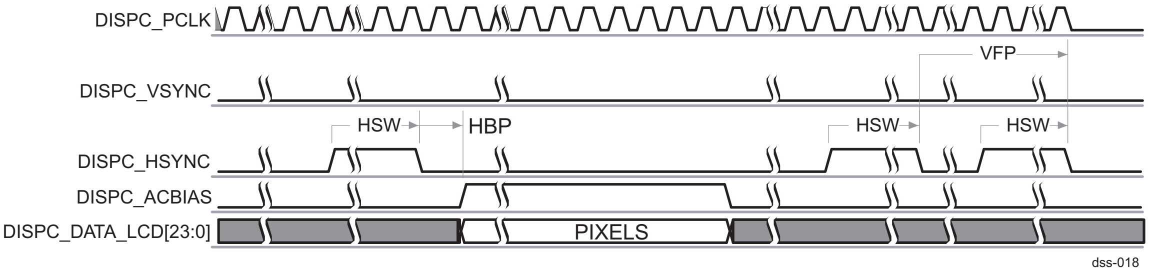


Figure 13-21. Active Matrix Timing Diagram of Configuration 1 (End of Frame)



- Active matrix timing configuration 2
  - DSS.DISPC_POL_FREQ[17] ONOFF bit = 1
  - DSS.DISPC_POL_FREQ[16] RF bit = 1

    The DISPC_HSYNC and DISPC_VSYNC signals are driven on the rising edge of DISPC_PCLK.
  - DSS.DISPC_POL_FREQ[15] IEO = 1

    The DISPC_ACBIAS signal is active low.
  - DSS.DISPC_POL_FREQ[14] IPC = 1

    The pixel data is driven on the falling edge of DISPC_PCLK.
  - DSS.DISPC_POL_FREQ[13] IHS = 1

    The DISPC_HSYNC signal is active low.
  - DSS.DISPC_POL_FREQ[12] IVS = 1

    The DISPC_VSYNC signal is active low.

Figure 13-22. Active Matrix Timing Diagram of Configuration 2 (Start of Frame)

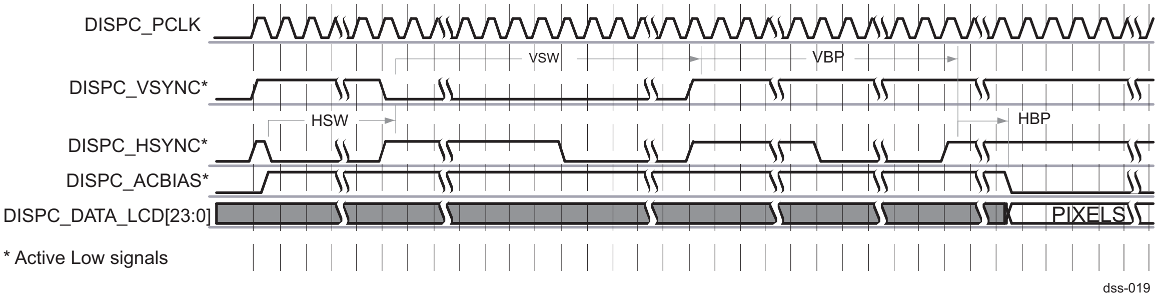


Figure 13-23. Active Matrix Timing Diagram of Configuration 2 (Between Lines)

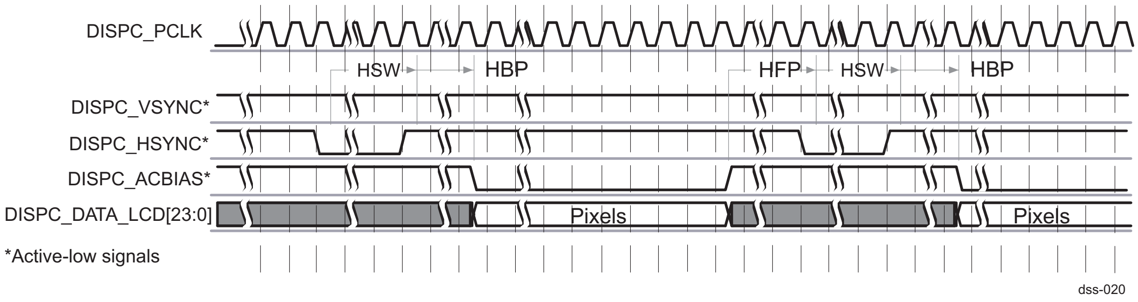


Figure 13-24. Active Matrix Timing Diagram of Configuration 2 (Between Frames)

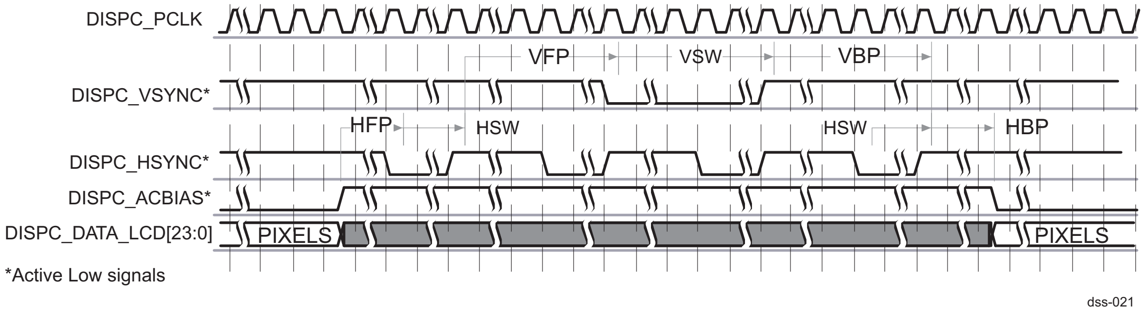
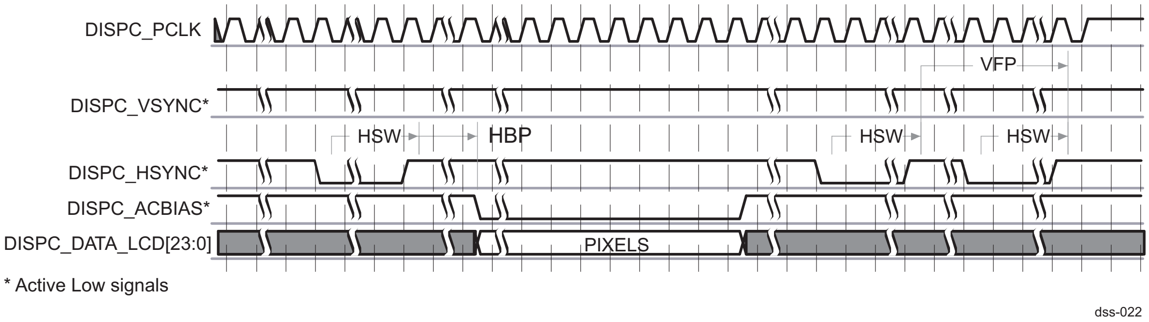


Figure 13-25. Active Matrix Timing Diagram of Configuration 2 (End of Frame)



- Active matrix timing configuration 3
  - DSS.DISPC_POL_FREQ[17] ONOFF bit = 1
  - DSS.DISPC_POL_FREQ[16] RF bit = 1

    The DISPC_HSYNC and DISPC_VSYNC signals are driven on the rising edge of DISPC_PCLK.
  - DSS.DISPC_POL_FREQ[15] IEO = 0

    The DISPC_ACBIAS signal is active high.
  - DSS.DISPC_POL_FREQ[14] IPC = 0

    The pixel data are driven on the rising edge of DISPC_PCLK.
  - DSS.DISPC_POL_FREQ[13] IHS = 0

    The DISPC_HSYNC signal is active high.
  - DSS.DISPC_POL_FREQ[12] IVS = 0

    The DISPC_VSYNC signal is active high.

Figure 13-26. Active Matrix Timing Diagram of Configuration 3 (Start of Frame)

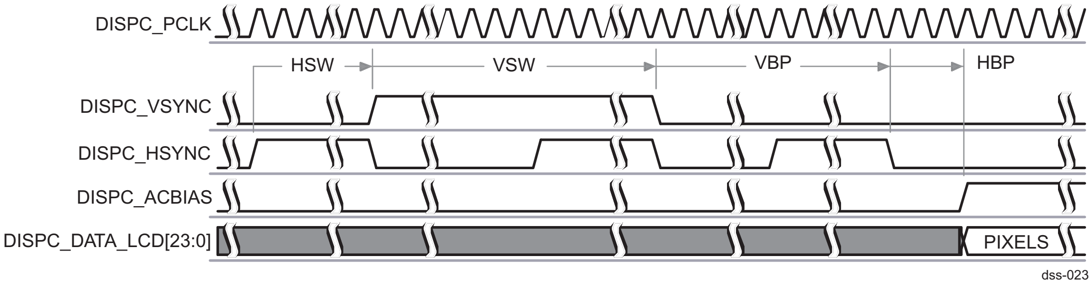


Figure 13-27. Active Matrix Timing Diagram of Configuration 3 (Between Lines)

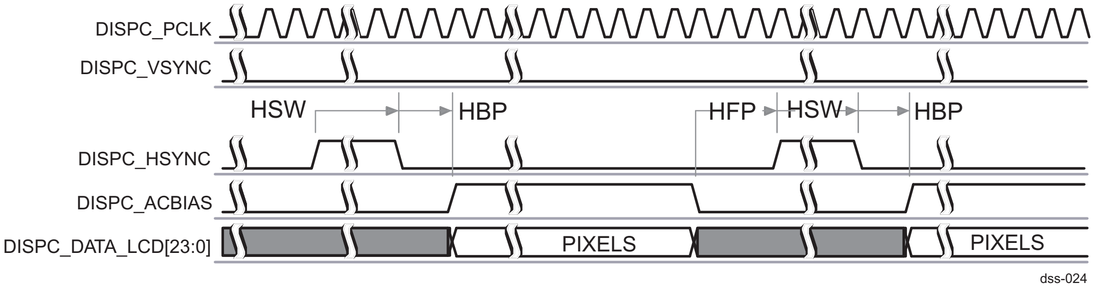


Figure 13-28. Active Matrix Timing Diagram of Configuration 3 (Between Frames)

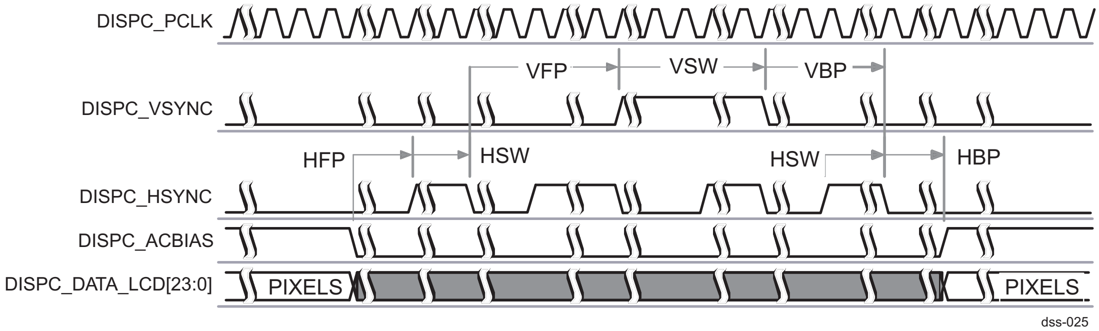
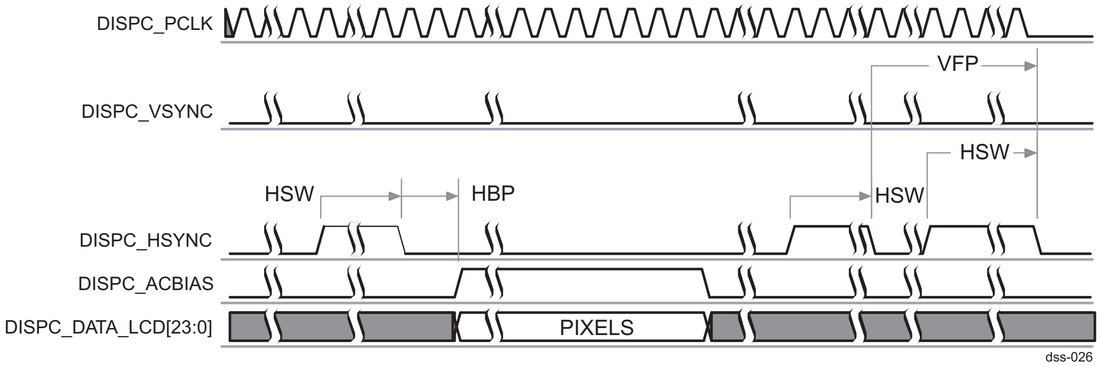


Figure 13-29. Active Matrix Timing Diagram of Configuration 3 (End of Frame)



- Passive matrix timing configuration
  - DSS.DISPC_POL_FREQ[17] ONOFF bit = 0
  - DSS.DISPC_POL_FREQ[16] RF bit = 0

    The DISPC_HSYNC and DISPC_VSYNC signals are driven on the opposite edge of DISPC_PCLK from the pixel data.

- DSS.DISPC_POL_FREQ[15] IEO = 0
  The DISPC_ACBIAS signal is active high.
- DSS.DISPC_POL_FREQ[14] IPC = 0
  The pixel data are driven on the rising edge of DISPC_PCLK.
- DSS.DISPC_POL_FREQ[13] IHS = 0
  The DISPC_HSYNC signal is active high.
- DSS.DISPC_POL_FREQ[12] IVS = 0
  The DISPC_VSYNC signal is active high.

Figure 13-30. Passive Matrix Timing Diagram (Start of Frame)

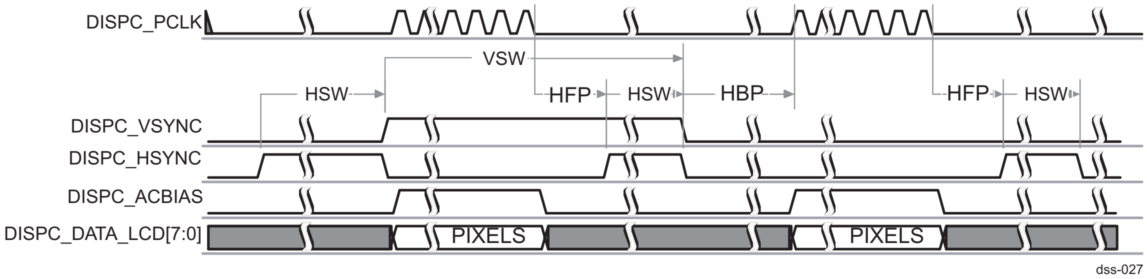


Figure 13-31. Passive Matrix Timing Diagram (Between Lines)

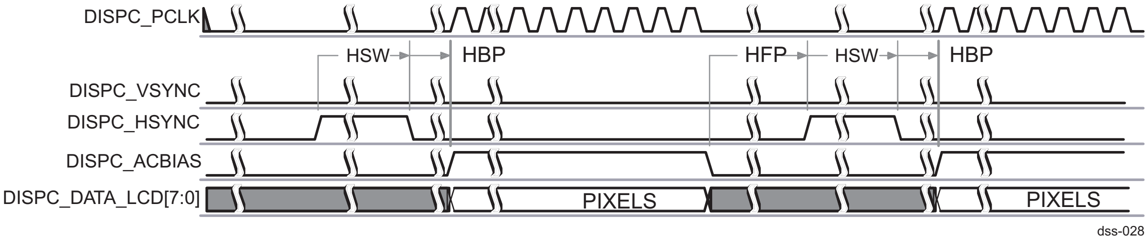


Figure 13-32. Passive Matrix Timing Diagram (Between Frames)

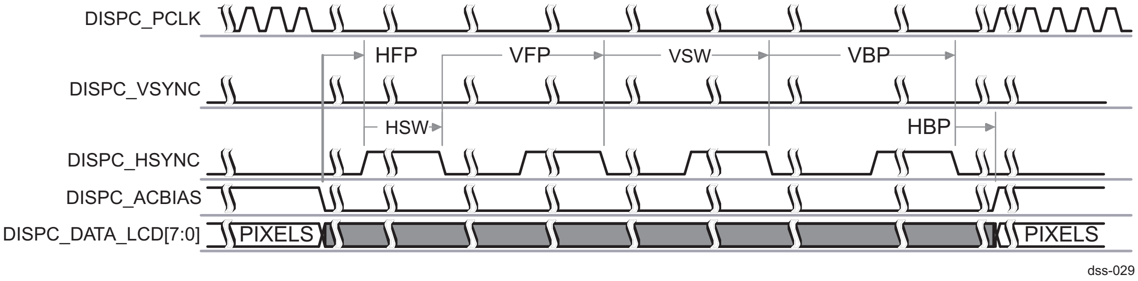
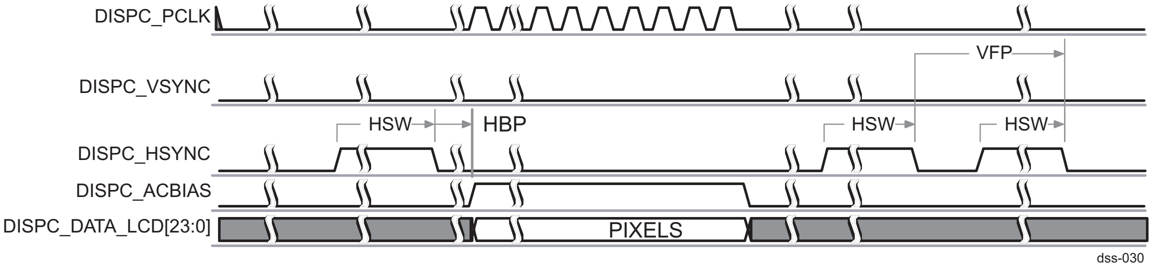


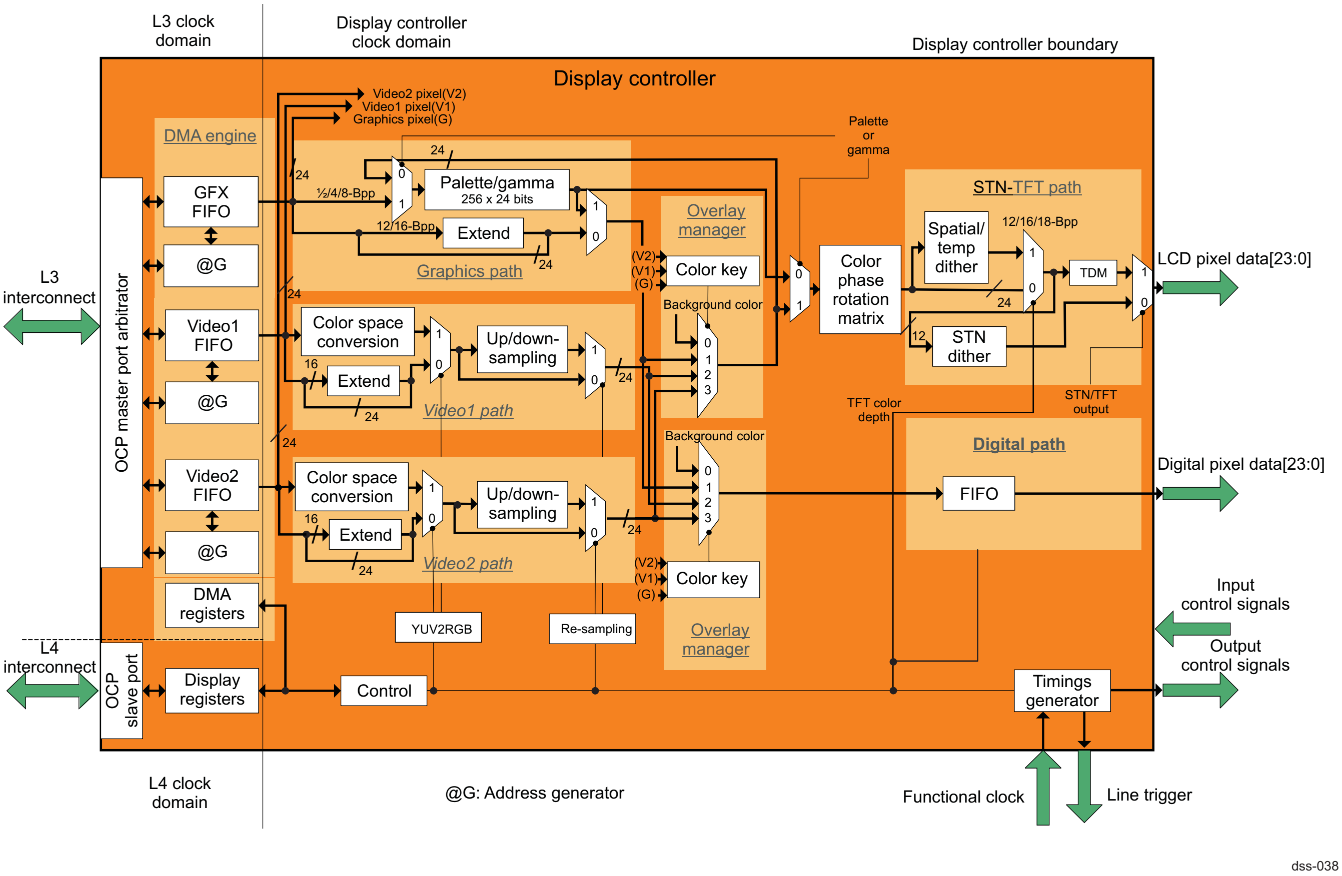
Figure 13-33. Passive Matrix Timing Diagram (End of Frame)



13.3.3 Display Controller Functionalities

The display controller can read and display the encoded pixel data stored in memory (see Figure 13-34).

Figure 13-34. Display Controller Architecture Overview



Several processes can be configured to manage the graphics pipeline (palette, gamma table correction) and video pipeline (color space conversion, upsampling, downsampling, overlay, and transparency features).

The internal timing generator logic generates the LCD input signals. The external timing generator generates the appropriated signals to drive the digital output. The data from the two overlay managers are sent on the two concurrent 24-bit buses outside the display controller module. The memory accessed by the display controller is either the SDRAM memory or the SRAM memory.

13.3.3.1 Display Modes

13.3.3.1.1 LCD Output

The display subsystem supports two types of display technologies (both monochrome and color modes):

- Passive matrix displays
- Active matrix displays

The passive matrix display mode supports 3375 possible colors, allowing 16, 256, or 3375 colors to be displayed in each frame, depending on the color depth. The monochrome LCD has 15 grayscale levels available.

In active matrix display mode, the configuration of colors depends on the color depth:

- 24 BPP supports 16,777,216 colors.
- 18 BPP supports 262,144 colors.
- 16 BPP supports 65,536 colors.
- 12 BPP supports 4096 colors.

13.3.3.1.2 Digital Output

The digital output is always a 24-bit RGB value based on an external pixel request.

13.3.3.2 Graphics Pipeline

The graphics pipeline is connected to the graphics FIFO controller for the input port and to the two overlay managers (LCD and digital). It consists of one 256-entry palette and some programmable replication logic. The replication logic is used to convert the RGB pixels, excluding the RGB24 format, into RGB24 format based on user programming (replication of the most-significant bits [MSBs] for the RGB24 LSBs or use of 0s). The first unit connected to the input port of the graphics pipeline is the replication logic used for RGB pixels, then the second unit is the palette for concerned pixels.

13.3.3.2.1 Graphics Memory Format

The supported formats for the graphics layer are CLUT bitmaps (1-, 2-, 4-, and 8-BPP) and true color bitmaps in RGB formats (12-, 16-, and 24-BPP [packet and nonpacket RGB24]) and in ARGB or RGBA formats (ARGB 16-, and 32-BPP, and RGBA 32-BPP) as follows:

- BITMAP 1-BPP data memory organization (CLUT) (little endian)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P31	P30	P29	P28	P27	P26	P25	P24	P23	P22	P21	P20	P19	P18	P17	P16	P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0

dss-T039

- BITMAP 1-BPP data memory organization (CLUT) (little endian + nibble mode)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P24	P25	P26	P27	P28	P29	P30	P31	P16	P17	P18	P19	P20	P21	P22	P23	P8	P9	P10	P11	P12	P13	P14	P15	P0	P1	P2	P3	P4	P5	P6	P7

dss-T040

- BITMAP 1-BPP data memory organization (CLUT) (big endian)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0	P15	P14	P13	P12	P11	P10	P9	P8	P23	P22	P21	P20	P19	P18	P17	P16	P31	P30	P29	P28	P27	P26	P25	P24

dss-T060

- BITMAP 1-BPP data memory organization (CLUT) (big endian + nibble mode)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	P31

dss-T061

- BITMAP 2-BPP data memory organization (CLUT) (little endian)

<table><tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="2">P15</td><td colspan="2">P14</td><td colspan="2">P13</td><td colspan="2">P12</td><td colspan="2">P11</td><td colspan="2">P10</td><td colspan="2">P9</td><td colspan="2">P8</td><td colspan="2">P7</td><td colspan="2">P6</td><td colspan="2">P5</td><td colspan="2">P4</td><td colspan="2">P3</td><td colspan="2">P2</td><td colspan="2">P1</td><td colspan="2">P0</td></tr></table>

dss-T041

- BITMAP 2-BPP data memory organization (CLUT) (little endian + nibble mode)

<table><tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="2">P12</td><td colspan="2">P13</td><td colspan="2">P14</td><td colspan="2">P15</td><td colspan="2">P8</td><td colspan="2">P9</td><td colspan="2">P10</td><td colspan="2">P11</td><td colspan="2">P4</td><td colspan="2">P5</td><td colspan="2">P6</td><td colspan="2">P7</td><td colspan="2">P0</td><td colspan="2">P1</td><td colspan="2">P2</td><td colspan="2">P3</td></tr></table>

dss-T042

- BITMAP 2-BPP data memory organization (CLUT) (big endian)

<table><tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="2">P3</td><td colspan="2">P2</td><td colspan="2">P1</td><td colspan="2">P0</td><td colspan="2">P7</td><td colspan="2">P6</td><td colspan="2">P5</td><td colspan="2">P4</td><td colspan="2">P11</td><td colspan="2">P10</td><td colspan="2">P9</td><td colspan="2">P8</td><td colspan="2">P15</td><td colspan="2">P14</td><td colspan="2">P13</td><td colspan="2">P12</td></tr></table>

dss-T062

- BITMAP 2-BPP data memory organization (CLUT) (big endian + nibble mode)

<table><tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="2">P0</td><td colspan="2">P1</td><td colspan="2">P2</td><td colspan="2">P3</td><td colspan="2">P4</td><td colspan="2">P5</td><td colspan="2">P6</td><td colspan="2">P7</td><td colspan="2">P8</td><td colspan="2">P9</td><td colspan="2">P10</td><td colspan="2">P11</td><td colspan="2">P12</td><td colspan="2">P13</td><td colspan="2">P14</td><td colspan="2">P15</td></tr></table>

dss-T063

- BITMAP 4-BPP data memory organization (CLUT) (little endian)

<table><tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="4">Pixel 7</td><td colspan="4">Pixel 6</td><td colspan="4">Pixel 5</td><td colspan="4">Pixel 4</td><td colspan="4">Pixel 3</td><td colspan="4">Pixel 2</td><td colspan="4">Pixel 1</td><td colspan="4">Pixel 0</td></tr></table>

dss-T043

- BITMAP 4-BPP data memory organization (CLUT) (little endian + nibble mode)

<table><tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="4">Pixel 6</td><td colspan="4">Pixel 7</td><td colspan="4">Pixel 4</td><td colspan="4">Pixel 5</td><td colspan="4">Pixel 2</td><td colspan="4">Pixel 3</td><td colspan="4">Pixel 0</td><td colspan="4">Pixel 1</td></tr></table>

dss-T044

- BITMAP 4-BPP data memory organization (CLUT) (big endian)

<table><tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="4">Pixel 1</td><td colspan="4">Pixel 0</td><td colspan="4">Pixel 3</td><td colspan="4">Pixel 2</td><td colspan="4">Pixel 5</td><td colspan="4">Pixel 4</td><td colspan="4">Pixel 7</td><td colspan="4">Pixel 6</td></tr></table>

dss-T064

- BITMAP 4-BPP data memory organization (CLUT) (big endian + nibble mode)

<table><tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="4">Pixel 0</td><td colspan="4">Pixel 1</td><td colspan="4">Pixel 2</td><td colspan="4">Pixel 3</td><td colspan="4">Pixel 4</td><td colspan="4">Pixel 5</td><td colspan="4">Pixel 6</td><td colspan="4">Pixel 7</td></tr></table>

dss-T065

- BITMAP 8-BPP data memory organization (CLUT) (little endian)

<table><tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Pixel 3</td><td colspan="8">Pixel 2</td><td colspan="8">Pixel 1</td><td colspan="8">Pixel 0</td></tr></table>

dss-T045

- BITMAP 8-BPP data memory organization (CLUT) (big endian)

<table><tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Pixel 0</td><td colspan="8">Pixel 1</td><td colspan="8">Pixel 2</td><td colspan="8">Pixel 3</td></tr></table>

dss-T066

- RGB 12-BPP data memory organization (little endian)

<table><tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="4">Unused</td><td colspan="4">R1</td><td colspan="4">G1</td><td colspan="4">B1</td><td colspan="4">Unused</td><td colspan="4">R0</td><td colspan="4">G0</td><td colspan="4">B0</td></tr></table>

dss-T046

- RGB 12-BPP data memory organization (big endian)

<table><tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="4">Unused</td><td colspan="4">R0</td><td colspan="4">G0</td><td colspan="4">B0</td><td colspan="4">Unused</td><td colspan="4">R1</td><td colspan="4">G1</td><td colspan="4">B1</td></tr></table>

dss-T067

- RGB 16-BPP data memory organization (little endian)

<table><tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="5">R1</td><td colspan="6">G1</td><td colspan="5">B1</td><td colspan="5">R0</td><td colspan="6">G0</td><td colspan="5">B0</td></tr></table>

dss-T048

- RGB 16-BPP data memory organization (big endian)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0					G0						B0					R1					G1						B1				

dss-T070

- ARGB 16-BPP data memory organization (little endian)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A1				R1				G1				B1				A0				R0				G0				B0			

dss-T049

- ARGB 16-BPP data memory organization (big endian)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A0				R0				G0				B0				A1				R1				G1				B1			

dss-T069

- RGB 24-BPP data memory organization (little or big endian)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Unused								R								G								B							

dss-T050

- ARGB 32-BPP data memory organization (little or big endian)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A								R								G								B							

dss-T051

- RGBA 32-BPP data memory organization (little or big endian)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R								G								B								A							

dss-T052

- RGB 24-BPP packet data memory organization (little or big endian)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
B1								R0								G0								B0								+0x0
G2								B2								R1								G1								+0x4
R3								G3								B3								R2								+0x8

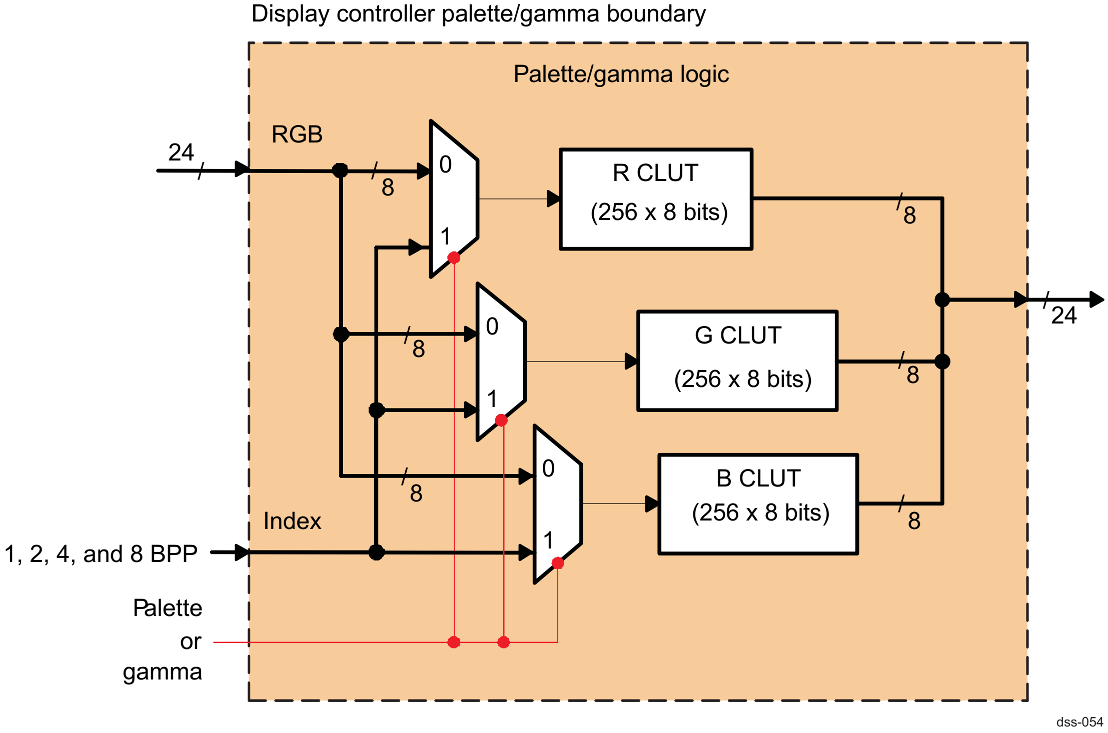
dss-T053

13.3.3.2.2 Color Look-Up Table/Gamma Table

The graphics path supports the palette/gamma table. Figure 13-35 shows the internal architecture of the color look-up/gamma table.

The palette is split into three memories of 256-bit x 8-bit entries. For bitmap (CLUT) indexes, the same value (1-, 2-, 4-, or 8-BPP) indexes the three memories. For gamma curve correction, each R, G, and B component indexes the corresponding memory to combine the three gamma curve values into a 24-bit value. The table can be reloaded every frame, once or never (at the beginning of the frame before fetching the pixels for the graphics and/or video windows).

Figure 13-35. Palette/Gamma Correction Architecture



13.3.3.2.2.1 *Color Look-Up Table*

The palette mode uses the encoded pixel values from the input graphics FIFO as pointers to index the 24-bit-wide palette: 1-BPP pixels address 2 palette entries, 2-BPP pixels address 4 palette entries, 4-BPP pixels address 16 palette entries, and 8-BPP pixels address 256 palette entries.

When a palette entry is selected by the encoded pixel value, the content of the entry is sent to the color/grayscale space/time base passive matrix dithering circuit, or to the color time base active matrix dithering circuit.

In color mode, the value within the palette is made up of three 8-bit fields, one for each color component (red, green, and blue). For color operation, an individual frame is limited to a selection of 256 colors (the number of palette entries). The format of one of the palette values in the memory is as follows:

- 24-BPP Data Memory Organization (Little Endian or Nibble)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Unused								R								G								B							

In monochrome mode, only one 8-bit value is present.

- 24-BPP Data Memory Organization (Little Endian or Nibble)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Unused								Unused								Unused								Gray							

After passing through the palette, 256 gray scales and 16,777,216 colors are numbers obtained. A redundancy introduced in the dithering logic step reduces these numbers when displaying. For passive matrix panels, the colors are limited to 15 gray scales and 3375 colors.

- Passive matrix technology

  The palette is bypassed in 12, 16, and 24 BPP. The palette is not used.
- Active matrix technology

The palette is bypassed in 12, 16, and 24 BPP, allowing up to 2^{24} = 16,777,216 colors to be displayed.

13.3.3.2.2.2 Gamma Table

In the gamma curve mode, the selected encoded pixel values based on the color keys from the video or graphics paths are sent to the gamma curve table. The mode is available only if the color look-up palette is not used for graphics. The output of the gamma curve processing is always sent to the LCD output. It is not available on digital output.

Each component of encoded pixel value is used as a pointer to index 1 out of 256 24-bit gamma curve entries in the table. Each 8-bit component is replaced with the 8-bit table value corresponding to an R, G, or B component. The format of one of the gamma curve values in the memory is as follows:

- 24-BPP Data Memory Organization (Little or Big Endian)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Unused								Gamma-R								Gamma-G								Gamma-B							

13.3.3.2.2.2.1 Replication Logic

The replication logic increases the color depth of the graphics and video encoded pixels (from true color RGB 12-, and 16-BPP to 24-BPP). The encoded value is shifted to the 24-bit alignment. The MSB bits are copied to the LSB missing ones. Then the graphics are merged with the video data based on the transparency color keys. When the replication logic is not selected, the encoded pixel values are shifted to the MSB boundary of the 24-bit format. The missing bit values are filled up with 0s.

This is an example for RGB16 extension:

- Original 16-BPP data:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0

dss-T055

- If replication logic is ON:

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R4	R3	R2	R1	R0	R4	R3	R2	G5	G4	G3	G2	G1	G0	G5	G4	B4	B3	B2	B1	B0	B4	B3	B2

dss-T056

- If replication logic is OFF:

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R4	R3	R2	R1	R0	0	0	0	G5	G4	G3	G2	G1	G0	0	0	B4	B3	B2	B1	B0	0	0	0

dss-T057

13.3.3.3 Video Pipeline

The video pipeline is connected to the video FIFO controller for the input port and to the two overlay managers (LCD and digital). It consists of the Re-Sampling unit, the Color Space Conversion Unit, and some programmable replication logic. The replication logic is used to convert the RGB pixels, excluding the RGB24 format, into RGB24 format based on user programming (replication of the MSBs for the RGB24 LSBs or use of 0s). The first unit connected to the input port of the video pipeline is the Re-Sampling Unit, then the replication logic used for RGB pixels, then the Color Space Conversion Unit for YUV4:2:2 pixels.

13.3.3.3.1 Video Memory Formats

The display subsystem supports the following formats for the video layer: YUV2, UYVY, RGB12, RGB16, RGB24 (non-packed and packed formats), ARGB16 (video channel 2 only), ARGB32 (video channel 2 only), and RGBA32 (video channel 2 only).

- RGB 12-BPP data memory organization (little endian)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Unused				R1				G1				B1				Unused				R0				G0				B0			

dss-T046

- RGB 12-BPP data memory organization (big endian)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Unused				R0				G0				B0				Unused				R1				G1				B1			

dss-T067

- RGB 16-BPP data memory organization (little endian)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R1					G1						B1					R0					G0						B0				

dss-T048

- RGB 16-BPP data memory organization (big endian)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0					G0						B0					R1					G1						B1				

dss-T070

- ARGB 16-BPP data memory organization (little endian + video 2 channel only)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A1				R1				G1				B1				A0				R0				G0				B0			

dss-T049

- ARGB 16-BPP data memory organization (big endian + video 2 channel only)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A0				R0				G0				B0				A1				R1				G1				B1			

dss-T069

- RGB 24-BPP data memory organization (little or big endian)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Unused								R								G								B							

dss-T050

- RGB 24-BPP packet data memory organization (little or big endian)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
B1								R0								G0								B0								+0x0
G2								B2								R1								G1								+0x4
R3								G3								B3								R2								+0x8

dss-T053

- ARGB 32-BPP data memory organization (little or big endian + video 2 channel only)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A								R								G								B							

dss-T051

- RGBA 32-BPP data memory organization (little or big endian + video 2 channel only)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R								G								B								A							

dss-T052

- UYVY 4:2:2 data memory organization (little endian)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Y1								Cr0								Y0								Cb0							

dss-T058

- UYVY 4:2:2 data memory organization (big endian)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cb0								Y0								Cr0								Y1							

dss-T071

- YUV2 4:2:2 data memory organization (little endian)

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Cr0</td><td colspan="8">Y1</td><td colspan="8">Cb0</td><td colspan="8">Y0</td></tr>
</table>

dss-T059

- YUV2 4:2:2 data memory organization (big endian)

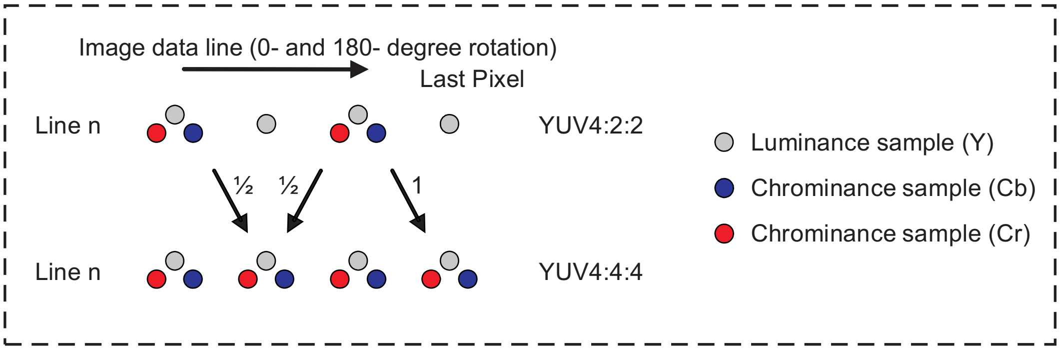
<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Y0</td><td colspan="8">Cb0</td><td colspan="8">Y1</td><td colspan="8">Cr0</td></tr>
</table>

dss-T072

13.3.3.3.2 Color Space Conversion

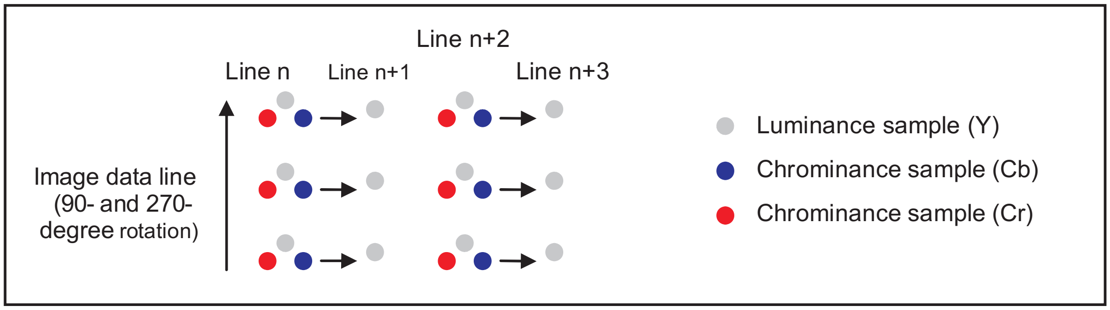
The color space conversion module converts the video-encoded pixel values from YCbCr 4:2:2 format into RGB24. Figure 13-36 and Figure 13-37 detail the YCbCr 4:2:2 conversion to YCbCr 4:4:4 depending on the rotation parameters.

Figure 13-36. YCbCr 4:2:2 to YCbCr 4:4:4 (0- or 180-Degree Rotation)



dss-061

Figure 13-37. YCbCr 4:2:2 to YCbCr 4:4:4 (90- or 270-Degree Rotation)



dss-060

The interpolation of the missing chrominance component is given by the equation in Figure 13-38.

Figure 13-38. Interpolation of the Missing Chrominance Component

$$Cb_n(YCbCr\ 444) = \frac{Cb_{n-1}(YCbCr\ 422) + Cb_{n+1}(YCbCr\ 422)}{2}\ (\text{n odd})$$

$$Cr_n(YCbCr\ 444) = \frac{Cr_{n-1}(YCbCr\ 422) + Cr_{n+1}(YCbCr\ 422)}{2}\ (\text{n odd})$$

dss-E062

First, to convert the YCbCr 4:2:2 encoded pixel values into YCbCr 4:4:4 format, the missing chrominance samples (Cb and Cr) are interpolated using the average values of the two closest values on the same line (1/2, 1/2) or are repeated from the second pixel in the same 32-bit container.

- In case of rotation 0-degree, for the last pixel, the chrominance samples are duplicated using the values from the previous pixel; otherwise, the chrominance samples are averaged using the two adjacent values.
- In case of 180-degree rotation, for the first pixel the chrominance samples missing are duplicated from

the adjacent pixel; otherwise, the chrominance samples are averaged using the two adjacent values.

- In case of rotation 90- and 270-degree, the missing chrominance components are duplicated from the adjacent pixel in the same 32-bit container.

In case of 5-tap configuration for the vertical filtering, the missing chrominance samples are always duplicated using the second chrominance samples in the same 32-bit value.

Then the pixels are converted from YCbCr color space into the RGB color space, because the output format of the color space conversion is RGB24 (8-bit value per component: Red, green, and blue). The following matrices show the 11-bit coefficients registers used to convert from YCbCr 4:4:4 into RGB24. Users set the coefficients according to the standard used to encode the pixel data in YCbCr color space.

In case of resampling, the YUV4:2:2 format is converted into YUV4:4:4. The YUV4:2:2-to-YUV4:4:4 processing is bypassed in the color space conversion unit.

If the active range for the luminance samples (Y) is [16:235] and [16:240] for the chrominance samples (Cb and Cr), the values of R, G, and B output components are clipped to the range [0:255]. The equation shown in Figure 13-39 gives the 11-bit coefficients of the YCbCr to RGB color space conversion.

Figure 13-39. YCbCr to RGB Registers (VIDFULLRANGE = 0)

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} RY & RCr & RCb \\ GY & GCr & GCb \\ BY & BCr & BCb \end{bmatrix} * \begin{bmatrix} Y - 16 \\ Cr - 128 \\ Cb - 128 \end{bmatrix}$$

dss-E063

If the active range for the luminance samples (Y) and chrominance samples (Cb and Cr) is [0:255], the values of R, G, and B output components are clipped to the range [0:255]. The equation shown in Figure 13-40 gives the 11-bit coefficients of the YCbCr-to-RGB color space conversion.

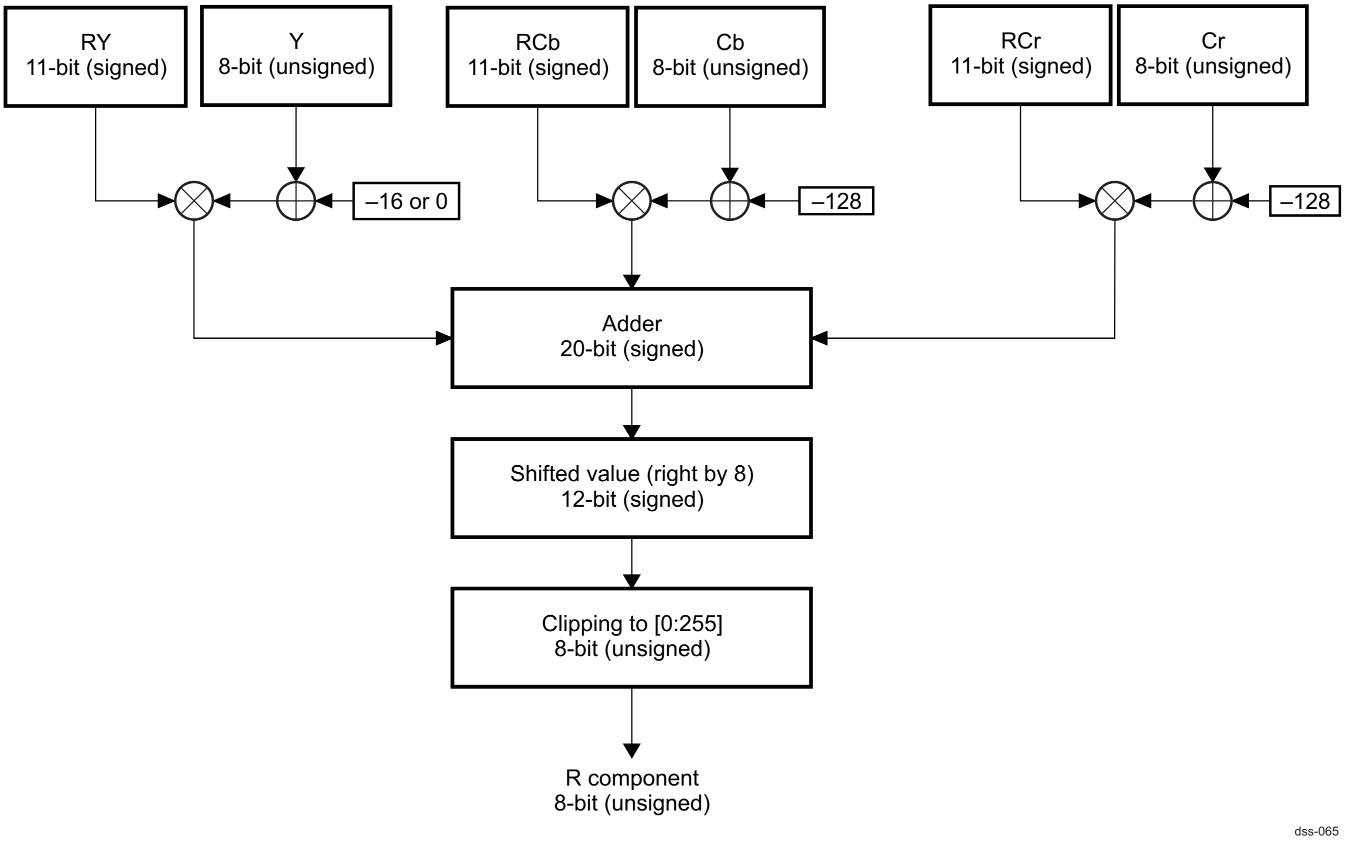
Figure 13-40. YCbCr to RGB Registers (VIDFULLRANGE = 1)

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} RY & RCr & RCb \\ GY & GCr & GCb \\ BY & BCr & BCb \end{bmatrix} * \begin{bmatrix} Y \\ Cr - 128 \\ Cb - 128 \end{bmatrix}$$

dss-E064

Figure 13-41 describes the computation for the calculation of the R component. The same computation applies for the G and B components:

Figure 13-41. Color Space Conversion Macro-Architecture



13.3.3.3.3 Hardware Cursor

The video layer can be used to display the hardware cursor. The encoded pixel data for the cursor image are in RGB12, RGB16 or RGB24 formats and the color space conversion block is bypassed. The transparency color key can be used when a non rectangle shape is used.

The alpha blending can be used to show a partial transparent cursor. When the alpha blender is enabled, the graphics layer is on top of the video layers. The cursor uses the graphics layer. The pixel alpha blending or the transparency color key can be used.

13.3.3.3.4 Up-/Down-Sampling

The video layer has a dedicated resizing block to upsample and downsample the video-encoded pixels. The supported input formats from memory are RGB24, RGB16, and YUV4:2:2

(RGB12 and all the alpha formats like ARGB and RGBA are not supported)

Users must set the right size and position of the original video before resizing for the upsampled/downsampled video to be inside the display screen boundaries.

The filtering applies on each component independently R, G, and B).

For the horizontal up/downsampling, the equation is R component with five taps):

$$\mathrm{Rout}(n) = \left(\sum_{i=-2}^{i=2} \mathrm{Ci}(\Phi) \times \mathrm{Rin}(n+i) \right) >> 7 \qquad (5)$$

dss-E066

For the vertical up/downsampling, the equation is R component with three taps):

$$\mathrm{Rout}(n) = \left(\sum_{i=-1}^{i=1} \mathrm{Ci}(\Phi) \times \mathrm{Rin}(n+i) \right) >> 7 \tag{6}$$

dss-E067

For the vertical up/downsampling, the equation is R component with five taps):

$$\mathrm{Rout}(n) = \left(\sum_{i=-2}^{i=2} \mathrm{Ci}(\Phi) \times \mathrm{Rin}(n+i) \right) >> 7$$

dss-E068

Rout: R component output

$\mathrm{Ci}(\Phi)$ dss-E069 : FIR filter coefficients

Rin: R component input (7)

The pixel (n + 1) is older than pixel (n). The line (n + 1) is older than line (n).

NOTE: The coefficients Ci() depend on the phase between input and output pixels.

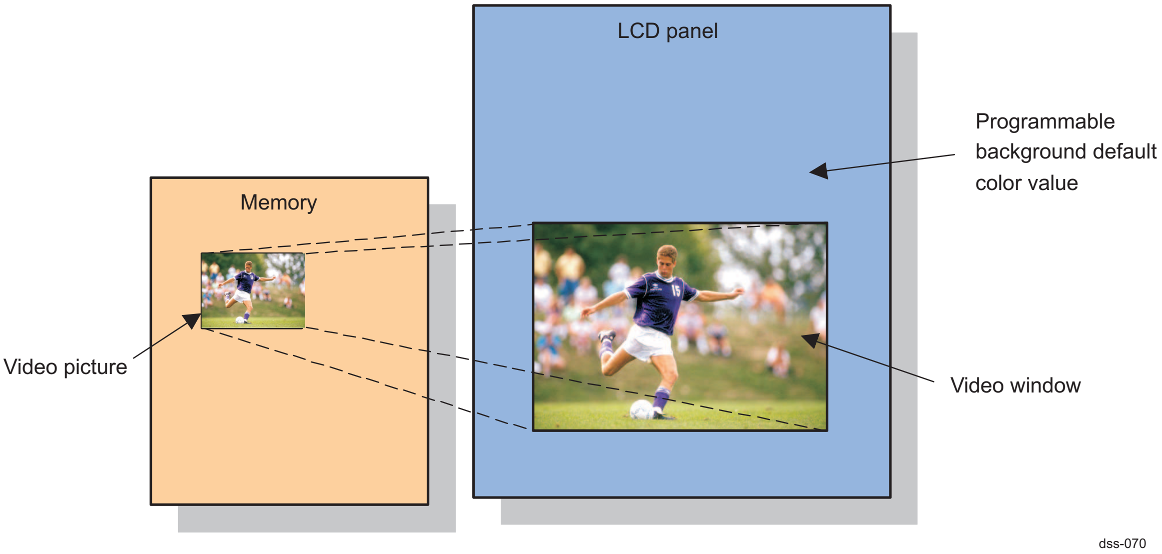
NOTE: If the 5-tap resizer is used for RGB16 and YUV4:2:2 picture formats, the width of the input picture must be a multiple of 2 pixels and more than 5 pixels:

DISPC_VIDn_ATTRS[21] VERTICAL_TAPS == 1

DISPC_VIDn_PICTURE_SIZE[10:0] VID_ORG_SIZE_X 4 and even

Figure 13-42 shows an example of video upsampling.

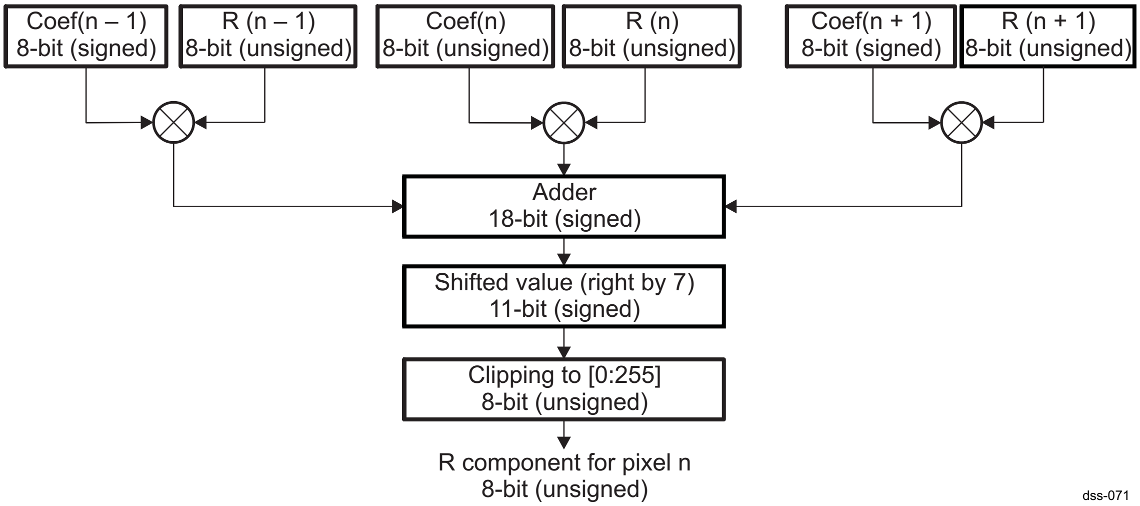
Figure 13-42. Video Upsampling



Filter Description

The up/downsampling filter is a poly-phase filter with five taps and eight phases for the horizontal filter and a programmable number of taps (three or five) and eight phases for vertical filter. The upsampling ratio is up to x8. The downsampling ratio using 3-tap configuration is/2. The downsampling ratio using 5-tap configuration is/4. The vertical filter is first applied to the encoded input pixel data; and then the horizontal filter is applied on the resulting pixel values to generate the output pixel values. Figure 13-43 shows the computation for the R component in the case of three coefficients (vertical filtering). The same computation applies to the G and B components.

Figure 13-43. Resampling Macro-Architecture (3-Coefficient Processing)



To determine if the minimum functional clock matches the down sampling ratio and the desired Pixel clock, the following formula must be used in conjunction with Table 13-12 and Table 13-13.

Ratio V when performing a vertical down-sampling only

$$h\_ratio = \frac{DISPC\_SIZE\_LCD.PPL}{DISPC\_VID\_SIZE.Vid\_Size\_X}$$

$$v\_ratio = \frac{DISPC\_VID\_PICTURE\_SIZE.Vid\_Org\_Size\_Y}{DISPC\_VID\_SIZE.Vid\_Size\_Y}$$

$$Ratio = \frac{v\_ration}{2 \times h\_ratio} \qquad If \quad 1 < v\_ratio \leq 2$$

$$Ratio = \max(\frac{v\_ration}{2 \times h\_ratio}, \frac{v\_ration - 2}{2 \times (h\_ratio - 1)}) \qquad If \quad 2 < v\_ratio \leq 4$$

dss_swpu108-E135

NOTE: For frequency ratio calculation on the TV output, it is correct to replace DISPC_SIZE_LCD with DISPC_SIZE_DIG.
When the down-sampling ratio is below 0.5, it is not possible to use a video in full screen.

Ratio H when performing a horizontal down-sampling only

$$Ratio = \frac{DISPC\_VID\_PICTURE\_SIZE.Vid\_Org\_Size\_X}{DISPC\_VID\_SIZE.Vid\_Size\_X}$$

dss_swpu108-E136

Ratio H+V when performing a horizontal and vertical down-sampling

Ratio = max (horizontal Ratio, vertical Ratio) as previously defined.

Table 13-12. Functional Clock Frequency Requirement in RGB16 YUV4:2:2—Active Matrix Display

Minimum Functional Clock (MHz)		Horizontal Resampling				
		Off	Up	1:1 – 1:2	1:2 – 1:3	1:3 – 1:4
Vertical Resampling	Off	AxPCLK	AxPCLK	2xPCLK	3xPCLK	4xPCLK
	Up	AxPCLK	AxPCLK	2xPCLK	3xPCLK	4xPCLK
	3-tap 1:1 to 1:2	2xPCLK	2xPCLK	4xPCLK	6xPCLK	8xPCLK
	5-tap 1:1 to 1:4	RatioxPCLK	RatioxPCLK	RatioxPCLK	RatioxPCLK	RatioxPCLK

With A = 1 in case all the data and synchronization signals are asserted and deasserted on the rising edge of the PCLK; otherwise, A = 2.

Table 13-13. Functional Clock Frequency Requirement in RGB24—Active Matrix Display

Minimum Functional Clock (MHz)		Horizontal Resampling				
		Off	Up	1:1 – 1:2	1:2 – 1:3	1:3 – 1:4
Vertical Resampling	Off	AxPCLK	AxPCLK	2xPCLK	3xPCLK	4xPCLK
	Up	AxPCLK	AxPCLK	2xPCLK	3xPCLK	4xPCLK
	3-tap 1:1 to 1:2	2xPCLK	2xPCLK	4xPCLK	6xPCLK	8xPCLK
	5-tap 1:1 to 1:4	RatioxPCLK	RatioxPCLK	2xRatioxPCLK	2xRatioxPCLK	2xRatioxPCLK

With A = 1 in case all the data and synchronization signals are asserted and deasserted on the rising edge of the PCLK; otherwise, A = 2.

Use case example:

An input picture of 1024*768 is scaled to an output picture of size of 800*600 and displayed onto a LCD of resolution1280*768 at a PCLK of 74.25 MHz with a DSS functional clock of 133 MHz.

In this example, a H+V down-sampling is done on the input picture. Firstly the Ratio V and H are determined and the resulting maximum value is taken to calculate the functional clock frequency required.

Ratio V: h_ratio = 1.6 and v_ratio = 1.28 then Ratio = 0.4

Ratio H: Ratio = 1.28

Ratio H+V: Ratio = max (1.28, 0.4) = 1.28

In this use case, the horizontal and vertical down sampling range are 1:1–1:2. The 3-tap or 5-tap configuration can be taken into consideration. Therefore, from Table 13-12 and Table 13-13, If in RGB16-YUV4:2:2:

- 3-taps → DSS functional clock = 4 * PCLK = 297 MHz
- 5-taps → DSS functional clock = Ratio * PCLK = 95.36 MHz

If in RGB24,

- 3-taps → DSS functional clock = 4 * PCLK = 297 MHz
- 5-taps → DSS functional clock = 2 * Ratio * PCLK = 190.72 MHz

In this use case, the pixel format supported is RGB16-YUV4:2:2 in a 5-tap configuration.

13.3.3.4 Overlay Support

CAUTION

Enabling overlay optimization (setting the DSS.DISPC_CTRL [12] OVLY_OPT bit) if no overlay region effectively exists (the DSS.DISPC_VIDn_ATTRS [0] EN bit is cleared, with n = 1, 2) leads to unpredictable behavior. The overlay optimization feature must be enabled only when an overlay area exists. Before enabling the overlay optimization, the DSS.DISPC_GFX_WINDOW_SKIP[31:0] GFX_WINDOW_SKIP bit field must be first set according to the video1 and graphics windows overlap.

The overlay mechanism consists of displaying more than one layer (graphics and video layers) using rules based on priority and transparency color keys.

When the pixel format is ARGB or RGBA, the color key match logic uses only the RGB value defined by ARGB or RGBA. The alpha blending factor is ignored.

The overlay managers are based on the same rules for priority and transparency color keys (see Figure 13-46).

Each data pipeline is assigned a single overlay related to a single display controller output.

The overlay manager is connected to all three outputs of the pipelines (graphics, video1 and video2). The output of the LCD overlay manger is connected to the Spatial/Temporal Dithering, and Passive Matrix units and back to the palette unit in the case of Gamma correction.

13.3.3.4.1 Priority Rule

The overlay manager can be configured in two distinct modes:

- Alpha mode (only source color key with the graphics layer)
- Normal mode (no alpha support)

The following rules apply in normal mode:

The video1 layer is always on top of the graphics layer. The video2 layer is always on top of the video1 and graphics. The display controller reads the data for each buffer from the system memory and, depending on the transparency color key values, displays either the pixels in the video layer, the pixels in the graphics layer, or the solid background color.

Each layer can have any size up to full-display screen. If there are no graphics or video-encoded pixels at a specific position, the programmable, solid background color appears (see Figure 13-48).

Figure 13-44. Overlay Manager in Normal Mode

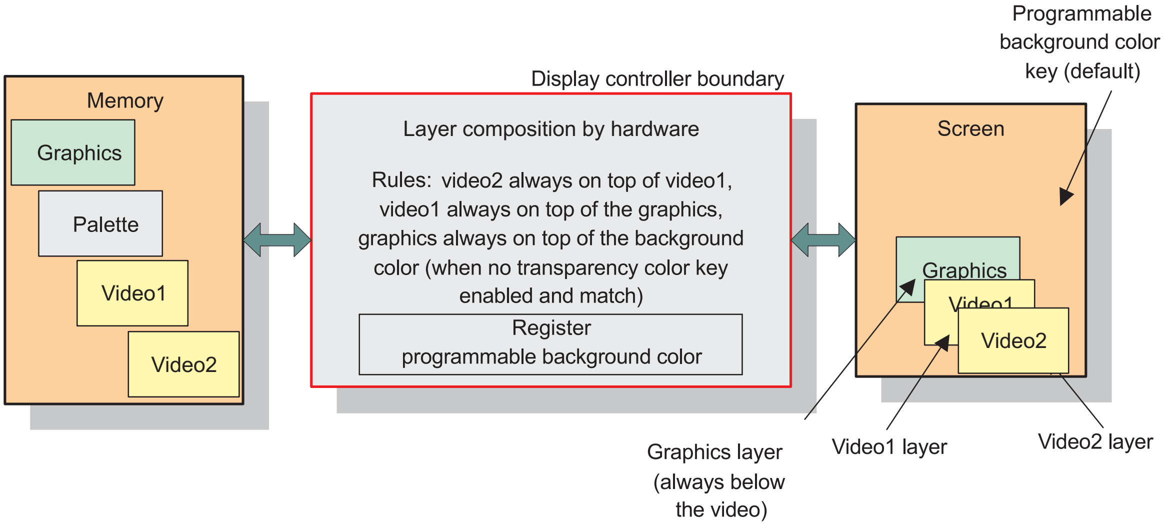
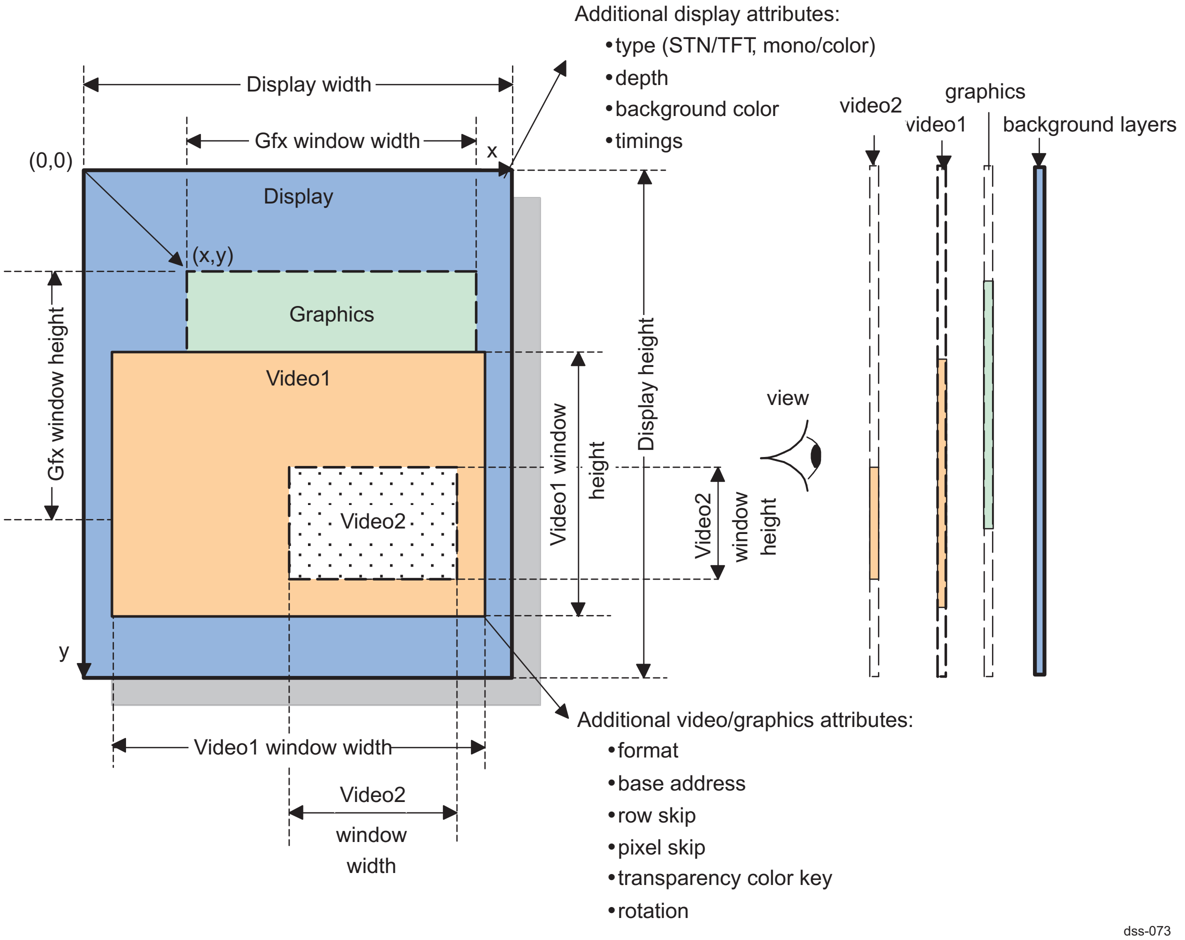


Figure 13-45. Display Attributes in Normal Mode



The following rules apply in alpha mode:

The video2 layer is always on top of the video1 layer. The graphics layer is always on top of the video1 and video2. The display controller reads the data for each buffer from the system memory and, depending on the transparency color key values, displays either the pixels in the video layer, the pixels in the graphics layer, or the solid background color.

Each layer can have any size up to full-display screen. If there are no graphics or video-encoded pixels at a specific position, the programmable, solid background color appears (see Figure 13-48).

Figure 13-46. Overlay Manager in Alpha Mode

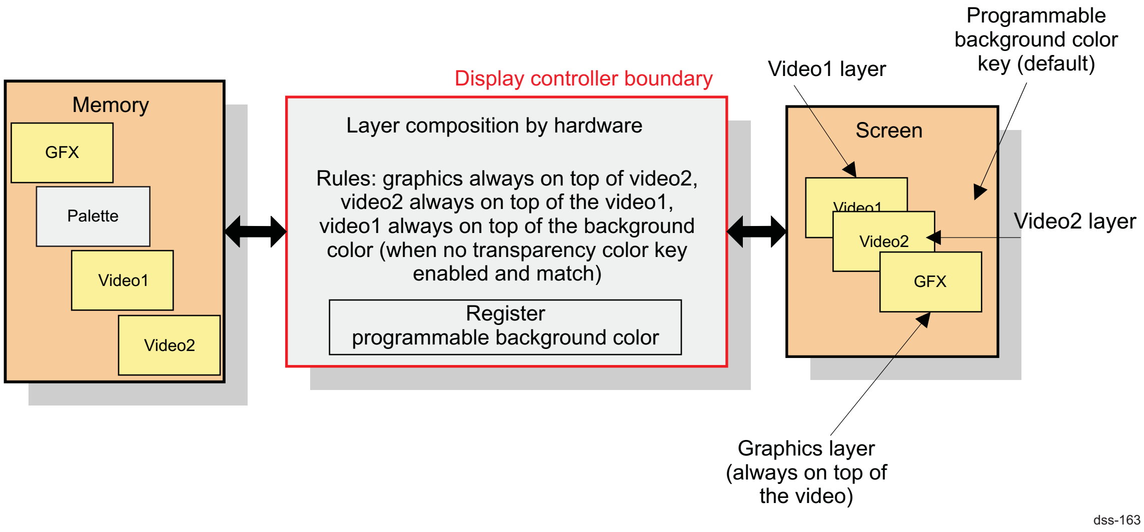


Figure 13-47. Display Attributes in Alpha Mode

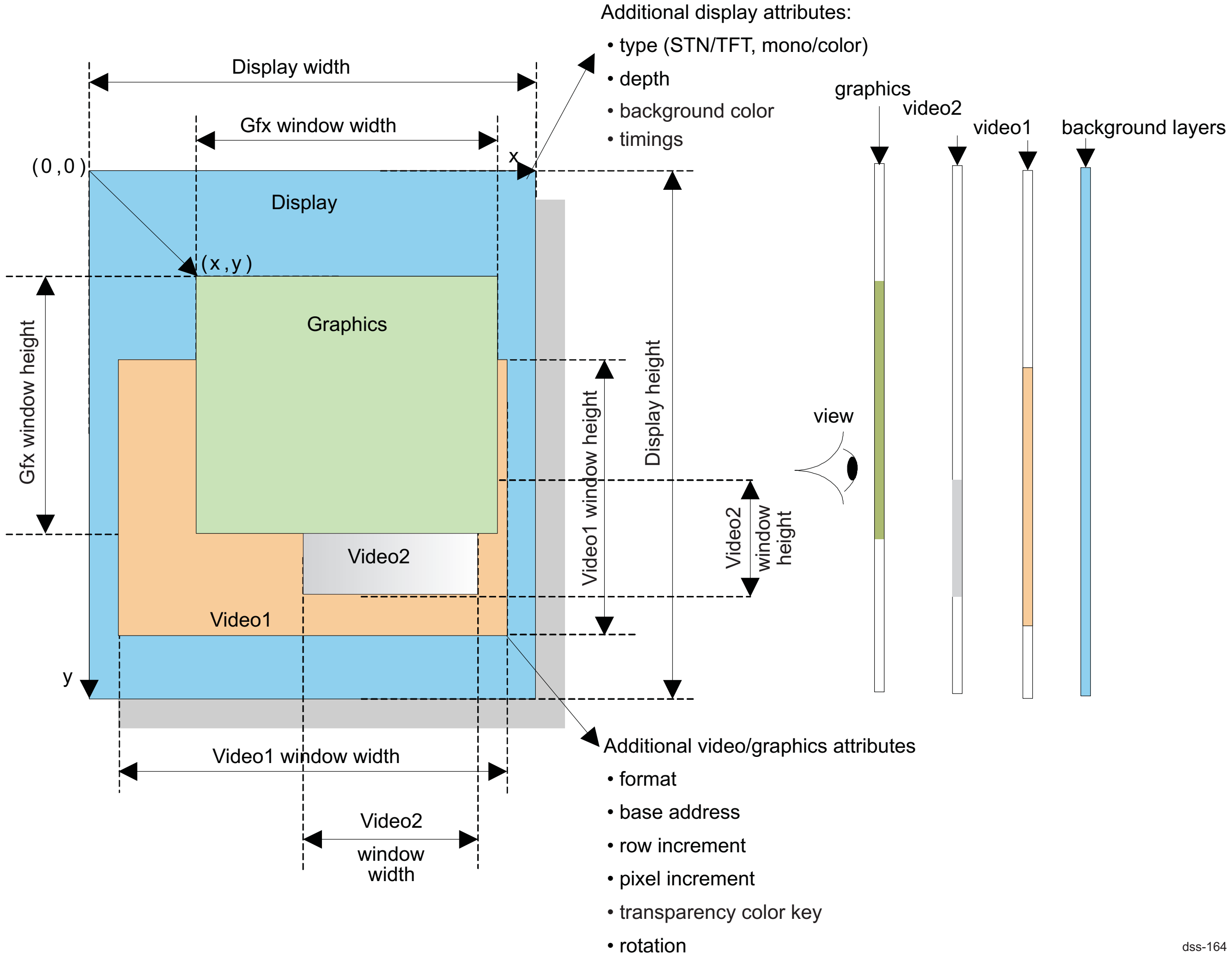
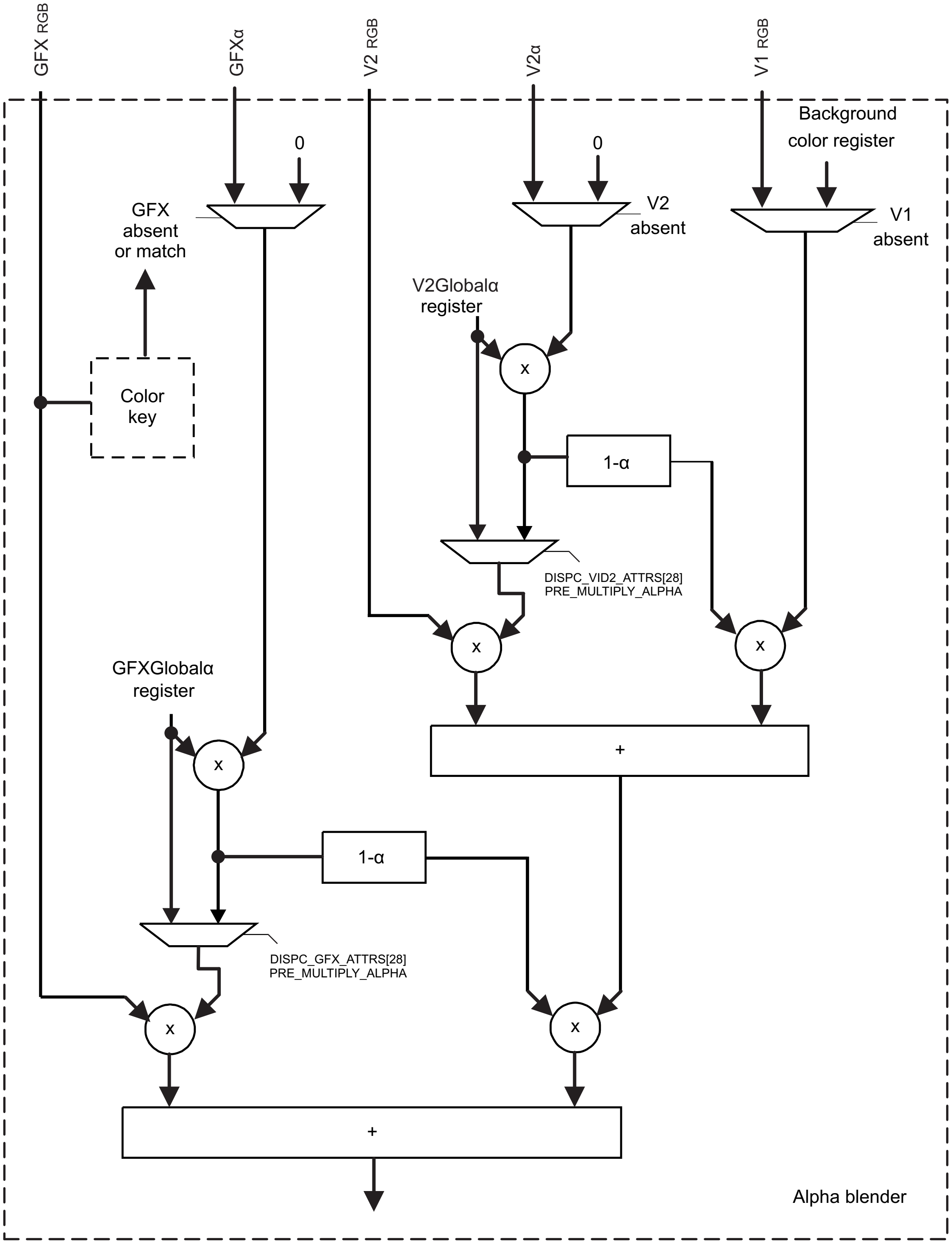


Figure 13-48 shows the alpha blending processing in detail.

Figure 13-48. Alpha Blending Macro Architecture with Pre-multiplied Alpha Support



NOTE: "1-alpha" operator corresponds to the basic 1's complement operation.

The pre-multiplied alpha option is accessible through DSS.DISPC_GFX_ATTRS[28] PRE_MULTIPLY_ALPHA and DSS.DISPC_VIDn_ATTRS[28] PRE_MULTIPLY_ALPHA registers bits. The following settings are available:

- PRE_MULTIPLY_ALPHA bit = '0' : Source is not pre-multiplied with alpha. Full blending is done in the

DISPC.

- PRE_MULTIPLY_ALPHA bit = '1' : Source is pre-multiplied with alpha. Partial blending is done.

NOTE: The pre-multiplied alpha option is only valid when bit fields DSS.DISPC_GFX_ATTRS[4:1] GFX_FMT and DSS.DISPC_VIDn_ATTRS[4:1] VID_FMT, respectively, are set to ARGB or RGBA formats. Otherwise, the PRE_MULTIPLY_ALPHA bit fields are ignored by the hardware.

The alpha blending value is defined by the pixel value (ARGB or RGBA formats). A global alpha blending value can be defined and used in combination with the pixel alpha blending value. If the pixel format contains no alpha blending value, the pixel alpha value is considered to be 0xFF.

In case of ARGB-444, the alpha blending is defined using a 4-bit value. It is converted into an 8-bit value by duplicating the 4-bit value. Table 13-14 details the alpha blending 4-bit values and the corresponding blending percentage.

Table 13-14. Alpha Blending 4-Bit Values

Alpha Blending 4-Bit Value (ARGB-444)	Alpha Blending 8-Bit Value (Converted Value)	% Blending
0x0	0x00	100% (transparent)
0x1	0x11	93.33%
0x2	0x22	86.6%
...	...	...
0xE	0xEE	6.6%
0xF	0xFF	0% (opaque)

13.3.3.4.2 Transparency Color Keys

13.3.3.4.2.1 Normal Mode

This section describes the features available in normal mode.

The two transparency color keys are the video source transparency color key and the graphics destination transparency color key. The encoded pixel color value is compared to the transparency color key. For CLUT bitmaps, the palette index is compared to the transparency color key and not to the palette value pointed out by the palette index.

NOTE: The video source transparency color key and graphics destination transparency color key cannot be active at the same time.

- Video source transparency color key value:

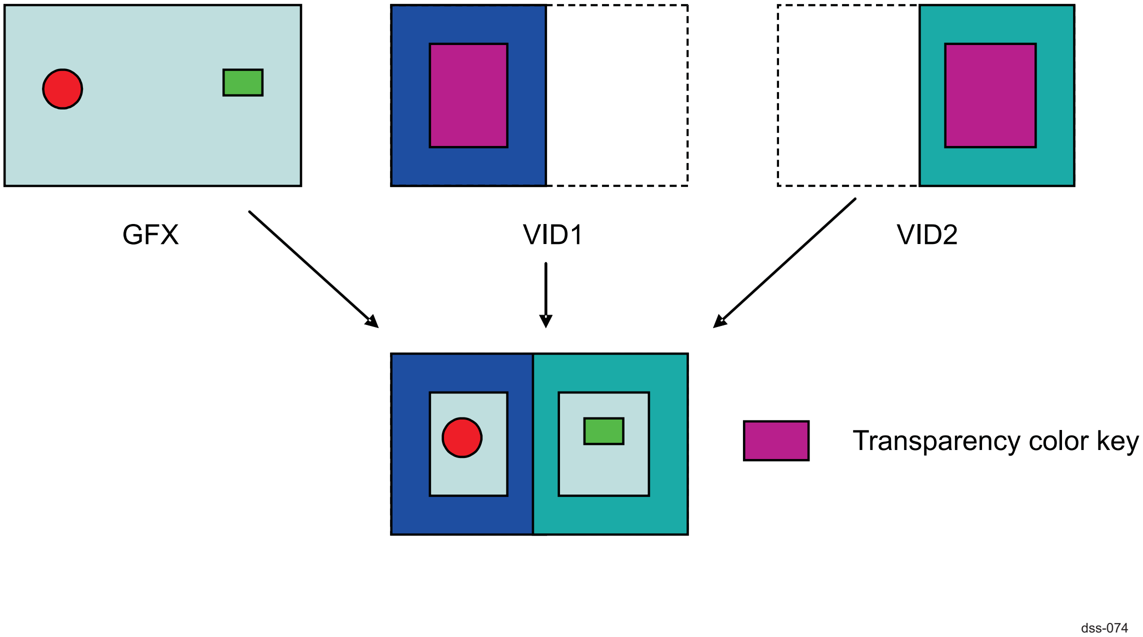
  The video source transparency color key value defines the encoded pixel data considered as the transparent pixel. The encoded pixel values with the source color key value are pixels not visible on the screen, and the underlayer encoded pixel values or solid background color are visible.

  The video source transparency color key can be used only if the color space conversion and the up/down-scaling modules are disabled. The format of the data is RGB 16. (This feature handles the hardware cursor displayed by one of the video layers.)

  To enable the video source transparency color key, set to 0x1 the DSS.DISPC_CFG[11] TCK_LCD_SELECTION bit for LCD output or the DSS.DISPC_CFG[13] TCK_DIG_SELECTION bit for digital output. Program the DSS.DISPC_CFG[10] TCK_LCD_EN bit (LCD output) or the DSS.DISPC_CFG[12] TCK_DIG_EN bit (digital output) to enable or disable the transparency color key.

  An example is shown in Figure 13-49: The video source transparency is applied on video1 (VID1) and video2 (VID2) layers. The pixels with the transparency color key are not displayed; instead, underlying layers are shown.

Figure 13-49. Video Source Transparency Example



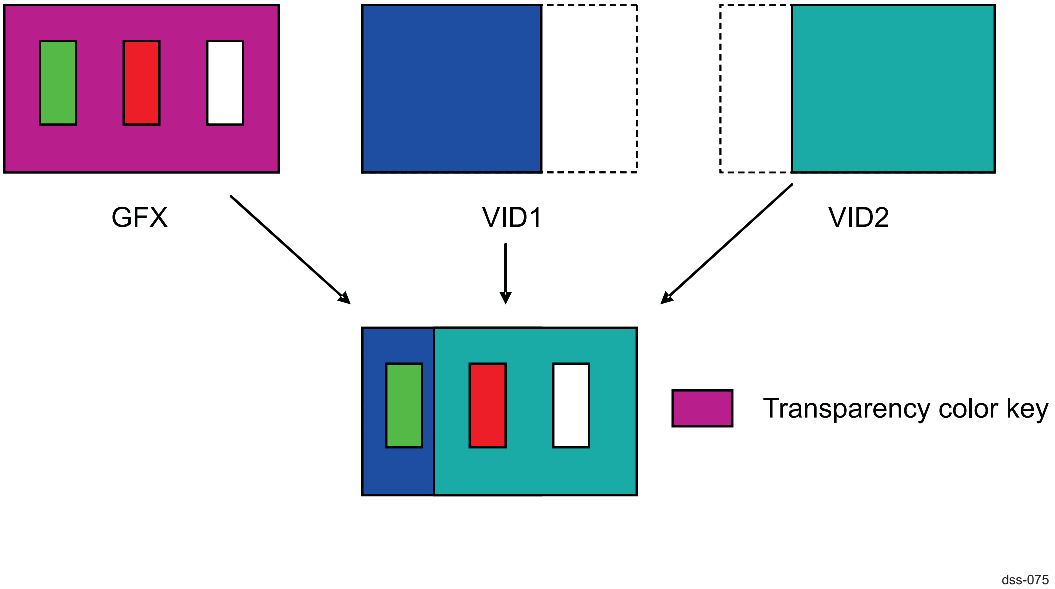
- Graphics destination transparency color key value:

  The graphics destination transparency color key value defines the encoded pixels in the video layers to be displayed. The encoded pixel values with the destination color key value are pixels not visible on the screen and the pixels different from the transparency color key are displayed over the video layers. The destination transparency color key is applicable only in the graphics region when graphics and video overlap; otherwise, the destination transparency color key is ignored.

  To enable the graphics destination transparency color key, set to 0x0 the DSS.DISPC_CFG[11] TCK_LCD_SELECTION bit for LCD output or the DSS.DISPC_CFG[13] TCK_DIG_SELECTION bit for digital output. Program the DSS.DISPC_CFG[10] TCK_LCD_EN bit (LCD output) or the DSS.DISPC_CFG[12] TCK_DIG_EN bit (digital output) to enable or disable the transparency color key.

  An example is shown in Figure 13-50: The destination transparency is applied on graphics (GFX) layer and the pixels without the transparency color key are displayed over the overlying layers.

Figure 13-50. Graphics Destination Transparency Example



13.3.3.4.2.2 Alpha Mode

This section describes the features available in alpha mode.

Only the graphics source transparency color key is available The encoded graphics pixel color value is compared to the transparency color key. The encoded pixel values with the source transparency key are not visible and the under-layer encoded pixel values or solid background color are visible. To enable the graphics source transparency color key, set to 0x0 the DSS.DISPC_CFG[11] TCK_LCD_SELECTION bit for LCD output or the DSS.DISPC_CFG[13] TCK_DIG_SELECTION bit for digital output. Program the DSS.DISPC_CFG[10] TCK_LCD_EN bit (LCD output) or the DSS.DISPC_CFG[12] TCK_DIG_EN bit (digital output) to enable or disable the transparency color key. In the case of CLUT bit maps, the palette index is compared to the transparency color key and not the palette value pointed out by the palette index.

13.3.3.4.3 Overlay Optimization (Only Available in Normal Mode)

The display controller can be configured to take advantage of the fact that the graphics pixels under video window 1 are not visible when the transparency color key is not used. The optimization can be selected to reduce the bandwidth used to fetch the pixels for graphics. The color key must be disabled. The graphics pixels under the video window 1 are not fetched from system memory. At least the video window 1 and the graphics window must be enabled. The following graphic formats are supported: RGB (RGB16 and RGB24 packed and unpacked), YUV4:2:2, and BITMAP 8. The formats BITMAP 1, 2, and 4 are not supported. The video format can be RGB (RGB16, RGB24 packed and unpacked, and YUV4:2:2 formats). The DMA engine does not fetch the unnecessary graphics pixels to avoid extra bandwidth use. Only visible pixels from graphics and video buffers in system memory are fetched and displayed by the display controller.

13.3.3.5 Active/Passive Matrix Display Data Path

For active matrix display data path, the following blocks are serial and each of them can be bypassed:

- Color phase rotation
- Spatial/temporal dithering
- Multiple cycle data format

For passive matrix display data path, the following blocks are serial and each of them can be bypassed:

- Color phase rotation
- Spatial/temporal dithering
- Passive matrix technology

13.3.3.5.1 Color Phase Rotation

The Color Phase Rotation (CPR) can be used to correct the LCD output colorimetry in case of non pure white backlight.

The color phase rotation can be selected for passive matrix and active matrix panel. The logic is integrated after the LCD overlay manager or the palette while using the gamma correction and before the spatial/temporal dithering. The color phase rotation can be selected to correct the nonpure white backlight of the LCD module by using a programmable matrix to convert the 24-bit RGB pixel value into a new 24-bit RGB pixel value. The matrix is programmed through a set of nine 10-bit signed coefficients. The output of the calculation is clipped to [0:255]. The color phase rotation is processed by the equation shown in Figure 13-51.

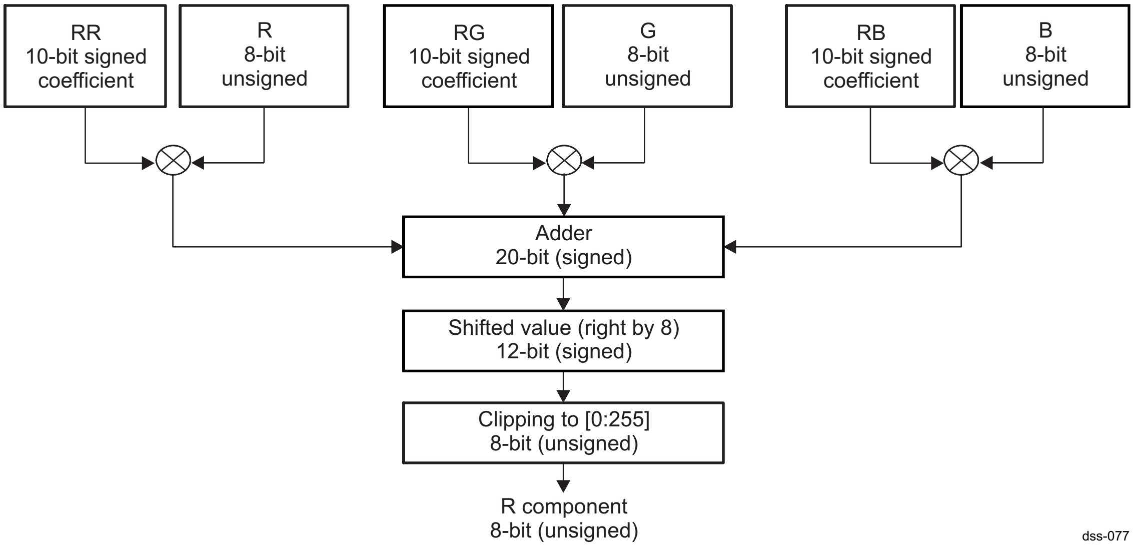
Figure 13-51. Color Phase Rotation Matrix

$$\begin{bmatrix} Rout \\ Gout \\ Bout \end{bmatrix} = \frac{1}{256} * \left(\begin{bmatrix} RR & RG & RB \\ GR & GG & GB \\ BR & BG & BB \end{bmatrix} * \begin{bmatrix} Rin \\ Gin \\ Bin \end{bmatrix} \right)$$

dss-E076

Figure 13-52 shows the color phase rotation macro-architecture.

Figure 13-52. Color Phase Rotation Macro Architecture



13.3.3.5.1.1 Spatial/Temporal Dithering

The spatial/temporal dithering logic can be selected for passive matrix and active matrix panel. The dithering logic is integrated after the color phase rotation and before the TDM and passive matrix units. The spatial/temporal dithering logic can be selected to enhance the quality of the passive matrix and active matrix outputs. The dithering logic can process the pixels over a single frame, two frames, or four frames. In the case of a single frame, only spatial processing is applied. In the case of multiple frames, spatial and temporal processing is applied to the pixels.

- Passive Matrix Technology: The passive matrix display dithering logic path is used. The spatial/temporal dithering logic can be selected. When selected, the pixels are preprocessed by the spatial/temporal dithering logic before the passive matrix display dithering logic. The output format of the spatial/temporal dithering logic is RGB 12-bit (not configurable).
- Active Matrix Technology: The encoded pixel values are used by spatial/temporal dithering logic to display the data in a lower color depth on the LCD panel. The spatial/temporal dithering algorithm is based on the (x,y) pixel position, the value of removed bits and the frame number. The picture quality is improved when enabling the spatial/temporal dithering logic. When spatial/temporal dithering is not enabled, the three MSBs of the pixel color components are output on the interface data bus if the interface data bus is smaller than the pixel format size. If the interface data bus is wider than the pixel format size, by programming the pixel components replication active/inactive, the MSB is replicated to the LSB of the interface data bus or the LSB is filled with 0s.

13.3.3.5.2 Passive Matrix Display Dithering Logic

- Passive matrix technology

  After the graphics data are merged with the video data from the video layers depending on the transparency status, the result is sent to the color/grayscale space-/time-based dither generator. The monochrome data and each RGB color component are encoded on 4 bits, which are the 4 MSBs of the pixel-encoded component 8-bit value defined by the merge of the graphics data and the video data.

  These 4-bit values are used to select on the 16 intensity levels. The gray/color intensity is controlled by turning individual pixels on and off at varying period rates, making the average time the pixel is off longer than the average time the pixel is on, thus producing more intense grays/colors. The dithering generator also uses the intensity of adjacent pixels in the calculation to give the screen image a smooth appearance. The proprietary dither algorithm is optimized to provide a range of intensity values that matches the visual perception of color/gray graduations.
- Active matrix technology

The passive matrix dithering logic is always bypassed in active displays.

NOTE: If the interface data bus is smaller than the pixel format size, dithering logic can be enabled. If the interface data bus is wider than the pixel format size, the dithering logic cannot be enabled and replication feature can be used.

13.3.3.5.3 Passive Matrix Display Output FIFO

- Passive matrix technology

  The display controller contains a 2-entry by 8-bit-wide output FIFO used to store pixel data before it is driven out to the LCD pins. Each time a modulated pixel value is output from the dither generator, it is placed into a serial shifter. The shifter can be configured to be 4 or 8 bits wide. Single-panel monochrome screens use either four or eight data lines; single-panel color screens use eight data pins.
- Active matrix technology

  The output FIFO is bypassed in active matrix mode.

13.3.3.5.4 Multiple Cycle Output Format

The pixels after the active matrix display processing are formatted on one or multiple cycles (from one to three cycles). The interface width can be 8-, 9-, 12-, or 16-bit. On three cycles, two pixels can concatenate and send to the panel. When the TDM is disabled, the display controller outputs the pixels using the conventional formats: Passive matrix display/active matrix display monochrome/color.

The following example shows an output configuration based on the interface width (8-bit) and the pixel format output (24-bit) (also see Table 13-15):

- The DSS.DISPC_CTRL[24:23] TDM_CYCLE_FMT bit field is set to 0x2 (three cycles for one pixel).
- The DSS.DISPC_DATA_CYCLEk (k=0) register is set to 0x00000008 (8 bits from pixel 1 for the first cycle).
- The DSS.DISPC_DATA_CYCLEk (k=1) register is set to 0x00000008 (8 bits from pixel 1 for the second cycle).
- The DSS.DISPC_DATA_CYCLEk (k=2) register is set to 0x00000008 (8 bits from pixel 1 for the third cycle).

Table 13-15. 8-Bit Interface Configuration/24-Bit Mode

	24-Bit Mode		
	1st Cycle	**2nd Cycle**	**3rd Cycle**
Data[7]	R0[7]	G0[7]	B0[7]
Data[6]	R0[6]	G0[6]	B0[6]
Data[5]	R0[5]	G0[5]	B0[5]
Data[4]	R0[4]	G0[4]	B0[4]
Data[3]	R0[3]	G0[3]	B0[3]
Data[2]	R0[2]	G0[2]	B0[2]
Data[1]	R0[1]	G0[1]	B0[1]
Data[0]	R0[0]	G0[0]	B0[0]

13.3.3.6 Video Line Buffer

The line buffer size is 1024 x 24-bit. There are six line buffers (1024 x 24-bit) that can be merged into three lines (2048 x 24-bit). Table 13-16 lists the maximum width depending on the TAP configuration and the pixel format.

Table 13-16. Maximum Width Allowed

Vertical Tap	Pixel Format	Maximum Width (Pixels)
3	RGB16 RGB24 YUV4:2:2	2048
5	RGB16 RGB24 YUV4:2:2	1024

13.3.3.7 Synchronized Buffer Update

A synchronization mismatch between the frame buffer and the display refreshes, named tearing effect, can lead to images that appear to be stretched on the screen. To avoid this, a synchronization mechanism is needed between the display controller and the process that updates the buffer. An interrupt is generated when the display reaches a predefined line number. This PROGRAMMEDLINENUMBER interrupt is a level signal and stays active during the programmed line of the display.

13.3.3.8 Multiple Buffer Support

Users update the base address of the buffer when the update of the working buffer has finished and is ready to be displayed. The register that contains the base address of the buffer is a shadow register that is read by the hardware at the next Vertical Front Porch (VFP).

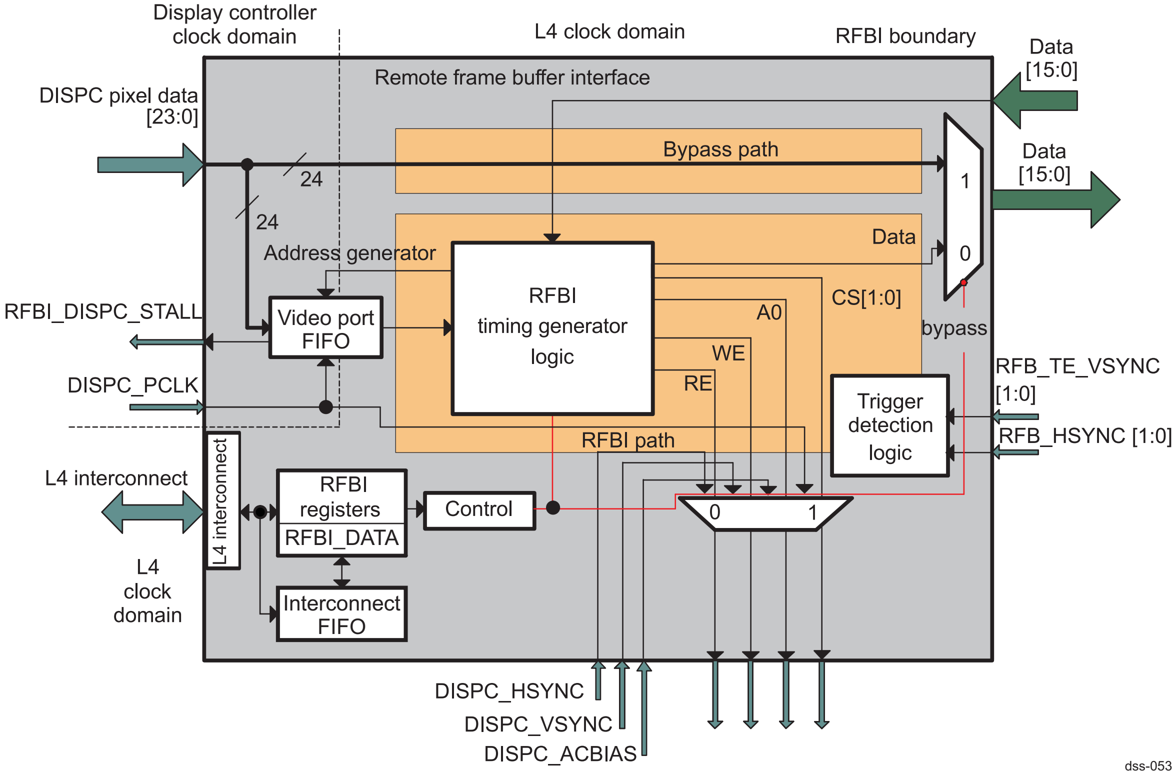
13.3.4 RFBI Functionalities

The RFBI module can capture the output pixel from the display controller and send the data to the RFB in the LCD panel. The application configures the RFBI module, sends commands, reads data, and configures the display controller to send data fetched from the system memory by the display controller DMA engine. The commands/data are sent using an 8-, 9-, 12-, or 16-bit parallel interface.

The display controller is configured to send the data in 12-, 16-, 18-, or 24-BPP format. In the video port FIFO, the encoded pixel values are in an LSB alignment independently of the endianness in system memory.

Figure 13-53 shows an overview of the RFBI architecture.

Figure 13-53. RFBI Architecture Overview



13.3.4.1 RFBI FIFO

The input video port FIFO receives data from the display controller at the pixel clock. The data in the video port FIFO are read by the RFBI and are sent to the LCD panel. The video port FIFO is 24 bits wide and each pixel in 12-, 16-, 18-, and 24-BPP format is stored in the video port FIFO using one 24-bit value aligned on the 24-bit LSB. Section 13.3.4.4, *Output Parallel Modes*, shows an example of an output configuration based on the interface width (16 bits) and the pixel format output (24 bits).

13.3.4.2 RFBI Interconnect FIFO

The interconnect FIFO receives the data from RFBI_DATA write requests to the L4 interconnect slave port. The data in the interconnect FIFO are read by the RFBI and sent to the LCD panel. The width of the interconnect FIFO is 32 bits. The size of the interconnect FIFO is 24 words of 32 bits (that is, 24 words of RFBI_DATA).

13.3.4.3 Input Pixel Formats

The supported pixel formats in the RFBI module are: RGB24-888, RGB18-666, RGB16-565, and RGB12-444 as output from the display controller and from the L4 (for writing parameters). In both cases, the pixels are formatted in accordance with the configuration of the output interfaces (multiple cycles).

13.3.4.4 Output Parallel Modes

The RFBI output modes are 8-, 9-, 12-, and 16-bit interfaces. Any mode can be selected regardless of the pixel format. Set the right configuration in the cycle registers to define a valid configuration for each output cycle.

The following example is an output configuration based on the 16-bit interface width and the 24-bit pixel format (i = 0 or 1) (see also Table 13-17):

- The DSS.RFBI_CONFIG_i[10:9] CYCLE_FMT bit field is set to 0x3 (three cycles for two pixels).

- The DSS.RFBI_DATA_CYCLE1_i register is set to 0x00000010 (16 bits from pixel 1 for the first cycle).
- The DSS.RFBI_DATA_CYCLE2_i register is set to 0x00080808 (8 bits from pixel 1 and pixel 2 and alignment of 8 bits from pixel 2 for the second cycle).
- The DSS.RFBI_DATA_CYCLE3_i register is set to 0x00100000 (16 bits from pixel 2 for the third cycle).

Table 13-17. 16-Bit Interface Configuration/24-Bit Mode

	24-Bit Mode		
	1st Cycle	**2nd Cycle**	**3rd Cycle**
Data[15]	R0[7]	B0[7]	G1[7]
Data[14]	R0[6]	B0[6]	G1[6]
Data[13]	R0[5]	B0[5]	G1[5]
Data[12]	R0[4]	B0[4]	G1[4]
Data[11]	R0[3]	B0[3]	G1[3]
Data[10]	R0[2]	B0[2]	G1[2]
Data[9]	R0[1]	B0[1]	G1[1]
Data[8]	R0[0]	B0[0]	G1[0]
Data[7]	G0[7]	R1[7]	B1[7]
Data[6]	G0[6]	R1[6]	B1[6]
Data[5]	G0[5]	R1[5]	B1[5]
Data[4]	G0[4]	R1[4]	B1[4]
Data[3]	G0[3]	R1[3]	B1[3]
Data[2]	G0[2]	R1[2]	B1[2]
Data[1]	G0[1]	R1[1]	B1[1]
Data[0]	G0[0]	R1[0]	B1[0]

13.3.4.5 Unmodified Bits

In a cycle, if every bit in the interface does not have a pixel value, the status of the unused bits can be programmed to be 0, 1, or the previous value (I/O power consumption optimization).

13.3.4.6 Bypass Mode

In bypass mode, the RFBI path is bypassed and the display controller data and signals are sent directly to the output interface of the RFBI.

13.3.4.7 Send Commands

The commands are written through the L4 interconnect and into the DSS.RFBI_CMD register. After a command is sent, another one can be accepted by the module and set. If the processing of a command is not complete, the MPU access to change the command stalls.

13.3.4.8 Read/Write

Depending on the status of A0, WE, and RE, the commands and display/parameter data are written to the panel (handled by the state-machine for the commands/parameter data and stored in memory for the display data), or the display data/status values are read from the LCD panel (status and display data in the LCD panel memory). The polarity of A0 (RFBI_A0 signal), WE (RFBI_WR signal), RE (RFBI_RD signal), and CSx (RFBI_CSx signal, with x = 0, 1) is programmable.

Table 13-18 describes the read/write function.

Table 13-18. Read/Write Function Description

A0 (RFBI_A0)	WE (RFBI_WR)	RE (RFBI_RD)	Function Description
1	0	1	Display data write, parameter data write
1	1	0	Display data read
0	1	0	Status read
0	0	1	Command data write

A minimum of RFBI_Cs cycle time, as defined in Table 13-19, is required to keep the RFBI_CSx signal asserted between write transfers of multiple pixels.

Table 13-19 indicates the minimum cycle time for RFBI_CSx, depending on the source of pixels (display controller or L4 interconnect slave port) and the cycle format (1pixel/cycle, 1 pixel/2 cycles, 1 pixel/3 cycles, or 2 pixels/3 cycles).

Table 13-19. Minimum Cycle Time for CSx/WE Always Asserted

RFBI Performance	RFBI_CONFIG_i[10:9] CYCLE_FMT	RFBI_CONFIG_i[8:7] L4_FMT	Minimum Cycle Time (in Number of L4 Cycles)
L4 interconnect	1 pixel/cycle	1 pixel	5
	1 pixel/2 cycles	1 pixel	4
	1 pixel/3 cycles	1 pixel	4
	2 pixels/3 cycles	1 pixel	6
	1 pixel/cycle	2 pixels	4
	1 pixel/2 cycles	2 pixels	4
	1 pixel/3 cycles	2 pixels	4
	2 pixels/3 cycles	2 pixels	6
Display Controller	1 pixel/cycle	N/A	4
	1 pixel/2 cycles	N/A	3
	1 pixel/3 cycles	N/A	3
	2 pixels/3 cycles	N/A	6

13.3.5 Hardware Requests

13.3.5.1 Display Controller DMA Request (Line Trigger)

One DMA synchronization line (DSS_LINE_TRIGGER) is connected to the sDMA by the sDMA controller (S_DMA_5) input line. This DMA request is not a classical one but a synchronization signal from the display subsystem to the sDMA informing the sDMA that a programmable number of lines are output to the LCD, and that the system memory can be updated. This allows the sDMA channel to be synchronized with the display subsystem internal DMA controller. In other words, it allows to synchronize a memory to memory frame buffer update based on the scan line of the frame buffer in system memory (SDRAM or SRAM) by the display controller. The DSS_LINE_TRIGGER DMA request is generated at a programmable line number defined in DSS.DISPC_LINE_NUMBER[10:0] LINE_NUMBER bit field.

13.3.5.2 RFBI DMA Request

The RFBI_DMA_REQ is used to receive data into the RFBI FIFO. The DMA request is always generated when there is enough room in the FIFO to accept the full burst.

13.3.5.3 DISPC Interrupt Request

The interrupt line indicates when one or more events are detected by the hardware. Each event is independently maskable by setting the DSS.DISPC_IRQEN register.

To check when a particular interrupt event occurs and to reset a particular event, the DSS.DISPC_IRQSTS register must be accessed. This register regroups all the status of the module internal events that generate an interrupt (read 0: No interrupt occurred; read 1: Interrupt occurred; write 1: Status bit reset). See the registers in Section 13.6 for more information on checking and clearing interrupt events. Table 13-20 lists the display subsystem interrupt events.

Table 13-20. Display Subsystem Interrupts

Interrupt Name	Description
FRAMEDONE	Active frame is complete and LCD output is disabled.
VSYNC	VSYNC interrupt occurred at the end of the frame.
EVSYNC_EVEN(1)	EVSYNC_EVEN interrupt occurred at the end of the frame. (EVSYNC is received and the field polarity is even.)
EVSYNC_ODD(1)	EVSYNC_ODD interrupt occurred at the end of the frame. (EVSYNC is received and the field polarity is odd.)
ACBIASCOUNTSTATUS	The ac-bias transition counter decremented to 0.
PROGRAMMEDLINENUMBER	The LCD reached the user-programmed line number.
GFXFIFOUNDERFLOW	The input graphics FIFO goes underflow.
GFXENDWINDOW	The screen reached the end of the graphics window. All data for the graphics window are fetched from memory and displayed on the screen.
PALETTEGAMMALOADING	The palette/gamma table is loaded.
OCPERROR	L3 interconnect sent SResp = ERR.
VID1FIFOUNDERFLOW	The input video1 FIFO goes underflow.
VID1ENDWINDOW	The screen reached the end of video1 window. All data for the video window are fetched from the memory and displayed on the screen.
VID2FIFOUNDERFLOW	The input video2 FIFO goes underflow.
VID2ENDWINDOW	The screen reached the end of video2 window. All data for the video window are fetched from the memory and displayed on the screen.
SYNCLOST	Interrupt occurs when VSYNC width/front or back porches are not wide enough to load the pipelines with data (LCD output).
SYNC_LOST_DIGITAL	Interrupt occurs when the display controller is not ready to output data when a digital request occurs. This interrupt informs that the timings of the NTSC/PAL video encoder are not set correctly.
WAKEUP	Occurs when the wakeup signal is asserted

(1) EVYNC interrupts (EVSYNC_EVEN and EVSYNC_ODD) are external interrupts received by the display controller and generated by the video encoder (VENC) module.

NOTE: To clear a synchronization lost interrupt, follow this sequence:

1. Clear the DSS.DISPC_CTRL[0] LCD_EN (LCD: SYNCLOST interrupt) or DSS.DISPC_CTRL[1] DIGITAL_EN (TV: SYNC_LOST_DIGITAL interrupt) bits.

   Check the interrupts.

   LCD: Verify that a FRAMEDONE interrupt occurs.

   TV : Verify that EVSYNC_EVEN or EVSYNC_ODD interrupts occur.
2. Set the DSS.DSS_SYSCONFIG[1] SOFTRESET bit to reset the display subsystem.
3. Set the display subsystem registers again.

NOTE: The SYNC_LOST_DIGITAL interrupts, which occur before the first VSYNC pulse signal (from the video encoder), must not be considered.

After the first VSYNC pulse signal, the SYNC_LOST_DIGITAL interrupt status bit must be cleared by writing 1 in the DSS.DISPC_IRQSTS[15] SYNC_LOST_DIGITAL bit; then the SYNC_LOST_DIGITAL interrupt can be enabled by setting the DSS.DISPC_IRQEN[15] SYNC_LOST_DIGITAL bit.

13.4 Programming Model

This section describes how to configure the display subsystem for the desired functionalities and also describes the programming models of the display controller, the RFBI and the video encoder.

The main configuration scenarios are:

- LCD panel support (bypass or RFBI mode)

  Configure the RFBI module (only if in RFBI mode; otherwise, the default values must remain), and then configure the display controller to the desired functionalities before the activities start.
- TV set support

  Configure the video encoder and then the display controller.
- Both LCD panel support (bypass or RFBI mode) and TV set support

  Configure the RFBI module (only if in RFBI mode; otherwise, leave the default values), configure the video encoder, and then configure the display controller.

13.4.1 Display Subsystem Reset

The display subsystem can receive a software reset that is propagated through all of the submodules to initialize the subsystem. The following procedure describes a possible sequence:

1. If the LCD is on, stop the LCD by setting the DSS.DISPC_CTRL[0] LCD_EN bit to 0.
   a. Reset the frame done status bit by writing 1 in the DSS.DISPC_IRQSTS[0] FRMDONE bit.
   b. Wait until the DSS.DISPC_IRQSTS[0] FRMDONE bit is set to 1. This shows that the end of frame has taken place and the LCD stop is complete.
2. To take the display subsystem out of reset, all clocks related to the display subsystem must be enabled and the DPLL4 must be enabled. The following clocks must be enabled to take the display subsystem out of reset:
   - PRCM.CM_FCLKEN_DSS[0] EN_DSS1 bit set to 1
   - PRCM.CM_FCLKEN_DSS[1] EN_DSS2 bit set to 1
   - PRCM.CM_FCLKEN_DSS[2] EN_TV bit set to 1
   - PRCM.CM_ICLKEN_DSS[0] EN_DSS bit set to 1

   Once the clocks are enabled as shown, the display subsystem can be taken out of reset.
3. Write 1 in the DSS.DSS_SYSCONFIG[1] SOFTRESET bit to apply the soft reset to the subsystem.
4. Read the DSS.DSS_SYSSTS[0] RESETDONE bit. If this bit is 1, the reset sequence is complete; otherwise, read this bit again (the reset sequence is not completed).

13.4.2 Display Subsystem Configuration Phase

The display subsystem configuration phase is important to configure the data flow for using the LCD panel or the TV set. Use the following flow:

1. To configure the top level of the functional clock of the display controller clock, set the DSS.DSS_CONTROL[0] DSS_CLK_SWITCH bit.
2. To configure the top level of the video encoder, set the DSS.DSS_CONTROL[2] VENC_CLOCK_MODE bit and the DSS.DSS_CONTROL[3] VENC_CLOCK_X4_EN bit for TV set support.
3. To configure the top level of the DAC stage, set the DSS.DSS_CONTROL[4] DAC_DEMEN bit for TV set support (if required).
4. Configure the RFBI module and/or the video encoder as needed.
5. Configure the display controller.

13.4.3 Display Controller Basic Programming Model

Some display controller registers are termed *shadow registers*, which are associated with the digital output and/or the LCD output. A shadow register change has no direct effect on the configuration of the display controller unless the DSS.DISPC_CTRL[5] GO_LCD bit is set for the LCD output and/or the DSS.DISPC_CTRL[6] GO_DIGITAL bit is set for the digital output.

In the case of the digital output, after programming the shadow registers, the DSS.DISPC_CTRL[6] GO_DIGITAL bit must be set to 1. If this bit is not set, the configuration of the display controller will have no effect. This setting indicates that all display controller shadow registers are programmed and that hardware can update the internal registers at the external EVSYNC.

In the case of the LCD output, after programming the shadow registers, the DSS.DISPC_CTRL[5] GO_LCD bit must be set to 1. If this bit is not set, the configuration of the display controller will have no effect. This setting indicates that all display controller shadow registers are programmed and that hardware can update the internal registers at the VFP start period.

Before setting either the DSS.DISPC_CTRL[5] GO_LCD or DSS.DISPC_CTRL[6] GO_DIGITAL bit, ensure that the bit is cleared.

Table 13-21 lists the shadow registers.

Table 13-21. Shadow Registers

Shadow Register Name	Updated on VFP Start Period (LCD output)	Updated on External VSYNC (Digital output)
DSS.DISPC_CTRL	X[1]	X[1]
DSS.DISPC_CFG	X	X
DSS.DISPC_DEFAULT_COLOR_m (m = 0)	X	
DSS.DISPC_DEFAULT_COLOR_m (m = 1)		X
DSS.DISPC_TRANS_COLOR_m (m = 0)	X	
DSS.DISPC_TRANS_COLOR_m (m = 1)		X
DSS.DISPC_LINE_NUMBER	X	
DSS.DISPC_TIMING_H	X	
DSS.DISPC_TIMING_V	X	
DSS.DISPC_POL_FREQ	X	
DSS.DISPC_DIVISOR	X	
DSS.DISPC_SIZE_DIG		X
DSS.DISPC_SIZE_LCD	X	
DSS.DISPC_GFX_BAj (j = 0,1)	X	X
DSS.DISPC_GFX_POSITION	X	X
DSS.DISPC_GFX_SIZE	X	X
DSS.DISPC_GFX_ATTRS	X	X
DSS.DISPC_GFX_FIFO_THR	X	X
DSS.DISPC_GFX_ROW_INC	X	X
DSS.DISPC_GFX_PIXEL_INC	X	X
DSS.DISPC_GFX_WINDOW_SKIP	X	X
DSS.DISPC_GFX_TBL_BA	X	X
DSS.DISPC_GFX_PRELOAD	X	X
DSS.DISPC_CPR_COEF_R	X	
DSS.DISPC_CPR_COEF_G	X	
DSS.DISPC_CPR_COEF_B	X	
DSS.DISPC_VIDn_BAj (j= 0,1)	X	X
DSS.DISPC_VIDn_POSITION	X	X
DSS.DISPC_VIDn_SIZE	X	X
DSS.DISPC_VIDn_ATTRS	X	X

---

(1) Some of the register bit fields are shadow bits. For more information, see Section 13.6, *Display Subsystem Register Manual*.

Table 13-21. Shadow Registers (continued)

Shadow Register Name	Updated on VFP Start Period (LCD output)	Updated on External VSYNC (Digital output)
DSS.DISPC_VIDn_FIFO_THR	X	X
DSS.DISPC_VIDn_ROW_INC	X	X
DSS.DISPC_VIDn_PIXEL_INC	X	X
DSS.DISPC_VIDn_FIR	X	X
DSS.DISPC_VIDn_PICTURE_SIZE	X	X
DSS.DISPC_VIDn_ACCUl (l = 0,1)	X	X
DSS.DISPC_VIDn_FIR_COEF_H_i (i = 0 to 7)	X	X
DSS.DISPC_VIDn_FIR_COEF_HV_i (i = 0 to 7)	X	X
DSS.DISPC_VIDn_FIR_COEF_Vi (i = 0 to 7)	X	X
DSS.DISPC_VIDn_CONV_COEFi (i = 0 to 4)	X	X
DSS.DISPC_VIDn_PRELOAD	X	X
DSS.DISPC_DATA_CYCLEk (k = 0 to 3)	X	

13.4.3.1 Display Controller Configuration

The following registers define the display controller configuration:

- DSS.DISPC_SYSCFG
- DSS.DISPC_SYSSTS
- DSS.DISPC_IRQSTS
- DSS.DISPC_IRQEN

13.4.3.2 Graphics Layer Configuration

The graphics layer configuration is common to the LCD and the TV set.

13.4.3.2.1 Graphics DMA Registers

The following registers define the graphics DMA engine configuration:

- DSS.DISPC_CTRL
- DSS.DISPC_GFX_BAj
- DSS.DISPC_GFX_ATTRS
- DSS.DISPC_GFX_ROW_INC
- DSS.DISPC_GFX_PIXEL_INC
- DSS.DISPC_GFX_FIFO_THR
- DSS.DISPC_GFX_TBL_BA

The following fields define the attributes of the graphics DMA engine:

- Graphics layer enable (DSS.DISPC_GFX_ATTRS[0] GFX_EN bit): The default value of this bit at reset time is 0x0 (Disabled). The graphics DMA engine fetches encoded pixels from the system memory only when the graphics layer is enabled (a valid configuration is programmed for the graphics layer). The graphics window is present and the graphics pipeline is active.
- Burst size (DSS.DISPC_GFX_ATTRS[7:6] GFX_BURST_SIZE field): The default burst size at reset time is 4 x 32 bytes. The possible values are 4 x 32, 8 x 32, and 16 x 32 bytes. The burst size is initialized at boot time by the software and never changes as long as the display controller is enabled. This field indicates the maximum burst size for the specific pipeline. In case of misalignment, the DMA engine may issue single and/or smaller burst requests because the burst size must be aligned to the burst boundary.
- Preload configuration (DSS.DISPC_GFX_ATTRS[11] GFX_FIFO_PRELOAD bit): The default preload configuration uses the DSS.DISPC_GFX_PRELOAD register value (the reset value is 256 bytes) to

define the number of bytes to be fetched from system memory into the display controller graphics FIFO. By programming the DSS.DISPC_GFX_ATTRS[11] GFX_FIFO_PRELOAD bit, software users select between preload register (with 256 bytes as the reset value) and the high threshold value for preload of the encoded pixels. For best performance, the configuration of thresholds is defined using the FIFO size (in bytes) minus 1 for the high threshold, and the FIFO size (in bytes) minus the burst size (in bytes) for the low threshold, which provides 960, 992, and 1008, respectively, for burst sizes 16x32, 8x32, and 4x32. Note also that the preload value is defined based on the following display types:

  - Active matrix (TFT) display: DSS.DISPC_GFX_PRELOAD[11:0] PRELOAD = 0x60 (value is 96)
  - Color passive matrix (STN) display: DSS.DISPC_GFX_PRELOAD[11:0] PRELOAD = 0x72 (value is 114)
  - Monochrome passive matrix (STN) display: DSS.DISPC_GFX_PRELOAD[11:0] PRELOAD = 0xE0 (value is 224)

- Base address of the graphics buffer in system memory (DSS.DISPC_GFX_BAj registers): The default value of these two registers at reset time is 0x0. The horizontal resolution is one pixel because the base address is aligned on a pixel size boundary. In case of 4 BPP, the resolution is two pixels; for 2 BPP, resolution is four pixels; for 1 BPP, resolution is eight pixels; and for RGB24 packed format, the resolution is four pixels. The vertical resolution is one line. The register DSS.DISPC_GFX_BA0 defines the base address of the even field; and DSS.DISPC_GFX_BA1 defines the base of the odd field in the case of an external synchronization and based on the value of the input signal DISPC_FID and the polarity. To improve system throughput, the base address should be aligned on the burst size boundary.
- Graphics FIFO threshold (DSS.DISPC_GFX_FIFO_THR register): The low threshold (DSS.DISPC_GFX_FIFO_THR[11:0] GFX_FIFO_LOW_THR) and the high threshold (DSS.DISPC_GFX_FIFO_THR[27:16] GFX_FIFO_HIGH_THR) values define the FIFO DMA behavior. When the low level is reached, one or more requests are issued to the L3-based interconnect to fill up the FIFO to reach the high threshold. A request is issued as long as the FIFO has enough space available to accept a burst. The DMA engine then waits until the low level is reached to restart the requests. By setting the DSS.DISPC_CFG[14] FIFO_MERGE bit to 1, users merge the three FIFOs (GFX, VID1, and VID2). In this case, the low threshold (the DSS.DISPC_GFX_FIFO_THR[11:0] GFX_FIFO_LOW_THR bit field) and the high threshold (DSS.DISPC_GFX_FIFO_THR[27:16] GFX_FIFO_HIGH_THR bit field) values must be programmed with a multiplier factor of three (3 x value). By default, the FIFOs are not merged (the DSS.DISPC_CFG[14] FIFO_MERGE bit reset value is 0).
- Palette/gamma table used (DSS.DISPC_CFG[3] PALETTEGAMMA_TBL bit): The bit indicates if the palette must be loaded before the graphics data for the following frame. The bit is set by software and reset by hardware.
- Base address of the palette/gamma table buffer in system memory (DSS.DISPC_GFX_TBL_BA register): The default value of this register at reset time is 0x0. The base address is aligned on a 32-bit address. Depending on the pixel size of graphics data (1, 2, 4, or 8 BPP), 16 (1, 2, or 4 BPP), or 256 (8 BPP) x 32-bit values are loaded from system memory into the internal table memory. To load the table when using the memory as a gamma table, the graphics pipeline is enabled and then disabled by the software when the palette loaded interrupt is generated. The overlay manager ignores the graphics pipe when the table is used as a gamma table.

NOTE: In case of RGB16 format and optimization enabled, the base address is aligned on a 32-bit boundary and the number of bytes to skip is a multiple of 4 bytes.

- Graphics Priority (DSS.DISPC_GFX_ATTRS[14] GFX_ARBITRATION): The default value at reset time is 0x0. It is used to change between normal priority (value of 0) to high priority (value of 1) to change priority for the graphics channel vs. video channels. It can be used to give higher priority to the pipelines with real time constraint vs. non real time pipelines. For that is, pipelines associated to the LCD output in RFBI mode should have lower priority than pipelines associated to TV output.
- Graphics Self-Refresh (DSS.DISPC_GFX_ATTRS[15] GFX_SELF_REFRESH): The default value at reset time is 0x0. It is used to use the DMA FIFO without accessing the interconnect for multiple frames. Once, the data have been loaded to the DMA FIFO for displaying the frame, they are used for the following frames.

The sequence to activate the self-refresh is the following:

- Frame t: The bit field should be set at anytime during frame
- Frame t+1: Fetch of the data in the DMA FIFO and display of the frame
- Frame t+2: No access to the L3 interconnect, DMA FIFO uses to provide the pixels

The sequence to deactivate the self-refresh is the following:

- Frame t: No access to the L3 interconnect, DMA FIFO uses to provide the pixels, bit field can be changed at any time during the frame
- Frame t+1: Fetch of the data from system memory using the L3 interconnect

13.4.3.2.2 Graphics Layer Configuration Registers

The following registers define the graphics layer configuration:

- DSS.DISPC_CFG
- DSS.DISPC_GFX_POSITION
- DSS.DISPC_GFX_SIZE
- DSS.DISPC_GFX_ATTRS

The graphics layer is enabled/disabled by setting/resetting the DSS.DISPC_GFX_ATTRS[0] GFX_EN bit. When the graphics layer is disabled, the graphics window does not exist on the screen and the graphics pipeline and DMA are inactive.

Set a valid configuration before enabling the graphics layer. After a register change, either the DSS.DISPC_CTRL[6] GO_DIGITAL or DSS.DISPC_CTRL[5] GO_LCD bit must be set. The software must wait for the hardware to reset the bit before setting it. The software reset is not recommended because the application cannot ensure that the bit is reset before the hardware reset.

13.4.3.2.3 Graphics Window Attributes

The following fields define the attributes of the graphics window:

- Graphics format (DSS.DISPC_GFX_ATTRS[4:1] GFX_FMT bit field): The default value of this bit field at reset time is 0x0 (BITMAP 1-BPP). The graphics format can be either: BITMAP1, BITMAP2, BITMAP4, or BITMAP8 (CLUT) or RGB12, RGB16, or RGB24 (true-color formats).
- Graphics window X-position (DSS.DISPC_GFX_POSITION[10:0] GFX_POSX bit field): The default value at reset time is 0x0. The window X-position is from 0 to 2047 columns. All integer values in the range [0:2047] are allowed.
- Graphics window Y-position (DSS.DISPC_GFX_POSITION[26:16] GFX_POSY bit field): The default value of this bit field at reset time is 0x0.The window Y-position is from 0 to 2047 rows. All integer values in the range [0:2047] are allowed.
- Graphics window width (DSS.DISPC_GFX_SIZE[10:0] GFX_SIZEX bit field): The default value at reset time is 0x0 (1 pixel). The window width is from 1 to 2048 pixels. All integer values in the range [1:2048] are allowed for the following formats: 8 BPP, RGB12, RGB16, and RGB24. The width must be a multiple of eight pixels for 1 BPP, four pixels for 2 BPP, and two pixels for 4 BPP. The maximum bandwidth efficiency for accessing the pixels in system memory is reached when the width of the graphics window (in bytes) is a multiple of the graphics burst size defined in the DSS.DISPC_GFX_ATTRS[7:6] GFX_BURST_SIZE bit field (in bytes).

NOTE: When the RGB24 packed format is selected, the width must be a multiple of 12 bytes when the DSS.DISPC_GFX_ROW_INC register is not 1. When DSS.DISPC_GFX_ROW_INC register is 1, the width can be any size from 1 to 2048 pixels.

The entire pixels of the graphics window must be inside the LCD screen. Depending on the width of the buffer to be displayed in the graphics layer and the position, the width should be adjusted by software to limit the right edge of the window inside the screen.

- Graphics window height (DSS.DISPC_GFX_SIZE[26:16] GFX_SIZEY bit field): The default value at reset time is 0x0 (1 pixel). The window height is from 1 to 2048 pixels. All integer values in the range [1:2048] are allowed. The entire pixels of the graphics window must be inside the LCD screen.

Depending on the height of the buffer to be displayed in the graphics layer and the position, the height should be adjusted by software to limit the bottom edge of the window inside the screen

- Graphics data endianness (DSS.DISPC_GFX_ATTRS[10] GFX_ENDIAN bit): This bit indicates the endianness (little or big) of the graphics pixels. The default value at reset time is 0x0 (little endian).
- Graphics data nible mode (DSS.DISPC_GFX_ATTRS[9] GFX_NIBBLE_MODE bit): This bit indicates the nibble mode of the graphics pixels. The default value at reset time is 0x0 (Disable).
- Graphics replication logic enable (DSS.DISPC_GFX_ATTRS[5] GFX_REPLICATION_EN bit): The default value at reset time is 0x0 (Disable). The encoded pixel data in RGB format (RGB16) can be extended to 24-bit format with or without replication of the MSB part to fill up the LSB due to the 24-bit left alignment. If the replication logic is turned off, the LSB part is filled up with 0s.
- Graphics window skip enable (DSS.DISPC_CTRL[12] OVLY_OPT bit): By setting/resetting the bit, the overlay optimization is enabled or disabled. Before enabling the overlay optimization, the DSS.DISPC_GFX_WINDOW_SKIP[31:0] GFX_WINDOW_SKIP bit field must be set according to the video1 and graphics windows overlap. The default value at reset time is 0x0 (Disable). When video1 is not present, the DSS.DISPC_GFX_WINDOW_SKIP[31:0] GFX_WINDOW_SKIP bit field should be reset. When the color key is used, the DSS.DISPC_GFX_WINDOW_SKIP[31:0] GFX_WINDOW_SKIP bit field should be reset.
- Graphics window skip (DSS.DISPC_GFX_WINDOW_SKIP[31:0] GFX_WINDOW_SKIP bit field): The bit field represents the number of bytes to skip while fetching the graphics-encoded pixels when reaching the beginning of the video window. The optimization allows fetching only the visible graphics pixels. The color key cannot be selected because the graphics pixels under the video window are not present. The default value at reset time is 0x0 (0 byte).

Figure 13-54 through Figure 13-57 give examples of how to program the GFX_WINDOW_SKIP field for overlay optimization:

Figure 13-54. Overlay Optimization: Case 1

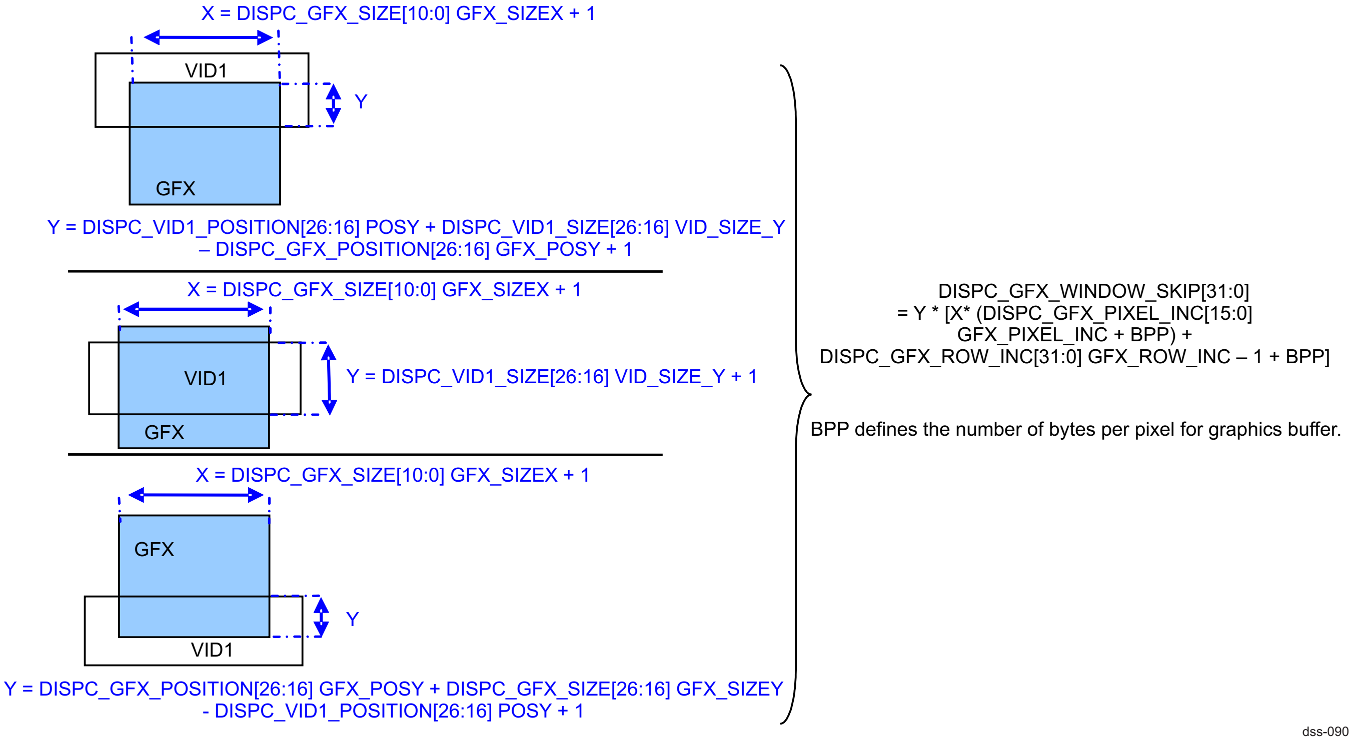


Figure 13-55. Overlay Optimization: Case 2

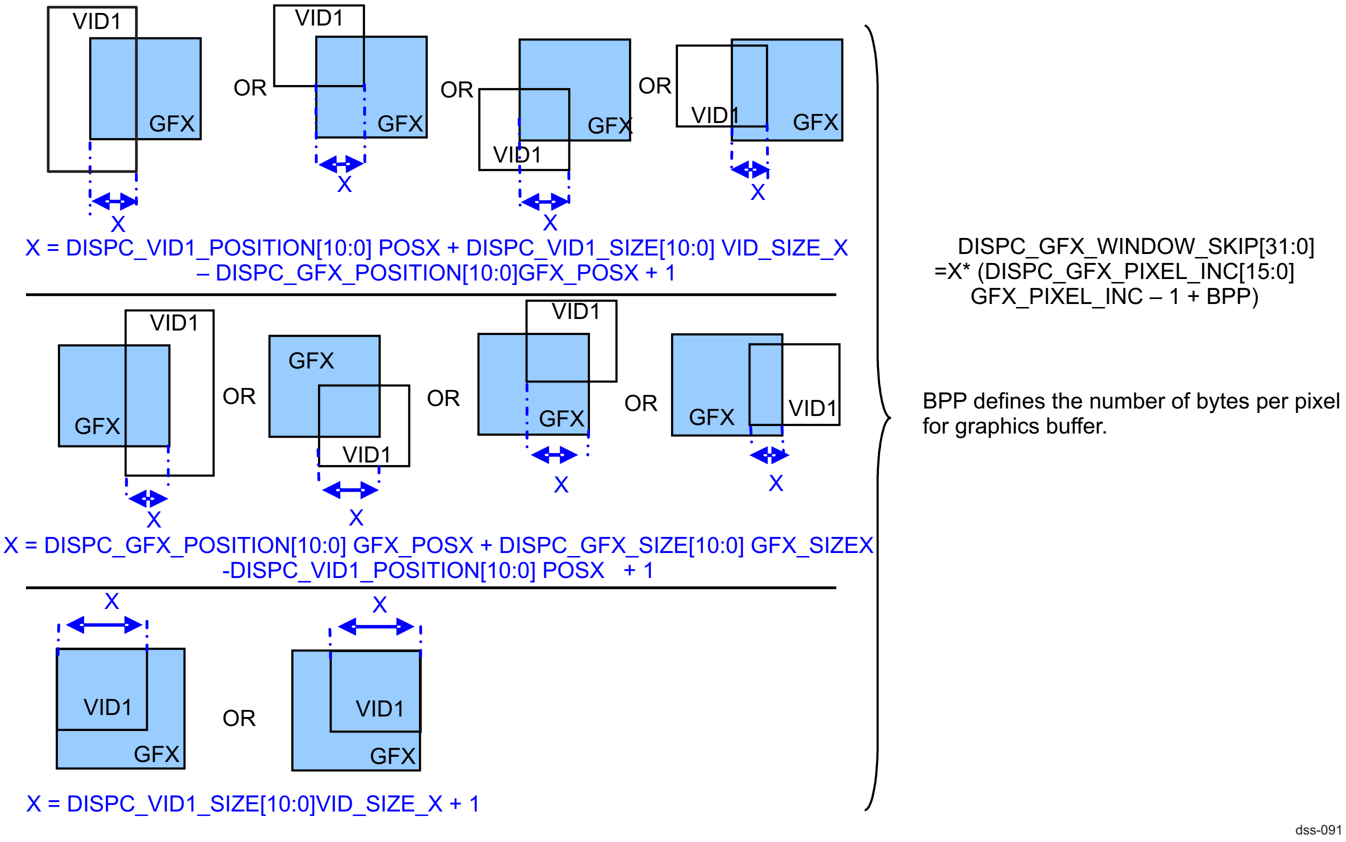


Figure 13-56. Overlay Optimization: Case 3

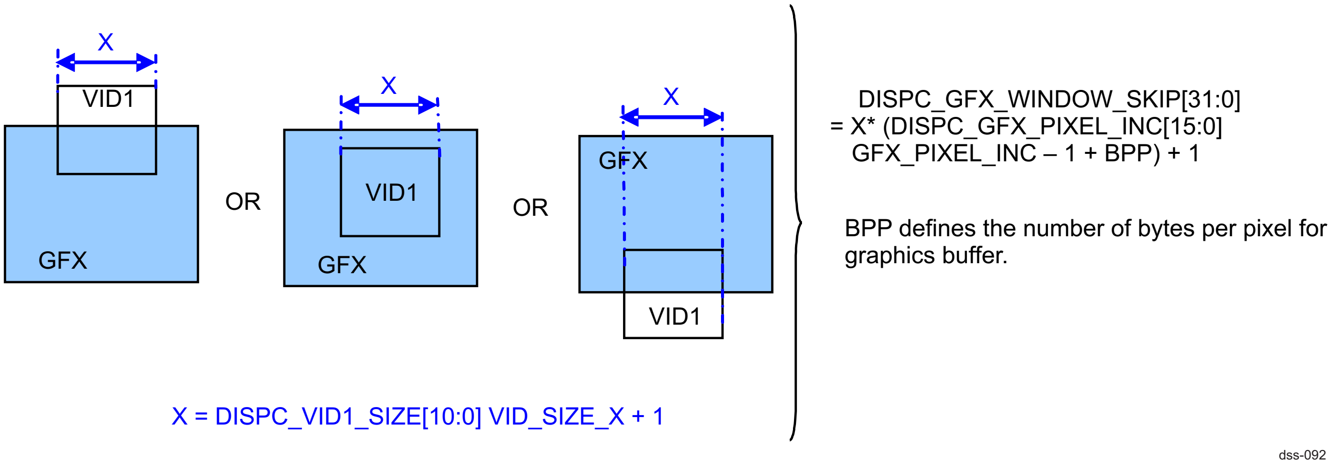
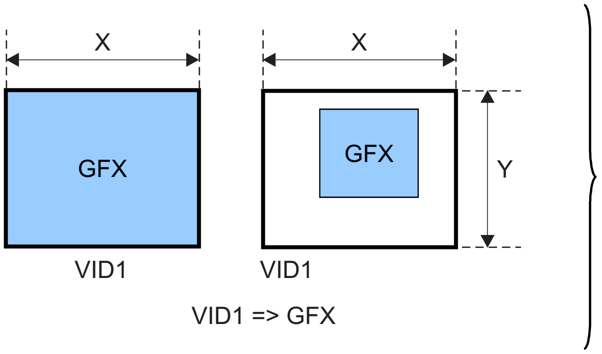


Figure 13-57. Overlay Optimization: Case 4



When the VID1 window overlaps the GFX window equally or more, the DISPC_GFX_ATTRS[0] GFX_EN bit should be clear.

camdss-093

13.4.3.3 Video Layer Configuration

The video layer configuration is common to the LCD and the TV set.

13.4.3.3.1 Video DMA Registers

The following registers define the video DMA engine configuration:

- DSS.DISPC_CTRL
- DSS.DISPC_VIDn_BAj
- DSS.DISPC_VIDn_ATTRS
- DSS.DISPC_VIDn_ROW_INC
- DSS.DISPC_VIDn_PIXEL_INC
- DSS.DISPC_VIDn_FIFO_THR
- DSS.DISPC_VIDn_PICTURE_SIZE

The following fields define the attributes of the graphics DMA engine:

- Video layer enable (DSS.DISPC_VIDn_ATTRS[0] EN bit): The default value of this bit at reset time is 0x0 (Disabled). The video DMA engine fetches encoded pixels from the system memory only when the video layer is enabled (a valid configuration is programmed for the video layer). The video window is present and the video pipeline is active.
- Burst size (DSS.DISPC_VIDn_ATTRS[15:14] BURST_SIZE bit field): The default burst size at reset time is 4 x 32 bytes. The possible values are 4 x 32, 8 x 32, and 16 x 32 bytes. The burst size is initialized at boot time by the software and never changes as long as the display controller is enabled. This bit field indicates the maximum burst size for the specific pipeline. In case of misalignment, the DMA engine may issue single and/or smaller burst requests, because the burst size must be aligned to the burst boundary.
- Preload configuration (DSS.DISPC_VIDn_ATTRS[19] FIFO_PRELOAD bit): The default preload configuration uses the DSS.DISPC_VIDn_PRELOAD register value (the reset value is 256 bytes) to define the number of bytes to be fetched from system memory into the display controller graphics FIFO. By programming the DSS.DISPC_VIDn_ATTRS[19] FIFO_PRELOAD bit, software users select between preload register (with 256 bytes as the reset value) and the high threshold value for preload of the encoded pixels. For best performance, the configuration of thresholds is defined using the FIFO size (in bytes) minus 1 for the high threshold, and the FIFO size (in bytes) minus the burst size (in bytes) for the low threshold, which provides 960, 992, and 1008, respectively, for burst sizes 16x32, 8x32, and 4x32. Note also that the preload value is defined based on the following display types:
  - Active matrix (TFT) display: DSS.DISPC_VIDn_PRELOAD[11:0] PRELOAD = 0xB0 (value is 176)
  - Color passive matrix (STN) display: DSS.DISPC_VIDn_PRELOAD[11:0] PRELOAD = 0x110 (value is 272)
  - Monochrome passive matrix (STN) display: DSS.DISPC_VIDn_PRELOAD[11:0] PRELOAD = 0x1B0 (value is 432)

- Base address of the video buffer in system memory (DSS.DISPC_VIDn_BAj registers): The default value at reset time is 0x0. The horizontal resolution is one pixel because the base address is aligned on pixel size boundary. In case of YCbCr 4:2:2 formats, the resolution is 2 pixels. In case of RGB24 packed format, the resolution is 4 pixels. The vertical resolution is one line. The register DSS.DISPC_VIDn_BA0 defines the base address of the even field, and DSS.DISPC_VIDn_BA1 defines the base of the odd field in the case of an external synchronization and based on the value of the input signal DISPC_FID and the polarity. To improve system throughput, the base address should be aligned on the burst size boundary.
- Video FIFO threshold (DSS.DISPC_VIDn_FIFO_THR register): The low threshold (DSS.DISPC_VIDn_FIFO_THR[11:0] VID_FIFO_LOW_THR) and the high threshold (DSS.DISPC_VIDn_FIFO_THR[27:16] VID_FIFO_HIGH_THR) values define the FIFO DMA behavior. When the low level is reached, one or more requests are issued to the L3-based interconnect to fill up the FIFO to reach the high threshold. A request is issued as long as the FIFO has enough space available to accept a burst. The DMA engine then waits until the low level is reached to restart the requests. By setting the DSS.DISPC_CFG[14] FIFO_MERGE bit to 1, users merge the three FIFOs (GFX, VID1, and VID2). In this case, the low threshold (the DSS.DISPC_VIDn_FIFO_THR[11:0] VID_FIFO_LOW_THR bit field with n corresponding to the active video channel 1 or 2) and the high threshold (DSS.DISPC_VIDn_FIFO_THR[27:16] VID_FIFO_HIGH_THR bit field with n corresponding to the active video channel 1 or 2) values must be programmed with a multiplier factor of three (3 x value). By default, the FIFOs are not merged (the DSS.DISPC_CFG[14] FIFO_MERGE bit reset value is 0).
- Video buffer width (DSS.DISPC_VIDn_PICTURE_SIZE[10:0] VID_ORG_SIZE_X): The default value at reset time is 0x0 (1 pixel). The buffer width in system memory is from 1 up to 2048 pixels. All the integer values in the range [1:2048] are allowed. Software users must program this bit field to the value minus 1.
- Video buffer height (DSS.DISPC_VIDn_PICTURE_SIZE[26:16] VID_ORG_SIZE_Y): The default value at reset time is 0x0 (1 pixel). The buffer height in system memory is from 1 up to 2048 pixels. All the integer values in the range [1:2048] are allowed. Software users must program this field to the value minus 1.
- Video data endianness (DSS.DISPC_VIDn_ATTRS[17] ENDIAN bit, with n=1 or 2): This bit indicates the endianness (little or big) of the video pixels. The default value at reset time is 0x0 (little endian).

13.4.3.3.2 Video Configuration Register

The following shadow registers define video layer n (with n = 1 or 2) configuration:

- DSS.DISPC_CFG
- DSS.DISPC_VIDn_POSITION
- DSS.DISPC_VIDn_SIZE
- DSS.DISPC_VIDn_ATTRS
- DSS.DISPC_VIDn_FIR
- DSS.DISPC_VIDn_PICTURE_SIZE
- DSS.DISPC_VIDn_FIR_COEF_H_i (with i = 0 to 7)
- DSS.DISPC_VIDn_FIR_COEF_HV_i (with i = 0 to 7)
- DSS.DISPC_VIDn_CONV_COEFi (with i = 0 to 4)
- The video layer n (with n = 1 or 2) is enabled/disabled by setting/resetting the DSS.DISPC_VIDn_ATTRS[0] EN field. If the video layer is disabled, the video window does not exist on the screen and the whole video pipeline and DMA are inactive. Before enabling the video layer, a valid configuration must be set. After a register change, either the DSS.DISPC_CTRL[6] GO_DIGITAL or DSS.DISPC_CTRL[5] GO_LCD bit must be set. The software must wait for the hardware to reset the bit before setting this bit. The software reset is not recommended because the application cannot ensure that the bit is reset before the hardware reset.

13.4.3.3.3 Video Window Attributes

The following fields define the attributes of video window n:

- Video format (DSS.DISPC_VIDn_ATTRS[4:1] FMT bit field, with n = 1 or 2): The default value at reset time is 0x0 (BITMAP 1 BPP, nonsupported format by the video pipeline).The video format can be RGB16, RGB24, YUV2 4:2:2 co-DSS sited, and UYVY 4:2:2 co-sited.
- Video window X-position (DSS.DISPC_VIDn_POSITION[10:0] POSX bit field, with n = 1 or 2): The default value at reset time is 0x0 (first column starting on the left edge of the screen). The window X-position is from 0 to 2047 columns. All integer values in the range [0:2047] are allowed.
- Video window Y-position (DSS.DISPC_VIDn_POSITION[26:16] POSY bit field, with n = 1 or 2): The default value at reset time is 0x0 (first row starting at the top of the screen). The window Y-position is from 0 to 2047 rows. All integer values in the range [0:2047] are allowed.
- Video window width (DSS.DISPC_VIDn_SIZE[10:0] VID_SIZE_X bit field, with n = 1 or 2): The default value at reset time is 0x0 (1 pixel). The window width is from 1 to 2048 pixels. All integer values in the range [1:2048] are allowed. The maximum bandwidth efficiency for accessing the pixels in system memory is reached when the width (in bytes) of the video window is a multiple of the video burst size defined in the DSS.DISPC_VIDn_ATTRS[15:14] VID_BURST_SIZE bit field (in bytes).

NOTE: When the RGB24 packed format is selected, the width must be a multiple of 12 bytes when the DSS.DISPC_VIDn_ROW_INC register is not 1. When the DSS.DISPC_VIDn_ROW_INC register is 1, the width can be any size from 1 to 2048 pixels.

The entire pixels of the video window must be inside the LCD screen. Depending on the width of the buffer to be displayed in the video layer and the position, the width should be adjusted by software to limit the right edge of the window inside the screen.

- Video window height (DSS.DISPC_VIDn_SIZE[26:16] VID_SIZE_Y bit field, with n = 1 or 2): The default value at reset time is 0x0 (1 pixel). The window height is from 1 to 2048 pixels. All integer values in the range [1:2048] are allowed. The entire pixels of the video window must be inside the LCD screen. Depending on the height of the buffer to be displayed in the video layer and the position, the height should be adjusted by software to limit the bottom edge of the window inside the screen.
- Video picture width in system memory (DSS.DISPC_VIDn_PICTURE_SIZE[10:0] VID_ORG_SIZE_X bit field, with n = 1 or 2): The default value at reset time is 0x0 (1 pixel). The window width is from 1 to 2048 pixels. All integer values in the range [1:2048] are allowed with RGB16 and RGB24 video data. For YUV2 4:2:2 and UYVY 4:2:2 formats, the width must be a multiple of two pixels. The maximum bandwidth efficiency for accessing the pixels in system memory is reached when the width (in bytes) of the video picture is a multiple of the video burst size defined in the DSS.DISPC_VIDn_ATTRS[15:14] VID_BURST_SIZE bit field (in bytes).
- Video picture height in system memory (the DSS.DISPC_VIDn_PICTURE_SIZE[26:16] VID_ORG_SIZE_Y bit field, with n = 1 or 2): The default value at reset time is 0x0 (1 pixel). The window width is from 1 to 2048 pixels. All integer values in the range [1:2048] are allowed.
- Video Priority (DSS.DISPC_VIDn_ATTRS[23] ARBITRATION): The default value at reset time is 0x0. It is used to change between normal priority (value of 0) to high priority (value of 1) to change priority for the video channel vs. other channels. It can be used to give higher priority to the pipelines with real time constraint vs. non real time pipelines. For that is, pipelines associated to the LCD output in RFBI mode should have lower priority than pipelines associated to TV output.
- Video Self-Refresh (DSS.DISPC_VIDn_ATTRS[24] SELF_REFRESH): The default value at reset time is 0x0. It is used to use the DMA FIFO without accessing the interconnect for multiple frames. Once, the data have been loaded to the DMA FIFO for displaying the frame, they are used for the following frames.
  The sequence to activate the self-refresh is the following:
  - Frame t: The bit field should be set at anytime during frame
  - Frame t+1: Fetch of the data in the DMA FIFO and display of the frame
  - Frame t+2: No access to the L3 interconnect, DMA FIFO uses to provide the pixels

  The sequence to deactivate the self-refresh is the following:
  - Frame t: No access to the L3 interconnect, DMA FIFO uses to provide the pixels, bit field can be changed at any time during the frame
  - Frame t+1: Fetch of the data from system memory using the L3 interconnect

13.4.3.3.4 Video Up-/Down-Sampling Configuration

The video horizontal up/downsampling block for video pipeline n (with n = 1 or 2) is enabled/disabled by setting/resetting the DSS.DISPC_VIDn_ATTRS[5] RESIZE_EN bit.

The video vertical up/downsampling block for video pipeline n is enabled/disabled by setting/resetting the DSS.DISPC_VIDn_ATTRS[6] RESIZE_EN bit.

Set a valid configuration before enabling the video up/downsampling block.

NOTE: Vertical and horizontal downsampling are limited to a 1/4 resize factor.

After a register change, either the DSS.DISPC_CTRL[6] GO_DIGITAL or DSS.DISPC_CTRL[5] GO_LCD bit must be set. The software must wait until the hardware resets this bit before setting it. The software reset is not recommended because the application cannot ensure that the bit is reset before the hardware reset.

The following fields define the configuration of the video up/downsampling block for video pipeline n:

- Vertical up/downsampling increment value (DSS.DISPC_VIDn_FIR[27:16] FIR_V_INC bit field, with n = 1 or 2): The unsigned integer value range is [1:4096]. The software calculates the value using the following equation:

$$FIR\_V\_INC[12:0] = 1024 \text{ x } \left(\frac{VID\_ORG\_SIZE\_Y[10:0]}{VID\_SIZE\_Y[10:0]}\right) \tag{8}$$

dss-E093

NOTE:

- If the FIR_V_INC[11:0] bit field value is greater than 4096, it is clipped to 4096. If VID_SIZE_Y[10:0] equals 0x1, VID_SIZE_Y[10:0] is replaced by 0x2 in the previous equation.
- The VID_ORG_SIZE_Y[10:0] and VID_SIZE_Y[10:0] bit field values must be programmed with the value desired minus 1.

- Horizontal up/downsampling increment value (DSS.DISPC_VIDn_FIR[11:0] FIR_H_INC bit field, with n = 1 or 2): The unsigned integer value range is [1:4096]. The software calculates the value using the following equation:

$$FIR\_H\_INC[12:0] = 1024 \text{ x } \left(\frac{VID\_ORG\_SIZE\_X[10:0]}{VID\_SIZE\_X[10:0]}\right) \tag{9}$$

dss-E094

NOTE:

- If the FIR_H_INC[11:0] bit field value is greater than 4096, it is clipped to 4096. If VID_SIZE_X[10:0] equals 1, VID_SIZE_X[10:0] is replaced by 2 in the previous equation.
- The VID_ORG_SIZE_X[10:0] and VID_SIZE_X[10:0] bit field values must be programmed with the value desired minus 1.

- Vertical up/downsampling accumulator value (DSS.DISPC_VIDn_ACCUl[25:16] VERTICAL_ACCU bit field): The unsigned integer value range is [0:1023]. The accumulator value indicates in which phase the vertical filtering starts. The value 0 indicates that 0 is the first phase used by the hardware to generate the first data (see Table 13-22).
- Vertical up/downsampling line buffer configuration (DSS.DISPC_VIDn_ATTRS[22] LINE_BUFFER_SPLIT bit): The default value at reset time is 0x0 (line buffers are not split). The backward compatibility is maintained versus OMAP2420 and OMAP2430 devices. When the bit field is set, each line buffer is split into two line buffers to be able to use six line buffers instead of three.
- Vertical up/downsampling line buffer configuration (DSS.DISPC_VIDn_ATTRS[21] VERTICAL_TAPS bit): The default value at reset time is 0x0 (3-tap configuration is used). If the bit field is reset, the 3-tap configuration is used. The backward compatibility is maintained versus OMAP2420 and OMAP2430 devices. When the bit field is set, the 5-tap configuration is used and the

DSS.DISPC_VIDn_ATTRS[22] LINE_BUFFER_SPLIT bit must be set to 1.

- Vertical up/downsampling line buffer configuration (DSS.DISPC_VIDn_ATTRS[20] OPTIMIZATION bit): The default value at reset time is 0x0 (no optimization). If the bit is set, the DMA engine fetches two pixels for each 32-bit OCP request (RGB16 and YUV4:2:2) while doing 90- and 270-degree rotation. If the bit is clear, the DMA engine fetches one pixel for each 32-bit OCP request (RGB16 and YUV4:2:2) while doing 90- and 270-degree rotation. The width and height of picture should be even to use the optimization. Even width is required for the input picture when the 5-tap configuration is used.

NOTE: If the 5-tap resizer is used for RGB16 and YUV4:2:2 picture formats, the width of the input picture must be a multiple of 2 pixels and more than 5 pixels. This leads to the following register configuration:

DISPC_VIDn_ATTRS[21] VERTICAL_TAPS == 1
DISPC_VIDn_PICTURE_SIZE[10:0] VID_ORG_SIZE_X 4 and even

- Horizontal up/downsampling accumulator value (DSS.DISPC_VIDn_ACCUl[9:0] HORIZONTAL_ACCU bit field): The unsigned integer value range is [0:1023]. The accumulator value indicates in which phase the horizontal filtering starts. The value 0 indicates that 0 is the first phase used by the hardware to generate the first data (see Table 13-22).

Table 13-22. Vertical/Horizontal Accumulator Phase

Accumulator Value	Phases f
0	0
128	1
256	2
384	3
512	4
640	5
768	6
896	7

- Vertical up/downsampling coefficients (DSS.DISPC_VIDn_FIR_COEF_HV_i registers, with n = 1 or 2, i = 0 to 7): The 3-tap vertical up/downsampling coefficients are defined in these registers. There are eight registers for the eight phases with three coefficients for each, or a total of 24 programmable coefficients for the vertical up/downsampling block. Each register contains two 8-bit signed coefficients and one 8-bit unsigned coefficient (the central one).

  In addition, there are 2-tap vertical up/downsampling coefficients defined in DSS.DISPC_VIDn_FIR_COEF_Vi registers. There are 8 registers for the 8 phases with 2 coefficients for each of them so a total of 16 programmable coefficients for the vertical up/downsampling block used in addition of the 3-tap registers defined above. Each register contains two 8-bit signed coefficients. In case of 5-tap configuration, both sets of registers DSS.DISPC_VIDn_FIR_COEF_HV_i and DSS.DISPC_VIDn_FIR_COEF_Vi are used. In case of 3-tap configuration, only one set of registers DSS.DISPC_VIDn_FIR_COEF_HV_i is used.
- Horizontal up/downsampling coefficients (DSS.DISPC_VIDn_FIR_COEF_H_i and DISPC_VIDn_FIR_COEF_HV_i registers, with n = 1 or 2, i = 0 to 7): The DSS.DISPC_VIDn_FIR_COEF_H_i register and the DSS.DISPC_VIDn_FIR_COEF_HV_i register define the 5-tap horizontal up/downsampling coefficients. There are eight registers for the eight phases with five coefficients for each register, or a total of 40 programmable coefficients for the horizontal up/downsampling block.

  Each DSS.DISPC_VIDn_FIR_COEF_H_i register contains three 8-bit signed coefficients and one 8-bit unsigned coefficient (the central one). Each DSS.DISPC_VIDn_FIR_COEF_HV_i register contains one 8-bit signed coefficient.

  The programmable coefficient for the FIR up/downsampling method must be adjusted based on application needs. For more details on scaling programming settings, see Section 13.5.1.

13.4.3.3.5 Video Color Space Conversion Configuration

The DSS.DISPC_VIDn_CONV_COEFi registers (with i = 0 to 4) has nine 11-bit coefficients defined for the programmable color space conversion block for video pipeline n (with n = 1 or 2).

The standard register coefficients are:

YCbCr-to-RGB Registers (VidFullRange=0)

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} RY & RCr & RCb \\ GY & GCr & GCb \\ BY & BCr & BCb \end{bmatrix} * \begin{bmatrix} Y - 16 \\ Cr - 128 \\ Cb - 128 \end{bmatrix} \tag{10}$$

dssE095

YCbCr to RGB Registers (VidFullRange=1)

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} RY & RCr & RCb \\ GY & GCr & GCb \\ BY & BCr & BCb \end{bmatrix} * \begin{bmatrix} Y \\ Cr - 128 \\ Cb - 128 \end{bmatrix} \tag{11}$$

dss-E096

Table 13-23 lists the color space conversion register values.

Table 13-23. Color Space Conversion Register Values

Coefficients	BT.601-5	BT.601-5 Range [0:255]	BT.709	BT.709 Range [0:255]
RY	298	256	298	256
RCr	409	351	459	394
RCb	0	0	0	0
GY	298	256	298	256
GCr	–208	–179	–137	–118
GCb	–100	–86	–55	–47
BY	298	256	298	256
BCr	0	0	0	0
BCb	517	443	541	465
VidFullRange	0	1	0	1

13.4.3.4 LCD-Specific Control Registers

The following registers define the LCD output configuration:

- DSS.DISPC_CTRL
- DSS.DISPC_CFG
- DSS.DISPC_DEFAULT_COLOR_m (m=0)
- DSS.DISPC_TRANS_COLOR_m (m=0)
- DSS.DISPC_TIMING_H
- DSS.DISPC_TIMING_V
- DSS.DISPC_POL_FREQ
- DSS.DISPC_DIVISOR
- DSS.DISPC_SIZE_LCD
- DSS.DISPC_DATA_CYCLEk

- DSS.DISPC_CPR_COEF_R, DSS.DISPC_CPR_COEF_G, DSS.DISPC_CPR_COEF_B

Setting/resetting the DSS.DISPC_CTRL[0] LCD_EN bit enables/disables the LCD output. A valid configuration must be set before enabling the LCD output.

13.4.3.4.1 LCD Attributes

The following fields define the attributes of the panel connected to the display controller:

- Monochrome or color panel (the DSS.DISPC_CTRL[2] MONO_COLOR bit)
- Passive Matrix or active Matrix panel (the DSS.DISPC_CTRL[3] STNTFT bit)
- Color depth (the DSS.DISPC_CTRL[9:8] TFT_DATA_LINES bit field)
- Number of lines per panel (the DSS.DISPC_SIZE_LCD[26:16] LPP bit field)
- Number of pixels per line (the DSS.DISPC_SIZE_LCD[10:0] PPL bit field)
- 4- or 8-bit interface for Passive Matrix monochrome panel (the DSS.DISPC_CTRL[4] M8B bit)

13.4.3.4.2 LCD Timings

The following bit fields define the timing generation of HSYNC/VSYNC:

- Horizontal front porch (the DSS.DISPC_TIMING_H[19:8] HFP bit field)
- Horizontal back porch (the DSS.DISPC_TIMING_H[31:20] HBP bit field)
- Horizontal synchronization pulse width (the DSS.DISPC_TIMING_H[7:0] HSW bit field)
- Vertical front porch (the DSS.DISPC_TIMING_V[19:8] VFP bit field)
- Vertical back porch (the DSS.DISPC_TIMING_V[31:20] VBP bit field)
- Vertical synchronization pulse width (the DSS.DISPC_TIMING_V[7:0] VSW bit field)
- On/Off control of HSYNC/VSYNC pixel clock (the DSS.DISPC_POL_FREQ[17] ONOFF bit)
- Program HSYNC/VSYNC rise or fall (the DSS.DISPC_POL_FREQ[16] RF bit)
- Invert HSYNC (the DSS.DISPC_POL_FREQ[13] IHS bit)
- Invert VSYNC (the DSS.DISPC_POL_FREQ[12] IVS bit)
- HSYNC gated (the DSS.DISPC_CFG[6] HSYNC_GATED bit)
- VSYNC gated (the DSS.DISPC_CFG[7] VSYNC_GATED bit)

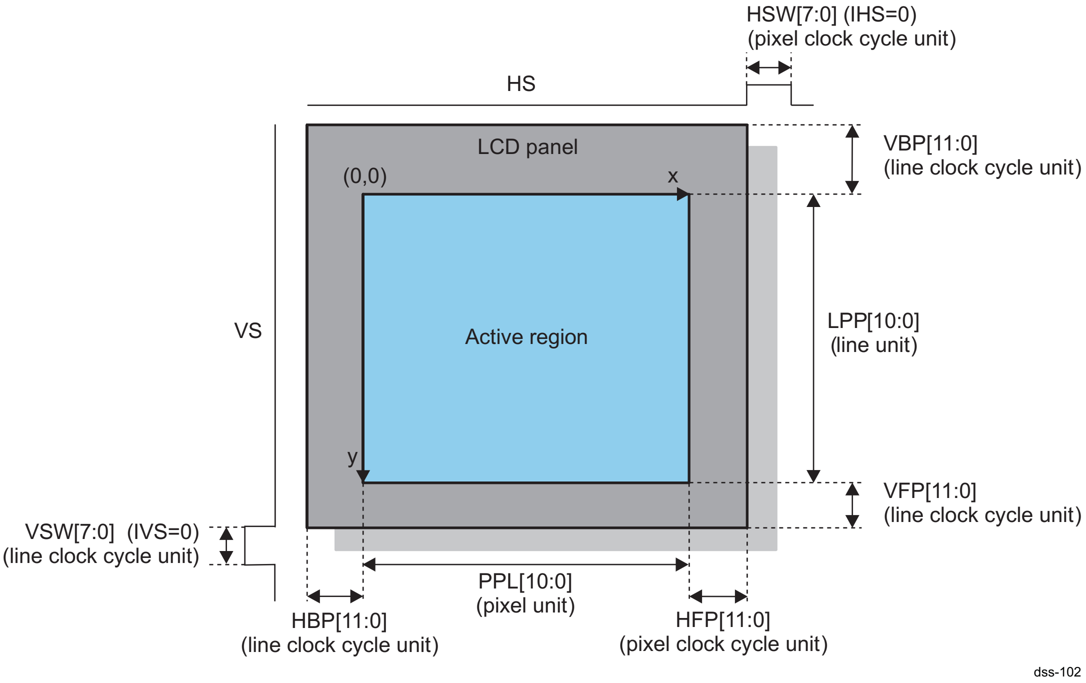
Table 13-24 describes the programming rules for LCD timing.

Table 13-24. Programming Rules

	No Downsampling	Downsampling H or V	Downsampling H + V
(HBP + HSW + HFP) * PCD	8	10	20

Figure 13-58 shows the timing values description in the case of an active matrix display.

Figure 13-58. Timing Values Description (Active Matrix Display)



The following bit fields define the timing generation of ac-bias (output enable in active matrix mode):

- Invert output enable (DSS.DISPC_POL_FREQ[15] IEO bit)
- ac-bias pin frequency (DSS.DISPC_POL_FREQ[7:0] ACB bit field)
- ac-bias pin transitions per interrupt (DSS.DISPC_POL_FREQ[11:8] ACBI bit field)
- ac-bias gated (DSS.DISPC_CFG[8] ACBIASGATED)

The following bit fields define the timing generation of the pixel clock:

- Pixel clock divisor (DSS.DISPC_DIVISOR[7:0] PCD bit field)
- Invert pixel clock (DSS.DISPC_POL_FREQ[14] IPC bit)
- Pixel clock gated (DSS.DISPC_CFG[5] PIXEL_CLK_GATED bit)

The 8-bit pixel clock divider (the DSS.DISPC_DIVISOR[7:0] PCD bit field) selects the pixel clock frequency. This bit field generates a range of pixel clock frequencies from LC/1 to LC/255, where LC is the logic clock from the divided functional clock of the display controller by the DSS.DISPC_DIVISOR[23:16] LCD bit field.

The pixel clock is defined by the following equation:

Pixel Clock = (FunctionalClock/LCD[7:0])/PCD[7:0]

Table 13-25 through Table 13-28 show the pixel clock frequency limitations depending the panel type (active or passive matrix) and the mode (color or monochrome).

Table 13-25. Pixel Clock Frequency Limitations - RGB16 and YUV4:2:2 Active Matrix Display

Min PCD Values			**Horizontal Resampling**				
			Off	**Up**	**1:1 - 1:2**	**1:2 - 1:3**	**1:3 - 1:4**
Vertical Resampling	Off		2 (1)(1)	2 (1)	2	3	4
	Up		2 (1)	2 (1)(1)	2	3	4
	1:1 - 1:2	3-tap	2	2	4	6	8
		5-tap	PCDmin	PCDmin	PCDmin	PCDmin	PCDmin
	1:2 - 1:4		PCDmin	PCDmin	PCDmin	PCDmin	PCDmin

---

(1) The PCD value can be 1 in case all the data and synchronization signals are asserted and deasserted on the rising edge of the pixel clock.

Table 13-26. Pixel Clock Frequency Limitations - RGB16 and YUV4:2:2 Passive Matrix Display - Mono4

Min PCD Values			**Horizontal Resampling**				
			Off	**Up**	**1:1 - 1:2**	**1:2 - 1:3**	**1:3 - 1:4**
Vertical Resampling	Off		4	4	8	12	16
	Up		4	4	8	12	16
	1:1 - 1:2	3-tap	8	8	16	24	32
		5-tap	4xPCDmin	4xPCDmin	4xPCDmin	4xPCDmin	4xPCDmin
	1:2 - 1:4		4xPCDmin	4xPCDmin	4xPCDmin	4xPCDmin	4xPCDmin

Table 13-27. Pixel Clock Frequency Limitations - RGB16 and YUV4:2:2 Passive Matrix Display - Mono8

Min PCD Values			**Horizontal Resampling**				
			Off	**Up**	**1:1 - 1:2**	**1:2 - 1:3**	**1:3 - 1:4**
Vertical Resampling	Off		8	8	16	24	32
	Up		8	8	16	24	32
	1:1 - 1:2	3-tap	16	16	32	48	64
		5-tap	8xPCDmin	8xPCDmin	4xPCDmin	8xPCDmin	8xPCDmin
	1:2 - 1:4		8xPCDmin	8xPCDmin	4xPCDmin	8xPCDmin	8xPCDmin

Table 13-28. Pixel Clock Frequency Limitations - RGB16 and YUV4:2:2 Passive Matrix Display - Color

Min PCD Values			**Horizontal Resampling**				
			Off	**Up**	**1:1 - 1:2**	**1:2 - 1:3**	**1:3 - 1:4**
Vertical Resampling	Off		3	3	6	9	12
	Up		3	3	6	9	12
	1:1 - 1:2	3-tap	6	6	12	18	24
		5-tap	3xPCDmin	3xPCDmin	3xPCDmin	3xPCDmin	3xPCDmin
	1:2 - 1:4		3xPCDmin	3xPCDmin	3xPCDmin	3xPCDmin	3xPCDmin

NOTE: In case of RGB24 format, Figure 13-59 is still valid, except the PCDmin values which must be multiplied by two.

The PCDmin for vertical downsampling only is defined by the following equations:

Figure 13-59. PCDmin Formulas (V Down-Sampling Only)

$$h\_ratio = \frac{DISPC\_SIZE\_LCD[10:0]PLL}{DISPC\_VIDn\_SIZE[10:0]VID\_SIZE\_X}$$

$$v\_ratio = \frac{DISPC\_VIDn\_PICTURE\_SIZE[10:0]VID\_ORG\_SIZE\_Y}{DISPC\_VIDn\_SIZE[10:0]VID\_SIZE\_Y}$$

$$PCD\min = \frac{v\_ratio}{2 \times h\_ratio} \quad 1 < v\_ratio \le 2$$

$$PCD\min = \max\left(\frac{v\_ratio}{2 \times h\_ratio}, \frac{v\_ratio - 2}{2 \times (h\_ratio - 1)}\right) \quad 2 < v\_ratio \le 4$$

dss-E103

The PCDmin for horizontal downsampling only is defined by the following formula:

While downsampling by n, *PCDmin = n*

For H+V downsampling, the formula is the following:

PCDmin = max(PCDmin H only, PCDmin V only) as defined above

The refresh rate depends on the following parameters:

- Horizontal front porch (the DSS.DISPC_TIMING_H[19:8] HFP bit field)
- Horizontal back porch (the DSS.DISPC_TIMING_H[31:20] HBP bit field)
- Horizontal synchronization pulse width (the DSS.DISPC_TIMING_H[7:0] HSW bit field)
- Vertical front porch (the DSS.DISPC_TIMING_V[19:8] VFP bit field)
- Vertical back porch (the DSS.DISPC_TIMING_V[31:20] VBP bit field)
- Vertical synchronization pulse width (the DSS.DISPC_TIMING_V[7:0] VSW bit field)
- Number of lines per panel (the DSS.DISPC_SIZE_LCD[26:16] LPP bit field)
- Number of pixels per line (the DSS.DISPC_SIZE_LCD[10:0] PPL bit field)
- 4- or 8-bit interface for the passive matrix monochrome panel (the DSS.DISPC_CTRL[4] M8B bit)

The following bit fields define the behavior of the internal blocks:

- Spatial/temporal dithering logic enabled (DSS.DISPC_CTRL[7] ST_DITHER_EN bit)
- Spatial/temporal dithering logic number of frames (DSS.DISPC_CTRL[31:30] SPATIALTEMPORALDITHERFRAMES bit field). The default value of this bit field at reset time is 0x0, which is 1 frame only (spatial processing without temporal dithering). The possible values are 0x0 (one frame), 0x1 (two frames), and 0x2 (four frames). The number of frames is initialized before enabling the spatial/temporal dithering unit. The software must not change this bit field value while the spatial/temporal unit is enabled.

The following bit field defines the clock gating strategy:

- In active matrix mode, the pixel clock is always gated or only when valid data are present (the DSS.DISPC_CFG[0] PIXEL_GATED bit).

13.4.3.4.3 LCD Overlay

The following bit fields define the overlay attributes of the LCD output:

- Transparency color key (the DSS.DISPC_TRANS_COLOR0i register (i = 0))
- Transparency color key enable (the DSS.DISPC_CFG[10] TCK_LCD_EN bit)
- Transparency color key selection between the destination graphics transparency color key and the source video transparency color key (the DSS.DISPC_CFG[11] TCK_LCD_SELECTION bit)
- The default solid background color is defined in the DSS.DISPC_DEFAULT_COLOR_m[23:0] DEFAULT_COLOR bit field (i=0).
- Alpha blender Enable (DSS.DISPC_CFG[18] LCD_ALPHABLDR_EN)
- Global alpha blending values (DSS.DISPC_GLOBAL_ALPHA[23:16] VID2_GLOBAL_ALPHA and DSS.DISPC_GLOBAL_ALPHA[7:0] GFX_GLOBAL_ALPHA). The value 0xFF corresponds to 100% opaque and 0 to 100% transparent

---

NOTE: The destination graphics transparency color key is available only to the overlay with which the graphics pipeline is connected. The software must set the correct configuration of the LCD and digital overlays.

NOTE: When the alpha blender is enabled, the destination transparency color key is not available and the source transparency color key applies to the graphics pixels and not the video pixels.

When all of these fields are set to the appropriate values, set the DSS.DISPC_CTRL[5] GO_LCD bit to indicate that all shadow registers of the pipelines connected to the LCD output are latched by the hardware (only if the DSS.DISPC_CTRL[0] LCD_EN bit is already set to 1). If the LCD output is disabled, the new values will be updated when the DSS.DISPC_CTRL[0] LCD_EN bit will be set to 1.

13.4.3.4.4 LCD TDM

The following fields define the multiple cycle output configuration:

- First cycle (the DSS.DISPC_DATA_CYCLEk (k=0) register)
- Second cycle (the DSS.DISPC_DATA_CYCLEk (k=1) register)
- Third cycle (the DSS.DISPC_DATA_CYCLEk (k=2) register)
- Enable (the DSS.DISPC_CTRL[20] TDM_EN bit)
- Parallel mode (the DSS.DISPC_CTRL[22:21] TDM_PARALLEL_MODE field)
- Cycle format (the DSS.DISPC_CTRL[24:23] TDM_CYCLE_FMT field)
- Unused bits (the DSS.DISPC_CTRL[26:25] TDM_UNUSED_BITS field)

When all of these bit fields are set to the appropriate values, set the DSS.DISPC_CTRL[5] GO_LCD bit to indicate that all shadow registers of the pipelines connected to the LCD output are latched by the hardware (only if the DSS.DISPC_CTRL[0] LCD_EN bit is already set to 1). If the LCD output is disabled, the new values will be updated when the DSS.DISPC_CTRL[0] LCD_EN bit will be set to 1.

13.4.3.4.5 LCD Spatial/Temporal Dithering

The following bit fields define the LCD spatial/temporal dithering configuration:

- Number of frames (the DSS.DISPC_CTRL[31:30] SPATIAL_TEMPORAL_DITHER bit field) with:
  - 0x0 Spatial only (default value)
  - 0x1 Spatial + Temporal over two frames
  - 0x2 Spatial + Temporal over four frames
  - 0x3 Reserved
- Enable (the DSS.DISPC_CTRL[7] ST_DITHER_EN bit)
  - 0x0 Disabled (default value)
  - 0x1 Enabled

When all of these bit fields are set to the appropriate values, set the DSS.DISPC_CTRL[5] GO_LCD bit to indicate that all shadow registers of the pipelines connected to the LCD output are latched by the hardware (only if the DSS.DISPC_CTRL[0] LCD_EN bit is already set to 1). If the LCD output is disabled, the new values will be updated when the DSS.DISPC_CTRL[0] LCD_EN bit will be set to 1.

13.4.3.4.6 LCD Color Phase Rotation

The following bit fields define the color phase rotation configuration:

- Enable (the DSS.DISPC_CFG[15] CPR bit)
  - 0x0 Disabled (default value)
  - 0x1 Enabled
- Red 10-bit signed coefficients used by the color phase rotation matrix (the DSS.DISPC_CPR_COEF_R

register)

- Green 10-bit signed coefficients used by the color phase rotation matrix (the DSS.DISPC_CPR_COEF_G register)
- Blue 10-bit signed coefficients used by the color phase rotation matrix (the DSS.DISPC_CPR_COEF_B register)

The programmable color phase rotation block for the LCD output has nine 10-bit coefficients defined in the DSS.DISPC_CPR_COEF_R, DSS.DISPC_CPR_COEF_G, and DSS.DISPC_CPR_COEF_B, as described in Figure 13-60 through Figure 13-63.

Figure 13-60. Color Phase Rotation Matrix

$$\begin{bmatrix} Rout \\ Gout \\ Bout \end{bmatrix} = \frac{1}{256} * \left(\begin{bmatrix} RR & RG & RB \\ GR & GG & GB \\ BR & BG & BB \end{bmatrix} * \begin{bmatrix} Rin \\ Gin \\ Bin \end{bmatrix} \right)$$

dss-E105

Figure 13-61. Color Phase Rotation Matrix (R Component Only)

$$Riout = \frac{1}{256} * (RR * Rin + RG * Gin + RB * Bin)$$

dss-E106

Figure 13-62. Color Phase Rotation Matrix (G Component Only)

$$Gout = \frac{1}{256} * (GR * Rin + GG * Gin + GB * Bin)$$

dss-E107

Figure 13-63. Color Phase Rotation Matrix (B Component Only)

$$Bout = \frac{1}{256} * (BR * Rin + BG * Gin + BB * Bin)$$

dss-E104

When all of these bit fields are set to the appropriate values, set the DSS.DISPC_CTRL[5] GO_LCD bit to indicate that all shadow registers of the pipelines connected to the LCD output are latched by the hardware (only if the DSS.DISPC_CTRL[0] LCD_EN bit is already set to 1). If the LCD output is disabled, the new values will be updated when the DSS.DISPC_CTRL[0] LCD_EN bit will be set to 1.

13.4.3.4.6.1 Color Phase Rotation - Diagonal Matrix

The Color Phase Rotation feature is useful when using an LCD backlight that is not white. By using a correct configuration of the R, G and B coefficients of CPR, the color bias of the screen can be corrected. The following paragraphs give an example of configuration of the CPR feature.

The easiest example of CPR configuration is a diagonal matrix. This way, the output colors depends on one input color only. Figure 13-64 gives the example of a diagonal matrix and the corresponding equation of the output components.

Figure 13-64. Diagonal Matrix Configuration

$$\begin{bmatrix} Rout \\ Gout \\ Bout \end{bmatrix} = \frac{1}{256} * \left(\begin{bmatrix} RR & 0 & 0 \\ 0 & GG & 0 \\ 0 & 0 & BB \end{bmatrix} * \begin{bmatrix} Rin \\ Gin \\ Bin \end{bmatrix} \right)$$

$$\longrightarrow Rout = \frac{1}{256} * \left(RR * Rin \right)$$

$$\longrightarrow Gout = \frac{1}{256} * \left(GG * Gin \right)$$

$$\longrightarrow Bout = \frac{1}{256} * \left(BB * Bin \right)$$

dss-200

According to these 3 new equations, each output component only depends on the corresponding input color. The coefficients can easily be used to reduce the impact of a non-white backlight.

Let's take the example of a "blue" backlight. In this case, users have the feeling that a blue film has been added on the screen, and then each color seems to be "too much blue". The goal is then to reduce the "Blue" component and to keep the "Red" and "Green" ones unchanged. The following matrix can be used for a reduction by a half of the blue component. Figure 13-65 gives the corresponding matrix and equations for each component.

Figure 13-65. Example - Diagonal Matrix Configuration

$$\begin{bmatrix} Rout \\ Gout \\ Bout \end{bmatrix} = \frac{1}{256} * \left(\begin{bmatrix} 256 & 0 & 0 \\ 0 & 256 & 0 \\ 0 & 0 & 128 \end{bmatrix} * \begin{bmatrix} Rin \\ Gin \\ Bin \end{bmatrix} \right)$$

$$\longrightarrow Rout = \frac{1}{256} * \left(256 * Rin \right) \Rightarrow Rout = Rin$$

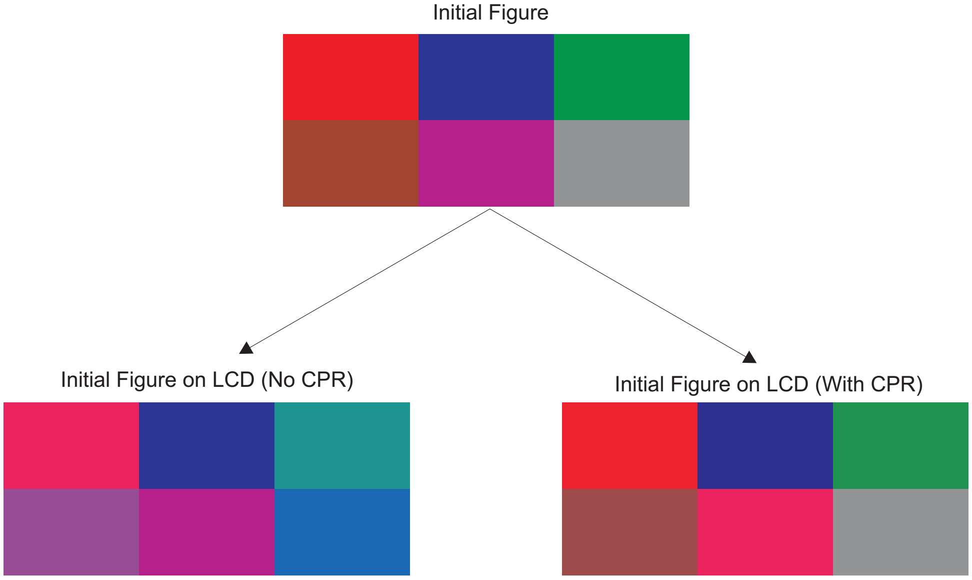
$$\longrightarrow Gout = \frac{1}{256} * \left(256 * Gin \right) \Rightarrow Gout = Gin$$

$$\longrightarrow Bout = \frac{1}{256} * \left(128 * Bin \right) \Rightarrow Bout = 0.5 * Bin$$

dss-201

Figure 13-66 shows the result of an image on a "blue" backlight screen with and without CPR.

Figure 13-66. Image With and Without CPR (Diagonal Matrix)



dss-202

A drawback of this diagonal matric is that the color reduction is linear. The contrast is then different from the initial image. It is then necessary to use the 6 other coefficients of the CPR matrix to better correct a non-white backlight. The goal is to find the correct coefficients that remove the color offset added by the non-white backlight.

13.4.3.4.6.2 Color Phase Rotation - Standard Matrix

In the following example, the LCD backlight adds an offset of 128 (B_offset) to the Blue component. Figure 13-67 shows an example of matrix that reduces the offset of the screen, the corresponding equations and the resulting output colors.

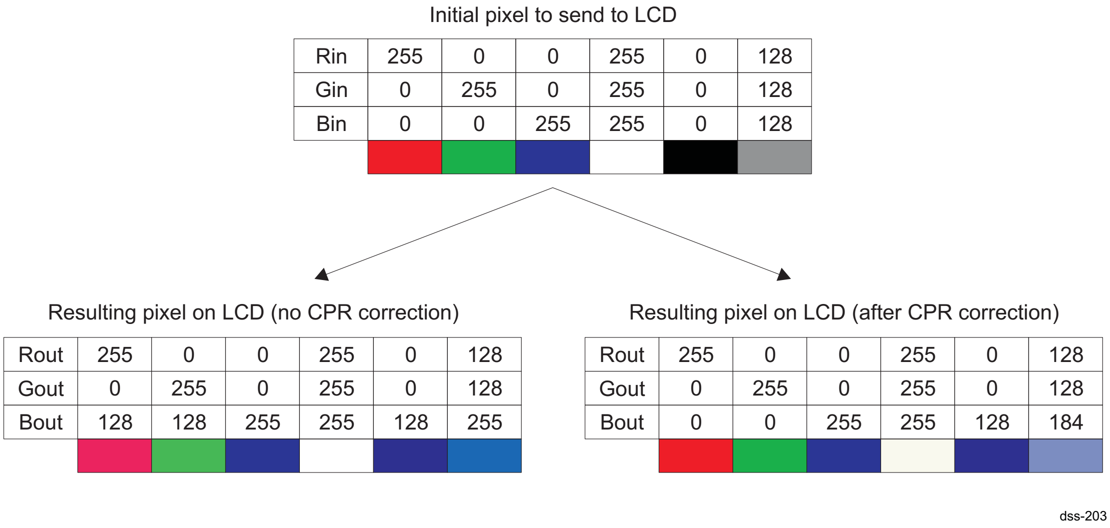
Figure 13-67. Example - Image With and Without CPR (Standard Matrix)

$$\begin{bmatrix} Rout \\ Gout \\ Bout \end{bmatrix} = \frac{1}{256} * \left(\begin{bmatrix} 256 & 0 & 0 \\ 0 & 256 & 0 \\ -129 & -129 & 370 \end{bmatrix} * \begin{bmatrix} Rin \\ Gin \\ Bin \end{bmatrix} \right)$$

$$\rightarrow \quad Rout = \frac{1}{256} * \left(256 * Rin \right)$$

$$\rightarrow \quad Gout = \frac{1}{256} * \left(256 * Gin \right)$$

$$\rightarrow \quad Bout = \frac{1}{256} * \left(-129 * Rin + -129 * Gin + 370 * Bin \right) + B\_offset$$



Initial pixel to send to LCD

Rin	255	0	0	255	0	128
Gin	0	255	0	255	0	128
Bin	0	0	255	255	0	128

Resulting pixel on LCD (no CPR correction)

Rout	255	0	0	255	0	128
Gout	0	255	0	255	0	128
Bout	128	128	255	255	128	255

Resulting pixel on LCD (after CPR correction)

Rout	255	0	0	255	0	128
Gout	0	255	0	255	0	128
Bout	0	0	255	255	128	184

This CPR matrix gives inputs and outputs very close. However, black cannot be corrected because of it's zero-components. No matter which coefficients are used in the matrix, the result will always be equal to the offset added by the LCD backlight.

13.4.3.5 TV Set-Specific Control Registers

The following registers define the digital output configuration:

- DSS.DISPC_CTRL
- DSS.DISPC_CFG
- DSS.DISPC_DEFAULT_COLOR_m (m=1)
- DSS.DISPC_TRANS_COLOR_m (m=1)
- DSS.DISPC_SIZE_DIG

The digital output is enabled/disabled by setting/resetting the DSS.DISPC_CTRL[1] DIGITAL_EN bit. A valid configuration must be set before the digital output can be enabled.

Perform the initialization sequence as follows:

1. Initialize the video encoder and the display controller configuration registers.
2. Set the DSS.DISPC_CTRL[6] GO_DIGITAL bit and the DSS.DISPC_CTRL[1] DIGITAL_EN bit to 1.
3. Wait for the first VSYNC pulse signal.
4. Clear the SYNC_LOST_DIGITAL interrupt by setting the DSS.DISPC_IRQSTS[15] SYNC_LOST_DIGITAL bit to 1.
5. Enable the SYNC_LOST_DIGITAL interrupt by setting the DSS.DISPC_IRQEN[15] SYNC_LOST_DIGITAL bit to 1.

13.4.3.5.1 Digital Timings

The following bit fields define the timing information:

- Data hold time (the DSS.DISPC_CTRL[19:17] HT bit field)
- Logic clock divisor (the DSS.DISPC_DIVISOR[23:16] LCD bit field)

The 8-bit pixel clock divider (DSS.DISPC_DIVISOR[23:16]) bit field is used to select the logic clock frequency. The LCD generates a range of pixel clock frequencies from FCK/1 to FCK/255, where FCK is the input functional clock of the display controller.

13.4.3.5.2 Digital Frame/Field Size

The following bit fields define the field size (frame if progressive mode):

- Number of lines per panel (the DSS.DISPC_SIZE_DIG[26:16] LPP bit field)
- Number of pixels per line (the DSS.DISPC_SIZE_DIG[10:0] PPL bit field)

13.4.3.5.3 Digital Overlay

The following bit fields define the overlay attributes of the digital output:

- Transparency color key (the DSS.DISPC_TRANS_COLOR_m register (m=1))
- Transparency color key enable (the DSS.DISPC_CFG[12] TCK_DIG_EN bit)
- Transparency color key selection between the destination graphics transparency color key and the source video transparency color key (the DSS.DISPC_CFG[13] TCK_DIG_SELECTION bit)
- The default solid background color is defined in the DSS.DISPC_DEFAULT_COLOR_m[23:0] DEFAULT_COLOR bit field (i=1).
- Alpha blender Enable (DSS.DISPC_CFG[19] TV_ALPHA_BLDR_EN)
- Global alpha blending values (DSS.DISPC_GLOBAL_ALPHA[23:16] VID2_GLOBAL_ALPHA and DSS.DISPC_GLOBAL_ALPHA[7:0] GFX_GLOBAL_ALPHA). The value 0xFF corresponds to 100% opaque and 0 to 100% transparent

NOTE: The destination graphics transparency color key is available only to the overlay with which the graphics pipeline is connected. The software must set the correct configuration of the LCD and digital overlays.

NOTE: When the alpha blender is enabled, the destination transparency color key is not available and the source transparency color key applies to the graphics pixels and not the video pixels.

When this bit field is set to the appropriate values, set the DSS.DISPC_CTRL[6] GO_DIGITAL bit to indicate that all shadow registers of the pipelines connected to the digital output are latched by the hardware (only if the DSS.DISPC_CTRL[1] DIGITAL_EN bit is already set to 1). If the digital output is disabled, the new values will be updated when the DSS.DISPC_CTRL[1] DIGITAL_EN bit will be set to 1.

13.4.4 RFBI Basic Programming Model

The RFBI programming model must be used for LCD display support only.

13.4.4.1 DISPC Control Registers

The following DISPC registers are used in RFBI mode:

- The STALL mode is selected by setting the DSS.DISPC_CTRL[11] STALL_MODE bit. The DSS.DISPC_CTRL[5] GO_LCD bit must not be set to 1, but the display controller configuration (DMA engine, pipelines associated to the LCD output,..) must be set before enabling the LCD output by setting the DSS.DISPC_CTRL[0] LCD_EN bit to 1.
- To enable the hardware handcheck to avoid underflow, the DSS.DISPC_CFG[16] FIFO_HAND_CHECK must be set to 1. The reset value of this bit is 0. The handcheck applies to the pipelines connected to the LCD output. It must be disabled before resetting the DSS.DISPC_CTRL[11]

STALL_MODE bit to 0. The new setting for the FIFO handcheck is used for the following frames.

NOTE: The LCD output is disabled at the end of the transfer of the frame. The software must reenable the LCD output to generate a new frame by setting the DSS.DISPC_CTRL[0] LCD_EN to 1. See Figure 13-68.

13.4.4.2 RFBI Control Registers

The following registers define the RFBI control registers:

- DSS.RFBI_CTRL
- DSS.RFBI_PIXEL_CNT
- DSS.RFBI_LINE_NUMBER

13.4.4.2.1 High Threshold

The DSS.RFBI_CTRL[6:5] HIGH_THR bit field is used to define the threshold to be used for the generation of the DMA request to receive data into the interconnect FIFO (24 x 32 FIFO depth) through the address of the register RFBI_DATA. It must be the size of the burst. The supported values are 4x32, 8x32 and 16x32. The system DMA receives the DMA request and is in charge of providing the correct number of bytes. If the DSS.RFBI_CTRL[7] DISABLE_DMA_REQ bit is reset, the DMA request is generated when there is enough room in the interconnect FIFO to accept the full burst. In case the RFBI receives writes L4 requests to the RFBI_DATA location when the interconnect FIFO is full, the request is not accepted. The RFBI waits for a free entry in the interconnect FIFO to accept the L4 request.

If the DSS.RFBI_CTRL[7] DISABLE_DMA_REQ bit is set, the DMA request is not generated. The threshold value is ignored.

NOTE: Software users can access the RFBI_DATA location without using the DMA request and without programming the high threshold value (backward mode).

13.4.4.2.2 Bypass Mode

Setting the DSS.RFBI_CTRL[1] BYPASS_MODE bit directly outputs the LCD controller output to the LCD panel. Resetting this bit directs the MPU module to send commands/parameters and data from the input video port FIFO.

13.4.4.2.3 Enable

Setting/resetting the DSS.RFBI_CTRL[0] EN bit enables/disables the RFBI module. The hardware resets the enable bit after all of the pixels are sent to the panel. The DSS.RFBI_PIXEL_CNT[31:0] PIXEL_CNT bit field value defines the number of pixels to send to the LCD panel. When the transfer is finished, the configuration used can be modified.

Table 13-29. RFBI Behavior

RFBI_CTRL[1] BYPASS_MODE bit value	RFBI_CTRL[0] ENABLE bit value	RFBI Behavior
0	0	L4 interconnect can write command/param/data and read data/status from the Remote Frame Buffer (RFB). L4 interconnect access can only be done to the CSx actually active
0	1	The DISPC sends pixels to the RFB.

The stall signal is asserted when the module is disabled. Through the L4 port, pixels can be sent to the LCD panel only when the pixel count has reach the value 0x0

NOTE: The LCD output is disabled at the end of the transfer of the frame. The software must reenable the LCD output to generate a new frame by setting the DSS.DISPC_CTRL[0] LCD_EN to 1. See Figure 13-68.

13.4.4.2.4 Configuration Selection

Setting the DSS.RFBI_CTRL[3:2] CONFIG_SELECT bit field selects the configuration number (1 or 0 if bits are set or reset). The registers associated with the configuration output the data to the LCD panel.

If both chip-selects are selected, the configuration for the first chip-select is used (except for the polarity of the RFBI_CS1 signal defined by the second configuration) and both devices connected to the CS signals are driven in parallel. In read mode, if both chip-selects are set, only RFBI_CS0 is asserted to read data from the device connected on RFBI_CS0. In write mode with two chip-selects selected, the RFBI can write to the two devices simultaneously.

13.4.4.2.5 ITE Bit

Set the DSS.RFBI_CTRL[4] ITE bit to start capturing the data from the display controller. This bit has no effect if the trigger mode is set to external. The display controller must be configured in the STALL mode to account for the RFBI_DISPC_STALL signal. Setting the trigger mode to external (DSS.RFBI_CONFIG_i[3:2] TRIGGER_MODE bit field set to 0x1 or 0x2) causes the DSS.RFBI_CTRL[4] ITE bit to be ignored. The corresponding chip-select must be selected when this bit is set by users.

The RFBI_DISPC_STALL signal is asserted when at least one of the following cases occur:

- Default status when no data to capture from the display controller
- High FIFO threshold reached
- End of the transfer (number of data to output)
- Reset of the RFBI module
- DSS.RFBI_CTRL[0] EN bit reset to 0x0

The RFBI_DISPC_STALL signal is deasserted when the DSS.RFBI_CTRL[0] EN bit is set to 0x1 and at least one of the following cases occur:

- Low FIFO threshold reached
- External TE occurs and the DSS.RFBI_CONFIG_i[3:2] TRIGGER_MODE bit field is set to 0x1 or 0x2 for automatic external trigger (start of the transfer, the FIFO pointers are reset, the FIFO is empty).
- DSS.RFBI_CTRL[4] ITE bit set to 0x1 by users (start of the transfer, the FIFO pointers are reset, the FIFO is empty).

13.4.4.2.6 Number of Pixels to Transfer

Setting the DSS.RFBI_PIXEL_CNT[31:0] PIXEL_CNT bit field value directs the application to indicate the number of pixels to be transferred to the LCD panel. The value can be changed only when the DSS.RFBI_CTRL[0] EN is reset.

During the transfer, the hardware decrements the register when a pixel is sent to the remote frame buffer. When the DSS.RFBI_CTRL[0] EN bit is set and a new value is written in the DSS.RFBI_PIXEL_CNT register when the current value in the register is a non-zero (the remaining number of pixels to transfer), the ongoing transfer is aborted.

From the L4 interconnect side, if the DSS.RFBI_CONFIG_i[10:9] CYCLE_FMT bit field is equal to 0x3 and the DSS.RFBI_CONFIG_i[8:7] L4_FMT bit field is equal to 0x0, an even number of write accesses to the data register must be performed before accessing any other register (CMD/PARAM/STATUS/READ).

When the DSS.RFBI_CONFIG_i[10:9] CYCLE_FMT bit field is 0x3 (2 pixels are sent over 3 cycles), the number of pixels to be programmed in the DSS.RFBI_PIXEL_CNT[31:0] PIXEL_CNT bit field must be a multiple of 2. If another CYCLE_FMT is used, the value for PIXEL_CNT can be odd or even. This constraint is valid for data provided on the L4 interconnect port and from the display controller.

If the DSS.RFBI_CONFIG_i[10:9] CYCLE_FMT bit field is equal to 0x3, the DSS.RFBI_CONFIG_i[8:7] L4_FMT bit field is equal to 0, and back-to-back register write is processed. The following registers should be written after the first data: RFBI_CMD, RFBI_PARAM, RFBI_READ, and RFBI_STS. The whole data transfer must first be performed before being able to write to any other registers (RFBI_CMD, RFBI_PARAM, RFBI_READ, and RFBI_STS).

13.4.4.2.7 Programmable Line Number

When the trigger mode is set to external trigger mode with HSYNC and VSYNC or the TE, hardware resets the line counter when the VSYNC occurs and, after a programmable number of lines (the HSYNC pulse occurs for every line), the transfer to the LCD panel begins. When the programmable line number is 0, only the VSYNC pulse indicates the beginning of the transfer in both modes: HSYNC/VSYNC and TE (logical OR operation between HSYNC and VSYNC).

13.4.4.3 RFBI Configuration

The following registers define the RFBI configuration:

- DSS.RFBI_SYSCONFIG
- DSS.RFBI_SYSSTS
- DSS.RFBI_CONFIG_0 (configuration 0) and DSS.RFBI_CONFIG_1 (configuration 1)
- DSS.RFBI_VSYNC_WIDTH
- DSS.RFBI_HSYNC_WIDTH

The configuration register for one configuration can be accessed only when the configuration is not in use (based on the value of the RFBI_CTRL[3:2] CONFIG_SELECT bit field).

13.4.4.3.1 Parallel Mode

The DSS.RFBI_CONFIG_i[1:0] PARALLEL_MODE bit field (where i = 0, 1) defines the width of the interface (8-, 9-, 12-, or 16-bit parallel).

13.4.4.3.2 Trigger Mode

Setting the DSS.RFBI_CONFIG_i[3:2] TRIGGER_MODE bit field configures the trigger on the external TE signal (RFBI_TE_VSYNC), or external with VSYNC/HSYNC with the programmable number of HSYNCs to begin the transfer in both cases or the internal programmable DSS.RFBI_CTRL[4] ITE bit.

13.4.4.3.3 VSYNC Pulse Width (Minimum Value)

The DSS.RFBI_VSYNC_WIDTH[15:0] MIN_VSYNC_PULSE_WIDTH bit field defines the minimum number of L4 clock cycles of the VSYNC pulse for detection on VSYNC. It allows differentiation between VSYNC and HSYNC, which are ORed on the same signal and is also used in the VSYNC/HSYNC mode on the two separate input lines.

- The VSYNC pulse width must be at least equal to two L4 cycles when HSYNC is not present.
- The VSYNC pulse width must be at least equal to four L4 cycles when HSYNC is present.

13.4.4.3.4 HSYNC Pulse Width (Minimum Value)

The DSS.RFBI_HSYNC_WIDTH[15:0] MIN_HSYNC_PULSE_WIDTH bit field defines the minimum number of L4 clock cycles of the HSYNC pulse for detection on HSYNC. It allows differentiation between VSYNC and HSYNC, which are ORed on the same signal, and is also used in the VSYNC/HSYNC mode on the separate two input lines. The HSYNC pulse width must always be at least equal to two L4 cycles to be detected.

13.4.4.3.5 Cycle Format

Setting the DSS.RFBI_CONFIG_i[10:9] CYCLE_FMT bit field (with i = 0, 1) defines which registers are used to format the data in the interconnect FIFO with the appropriate number of bits (starting from the LSB) and with the alignment on the interface as follows:

- DSS.RFBI_DATA_CYCLE_i (if DSS.RFBI_CONFIG_i[10:9] CYCLE_FMT bit field = 00) only

  or

- DSS.RFBI_DATA_CYCLE1_i and DSS.RFBI_DATA_CYCLE2_i (if DSS.RFBI_CONFIG_i[10:9] CYCLE_FMT bit field = 01)

  or

- DSS.RFBI_DATA_CYCLE1_i, DSS.RFBI_DATA_CYCLE2_i, and DSS.RFBI_DATA_CYCLE3_i (if DSS.RFBI_CONFIG_i[10:9] CYCLE_FMT bit field = 10)

The data from the display controller and from the L4 interconnect are formatted based on the configuration of the DSS.RFBI_DATA_CYCLE_i registers.

13.4.4.3.6 Unused Bits

Based on the configuration, the undefined bits for each cycle are defined with the previous values of the bits at the same position in the previous cycle, 0s, or 1s (the unused bits can be at any position). The DSS.RFBI_CONFIG_i[12:11] UNUSEDBITS bit field (with i = 0, 1) is used.

13.4.4.3.7 RFBI Timings

The timing registers for one configuration can be accessed only when the configuration is not in use (based on the value of the DSS.RFBI_CTRL[3:2] CONFIG_SELECT bit field). Granularity is defined using the DSS.RFBI_CONFIG_i[4] TIME_GRANULARITY bit. This feature allows the extension of programmable ranges of timing parameters for the RFBI interface. Refer to Table 13-30 for the bits configuration values.

- Chip-select assertion/deassertion time

  RFBI_A0 setup time to chip-select assertion is assured by the programmable chip-select assertion time from the start access time:
  DSS.RFBI_ONOFF_TIMEi[3:0] CS_ONTIME bit field (with i = 0, 1).

  The chip-select deassertion time from the start access time is programmable:
  DSS.RFBI_ONOFF_TIMEi[9:4] CS_OFFTIME bit field (with i = 0, 1)

CAUTION

Configuring DSS.RFBI_ONOFF_TIMEi[3:0] CS_ONTIME = DSS.RFBI_ONOFF_TIMEi[9:4] CS_OFFTIME = 0 (with i = 0, 1) is not supported and must be avoided. This configuration creates contention on the bus and progressively damages the LCD panel.

- Chip-select pulse width

  The total chip-select pulse width is the time when write cycle time or read cycle time has completed and is programmable:
  DSS.RFBI_CYCLE_TIMEi[17:12] CS_PULSE_WIDTH bit field (with i = 0, 1)

  It applies on the read-to-write, write-to-read, read-to-read, and write-to-write access based on:

  - The DSS.RFBI_CYCLE_TIMEi[19] RR_EN bit: Read-to-read access
  - The DSS.RFBI_CYCLE_TIMEi[20] WW_EN bit: Write-to-write access
  - The DSS.RFBI_CYCLE_TIMEi[18] RW_EN bit: Read-to-write access
  - The DSS.RFBI_CYCLE_TIMEi[21] WR_EN bit: Write-to-read access

  By default, it applies to any access (read-to-read, read-to-write, write-to-read, write-to-write) when the chip-select changes.

- Access time

The total access time is the time from when A0 becomes valid until data are sampled before deasserting the RE signal; access time is programmable:
DSS.RFBI_CYCLE_TIMEi[27:22] ACCESS_TIME bit field (with i = 0, 1)

When reading the data on the bus, the data are sampled at the end of the access time, which occurs before the end of the read off time (DSS.RFBI_ONOFF_TIMEi[29:24] RE_OFFTIME, with i = 0, 1).

- Write enable cycle time

  The total write enable cycle time is the time from when A0 becomes valid until write cycle completion; the write enable cycle time is programmable:
  The DSS.RFBI_CYCLE_TIMEi[5:0] WE_CYCLE_TIME bit field (with i = 0, 1)
- Write enable assertion/deassertion time

  The WE assertion delay time from start access time is programmable:
  DSS.RFBI_ONOFF_TIMEi[13:10] WE_ONTIME bit field (with i = 0, 1)

  The WE deassertion delay time from the start access time is programmable:
  DSS.RFBI_ONOFF_TIMEi[19:14] WE_OFFTIME bit field (with i = 0, 1)
- Read enable cycle

  The total read enable cycle time is the time when A0 becomes valid until read cycle completion; the read enable cycle time is programmable:
  The DSS.RFBI_CYCLE_TIMEi[11:6] RECYCLE_TIME bit field (with i = 0, 1)
- Read enable assertion/deassertion time

  The RE assertion delay time from the start access time is programmable:
  DSS.RFBI_ONOFF_TIMEi[23:20] RE_ONTIME bit field (with i = 0, 1)

  The RE deassertion delay time from the start access time is programmable:
  DSS.RFBI_ONOFF_TIMEi[29:24] RE_OFFTIME bit field (with i = 0, 1)

At cycle time completion (read access or write access) all control signals (RFBI_CSi, RFBI_WR, and RFBI_RD, with i = 0, 1) are deasserted regardless of their deassertion time parameter values, if they are not deasserted already.

However, an exception to this forced deassertion exists when a pipelined request to the same chip-select or to a different chip-select is pending. Also, a control signal with deassertion time parameters equal to the cycle time parameter is not necessarily deasserted when a pipelined request to the same chip-select or different chip-select is pending. This prevents any unnecessary glitch transitions.

If no inactive cycles are required between successive accesses to the same chip-select (the DSS.RFBI_CYCLE_TIMEi[17:12] CS_PULSE_WIDTH bit field = 0, with i = 0, 1), and if assertion time parameters associated with the following access equal 0, the asserted control signals (RFBI_CSi, RFBI_WR, and RFBI_RD, with i = 0, 1) stay asserted. This is applicable to any read/write-to-read/write access combination.

Table 13-30 summarizes the configurations values for each timing bit.

Table 13-30. RFBI Timings Configuration

Configuration bits[1]	Granularity[2]	
	one	two
DSS.RFBI_ONOFF_TIMEi[3:0] CS_ONTIME	0 to 15	0 to 30
DSS.RFBI_ONOFF_TIMEi[9:4] CS_OFFTIME	0 to 63	0 to 126
DSS.RFBI_CYCLE_TIMEi[17:12] CS_PULSE_WIDTH	0 to 63	0 to 126
DSS.RFBI_CYCLE_TIMEi[27:22] ACCESS_TIME	0 to 63	0 to 126
DSS.RFBI_CYCLE_TIMEi[5:0] WE_CYCLE_TIME	0 to 63	0 to 126
DSS.RFBI_ONOFF_TIMEi[13:10] WE_ONTIME	0 to 15	0 to 30
DSS.RFBI_ONOFF_TIMEi[19:14] WE_OFFTIME	0 to 63	0 to 126
DSS.RFBI_CYCLE_TIMEi[11:6] RECYCLE_TIME	0 to 63	0 to 126
DSS.RFBI_ONOFF_TIMEi[23:20] RE_ONTIME	0 to 15	0 to 30

[1] Where i = 0 or 1.

[2] Number of L4Clk cycles. The granularity can be configured using the DSS.RFBI_CONFIG_i[4] TIME_GRANULARITY bit.

Table 13-30. RFBI Timings Configuration (continued)

Configuration bits[1]	Granularity[2]	
	one	two
DSS.RFBI_ONOFF_TIMEi[29:24] RE_OFFTIME	0 to 63	0 to 126

13.4.4.3.8 RFBI State-Machine

Referring to Table 13-18, the signals RFBI_A0, RFBI_RD, and RFBI_WR are asserted/deasserted based on the register accessed (DSS.RFBI_CMD, DSS.RFBI_PARAM, DSS.RFBI_DATA, DSS.RFBI_READ, and DSS.RFBI_STS). When the DSS.RFBI_SYSSTS[8] BUSY bit is set by hardware, any access to the registers is stalled, except for the RFBI_DATA register.

The DSS.RFBI_SYSSTS[9] BUSY_RFBI_DATA bit indicates whether there are still pending data in the interconnect FIFO associated with the register RFBI_DATA only.

- Command register

  Write a command at a time by writing in the DSS.RFBI_CMD register. If the previous command is not processed, the DSS.RFBI_SYSSTS[8] BUSY bit is set by hardware and the access to writing a new command is stalled.

- Parameter register

  Write a parameter at a time by writing in the DSS.RFBI_PARAM register.

  If the previous parameter is not processed, the DSS.RFBI_SYSSTS[8] BUSY bit is set by hardware and the access to writing a new parameter is stalled.

- Data register

  Write one or two pixels at a time by writing in the RFBI_DATA register (when DSS.RFBI_CONFIG_i[10:9] CYCLE_FMT = 0x3 with i = 0, 1, two pixels must be written contiguously, no other access to RFBI registers except DSS.RFBI_DATA is allowed).

  The pixels are formatted based on the specified cycle format. If two pixels are written into the 32-data register, the DSS.RFBI_CONFIG_i[8:7] L4_FMT bit field indicates the number of pixels for each L4 access to the register and the order of the pixels

  If the previous data are not processed, the DSS.RFBI_SYSSTS[8] BUSY bit is set by hardware and any access for writing new data is stalled. When the DSS.RFBI_SYSSTS[8] BUSY bit is reset by hardware, the access is not stalled.

- Read/status register

  Send through the command and parameter registers the correct information to receive data in the data or status register. The read data from the LCD panel is initiated by writing into the DSS.RFBI_READ or DSS.RFBI_STS registers. In this case, the DSS.RFBI_SYSSTS[8] BUSY bit is set until the data are available in the register.

  When the DSS.RFBI_SYSSTS[8] BUSY bit is set by hardware, the read or write access is stalled until the register is updated with a new value from the LCD panel. To avoid the stall, the software can poll the DSS.RFBI_SYSSTS[8] BUSY bit until it is reset by hardware. To receive the data, send the appropriate command/parameters.

13.4.4.3.9 RFBI Configuration Flow Charts

The RFBI configuration depends on the trigger mode used by the application. The available trigger modes are:

- Internal trigger mode when setting the DSS.RFBI_CONFIG_i[3:2] TRIGGER_MODE bit field to 0x0
- External trigger mode:
  - TE external trigger mode when setting the DSS.RFBI_CONFIG_i[3:2] TRIGGER_MODE bit field to 0x1
  - HSYNC/VSYNC external trigger mode when setting the DSS.RFBI_CONFIG_i[3:2] TRIGGER_MODE bit field to 0x2

Figure 13-68 gives an example of how to program and use the RFBI module:

Figure 13-68. How to Use RFBI

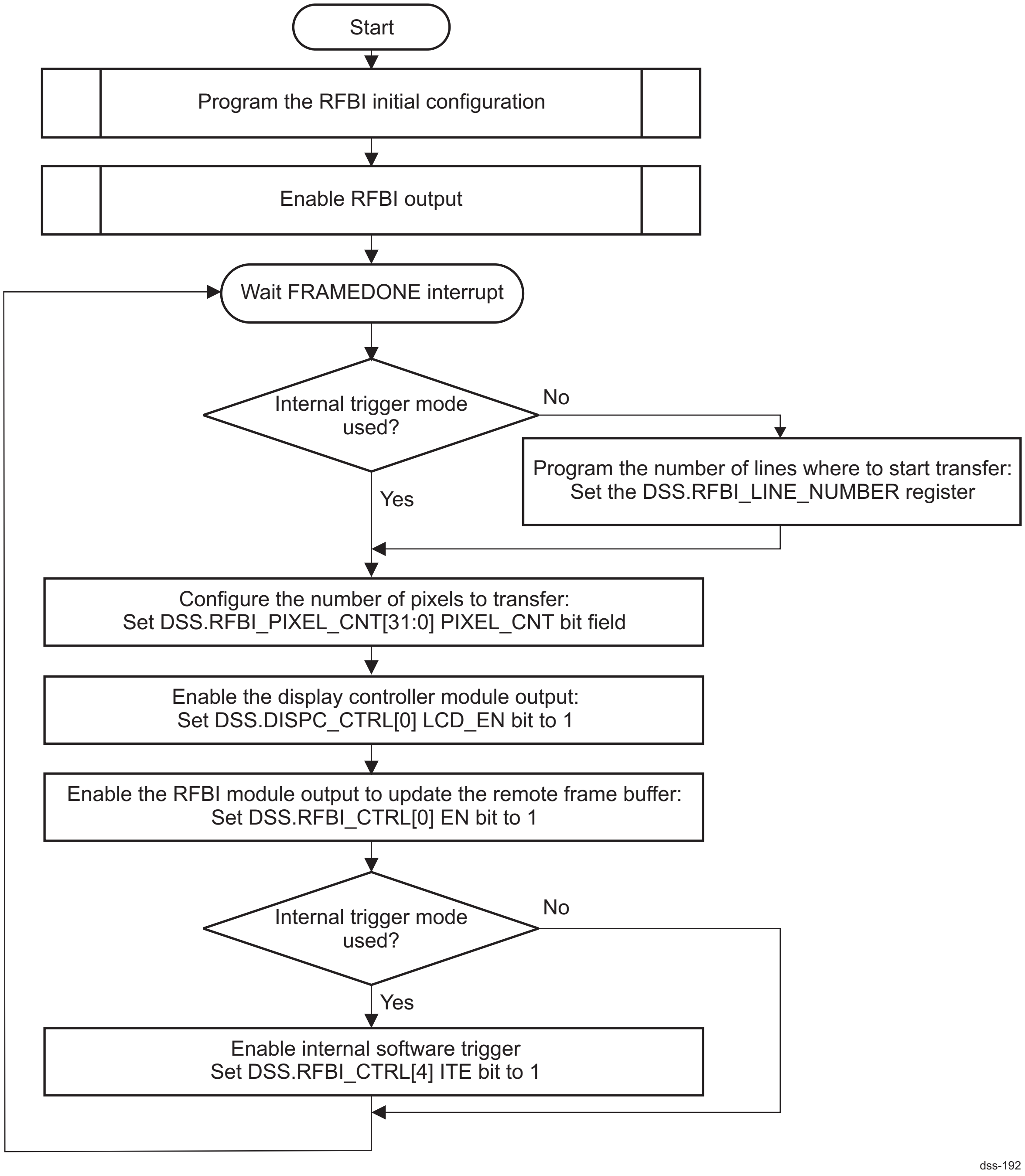


Figure 13-69 details how to configure the RFBI registers:

Figure 13-69. RFBI Initial Configuration

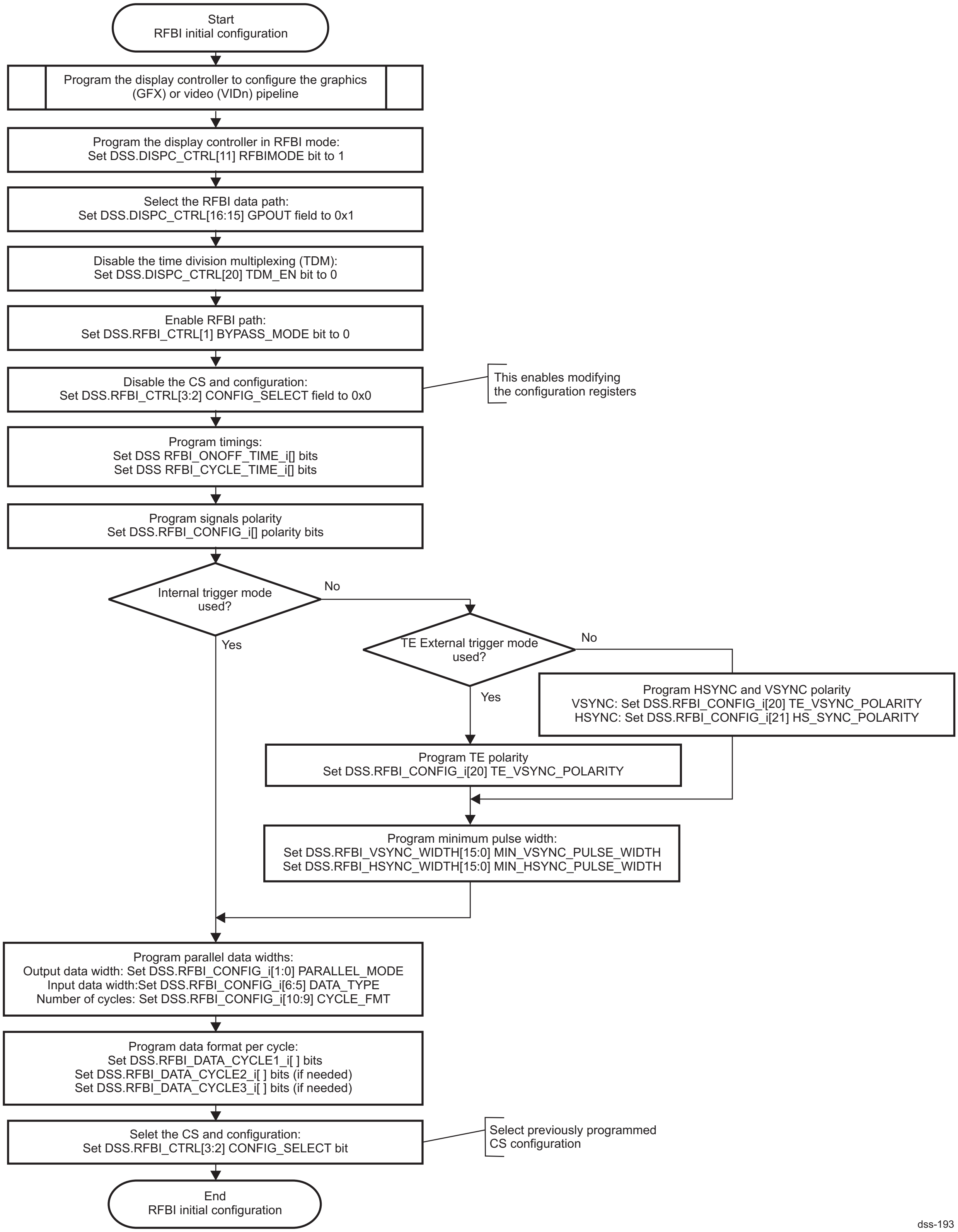
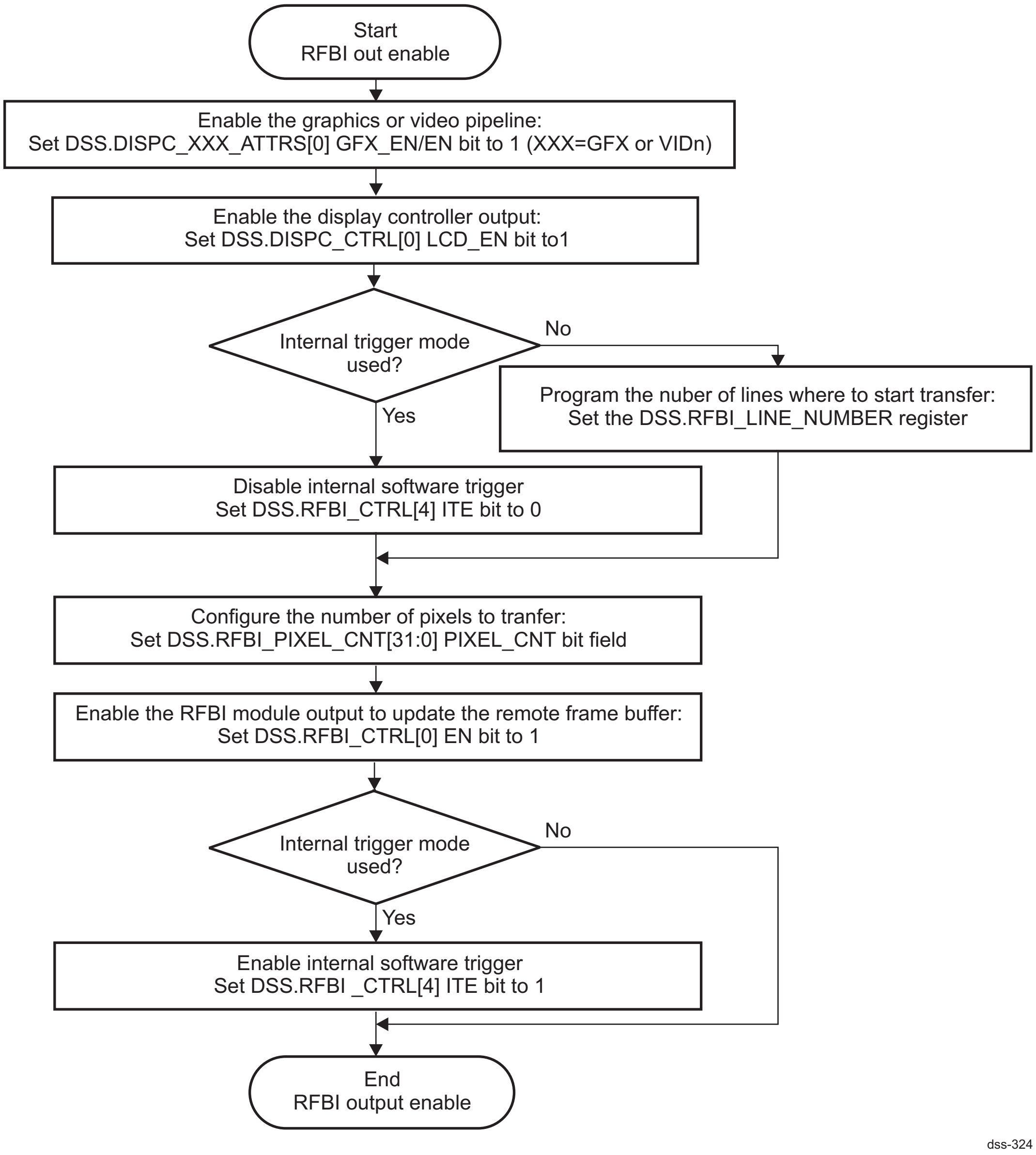


Figure 13-70 describes how to enable the RFBI module.

Figure 13-70. RFBI Output Enable



13.5 Use Cases

This section gives some generic use cases and tips for setting the modules of the display subsystem.

13.5.1 How to Configure the Scaling Unit in the DISPC Module

This section describes the scaling capability of the display controller (DISPC). The scaling unit is a part of the video pipeline is used when transferring pixels from the system memory (SDRAM or on-chip SRAM) to the LCD panel or the TV set. The scaling unit consists of two scaling blocks: The vertical scaling block followed by the horizontal scaling block. The input pixel format is RGB24. In case the pixel format in system memory is not RGB, the color space conversion unit in front of the scaling unit converts the YUV pixels into RGB pixels. The two scaling units are independent: Neither of them, only one, or both can be used simultaneously

13.5.1.1 Filtering

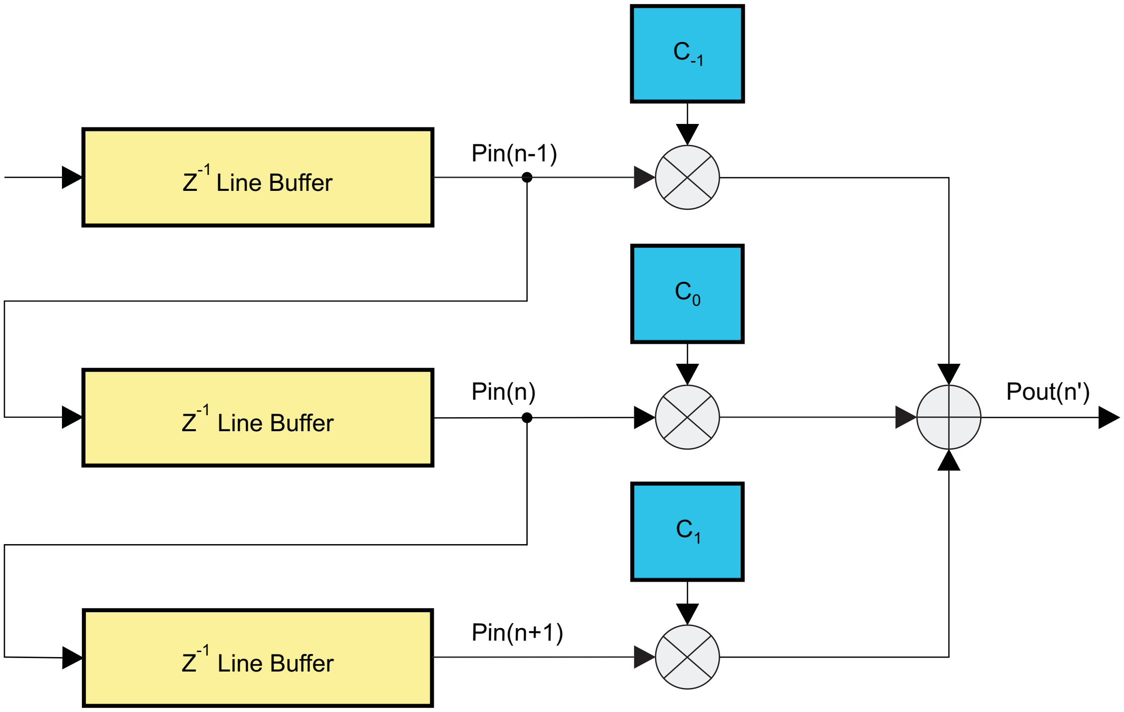
The scaling is used to down-scale, up-scale, or process the image while keeping the same size. It is applied independently horizontally and vertically. The same filtering applies for each color component (R, G, or B).

13.5.1.1.1 Vertical Filtering

The vertical filtering unit is based on a poly-phase rotation architecture with eight phases and three taps. That means that 24 coefficients are programmable

The vertical 3-tap filtering macro architecture is shown in Figure 13-71.

Figure 13-71. Vertical Filtering Macro Architecture (Three Taps)



dss-112

For the 3-tap vertical up/downsampling the equation is (with the example of R component):

$$Rout(n) = \left(\sum_{i=-1}^{i=1} Ci(\Phi) \times Rin(n+i) \right) >> 7 \tag{12}$$

dss-E067

Legend:

Rout: R component output

Ci():Vertical FIR coefficients

Rin: R component input

The line (n+1) is older than line (n).

NOTE: If the 5-tap resizer is used for RGB16 and YUV4:2:2 picture formats, the width of the input picture must be a multiple of 2 pixels and more than 5 pixels. This leads to the following register configuration:

DISPC_VIDn_ATTRS[21] VERTICAL_TAPS == 1

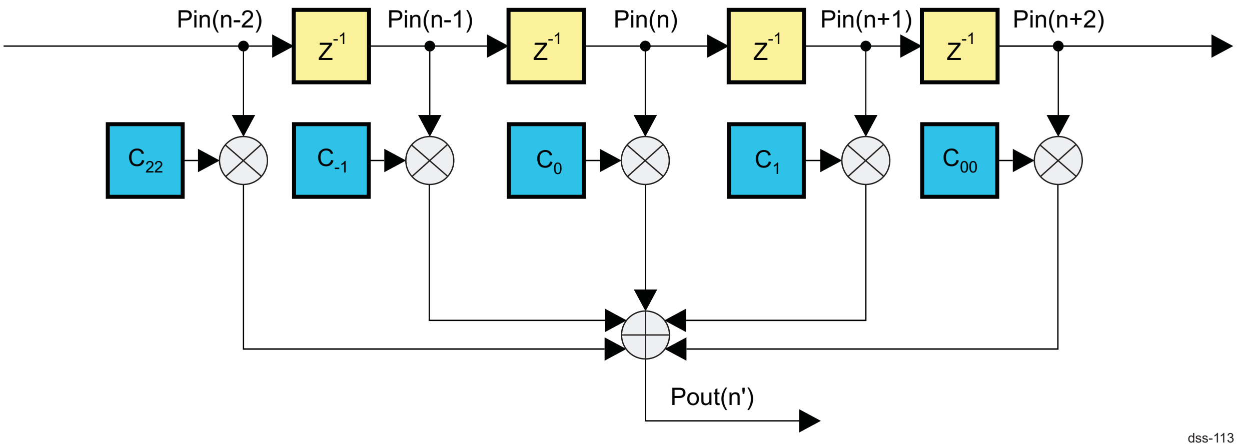
DISPC_VIDn_PICTURE_SIZE[10:0] VID_ORG_SIZE_X 4 and even

The programmable three coefficients of the poly-phase filters are signed 8-bit values (except for the central coefficient $C_0()$, which is unsigned).

The vertical filtering unit can be configured to support five taps.

The vertical 5-tap filtering macro architecture is shown in Figure 13-72.

Figure 13-72. Vertical Filtering Macro Architecture (Five Taps)



For the 5-tap vertical up/downsampling the equation is (with the example of R component):

$$Rout(n) = \left(\sum_{i=-2}^{i=2} Ci(\Phi) \times Rin(n+i) \right) >> 7 \qquad (13)$$

dss-E066

Legend:

Rout: R component output

Ci():Vertical FIR coefficients with $C_{+2}()=C_{00}()$ and $C_{-2}()=C_{22}()$

Rin: R component input

The line (n+1) is older than line (n).

The programmable five coefficients of the poly-phase filters are signed 8-bit values (except for the central coefficient C0(), which is unsigned).

In case of three taps, the memory lines are merged into three lines instead of six lines (one line is used as a cache line).

The first line is duplicated to fill up the two first lines (3-tap configuration) and the three first lines (5-tap configuration).

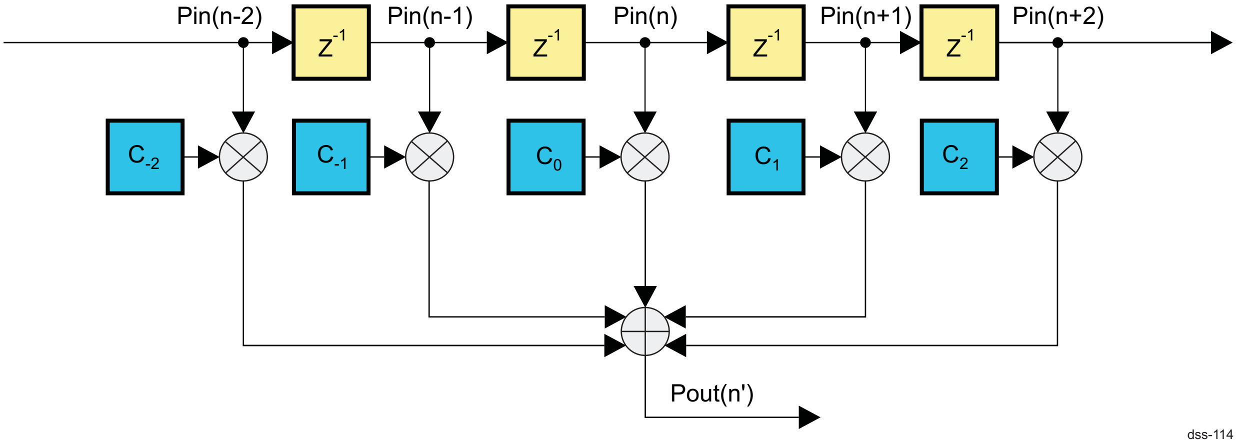
The last line is duplicated if the scaling logic requires loading of more lines and the last line has been reached

13.5.1.1.2 Horizontal Filtering

The horizontal filtering unit is based on a poly-phase rotation architecture with eight phases and five taps. That means that 40 coefficients are programmable.

The horizontal filtering macro architecture is shown in Figure 13-73.

Figure 13-73. Horizontal Filtering Macro Architecture (Five Taps)



For the 5-tap horizontal up/downsampling, the equation is (with the example of R component):

$$\mathrm{Rout}(n) = \left(\sum_{i=-3}^{i=3} Ci(\Phi) \times \mathrm{Rin}(n+i) \right) >> 7 \tag{14}$$

dss-E115

Legend:

Rout: R component output

Ci():Vertical FIR coefficients

Rin: R component input

The line (n+1) is older than line (n).

To horizontally and vertically filter the video layer, the phase is calculated separately. The programmable coefficients of the poly-phase filters are signed 8-bit values (except for the central coefficient C_0(), which is unsigned).

The first pixel is duplicated to fill up the three first pixel-buffers (5-tap configuration). The last pixel is duplicated if the scaling logic requires loading of more pixels and the last pixel has been reached

13.5.1.2 Scaling Algorithms

The up/downsampling finite state machines (FSM) below are detailed in this section.

Figure 13-74 presents the vertical up/downsampling FSM.

Figure 13-74. Vertical Up-/Down-Sampling Algorithm

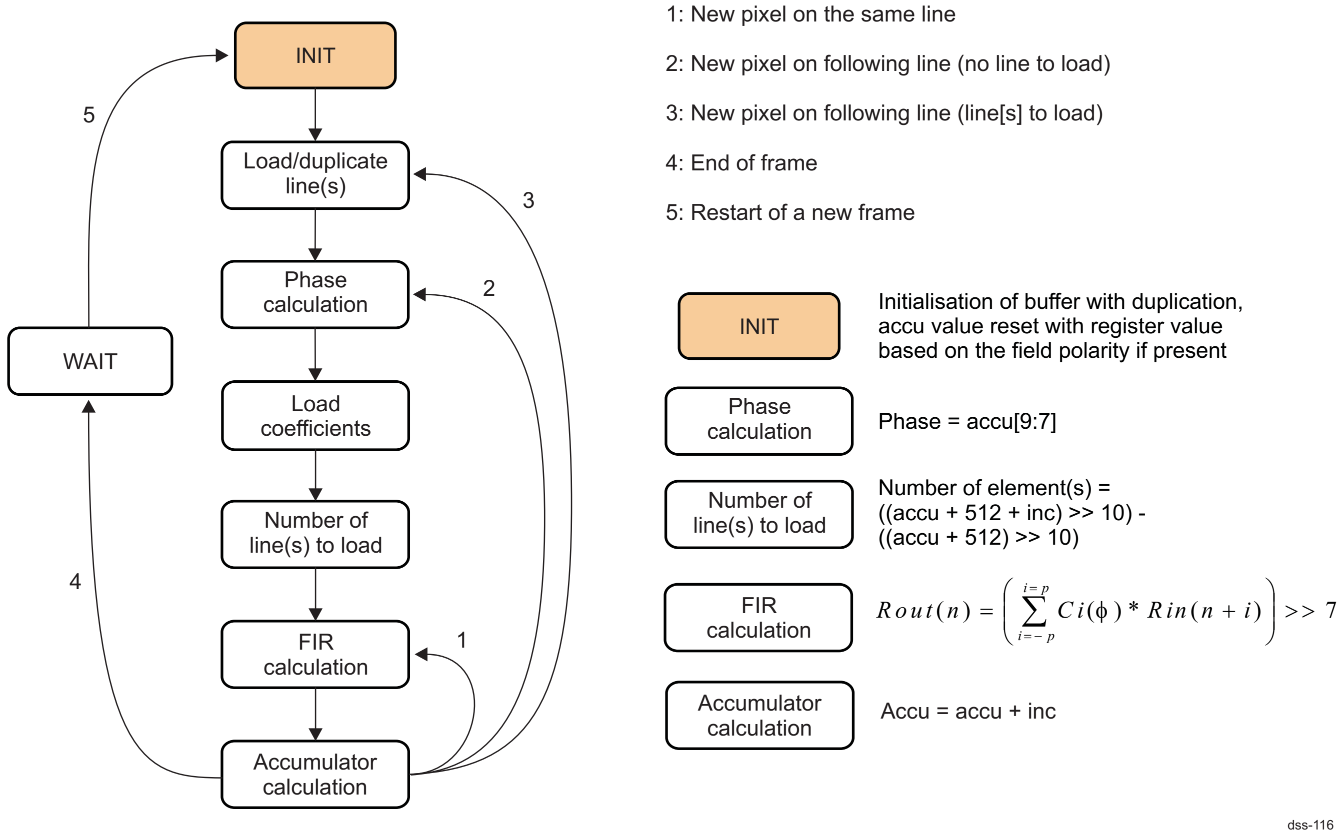
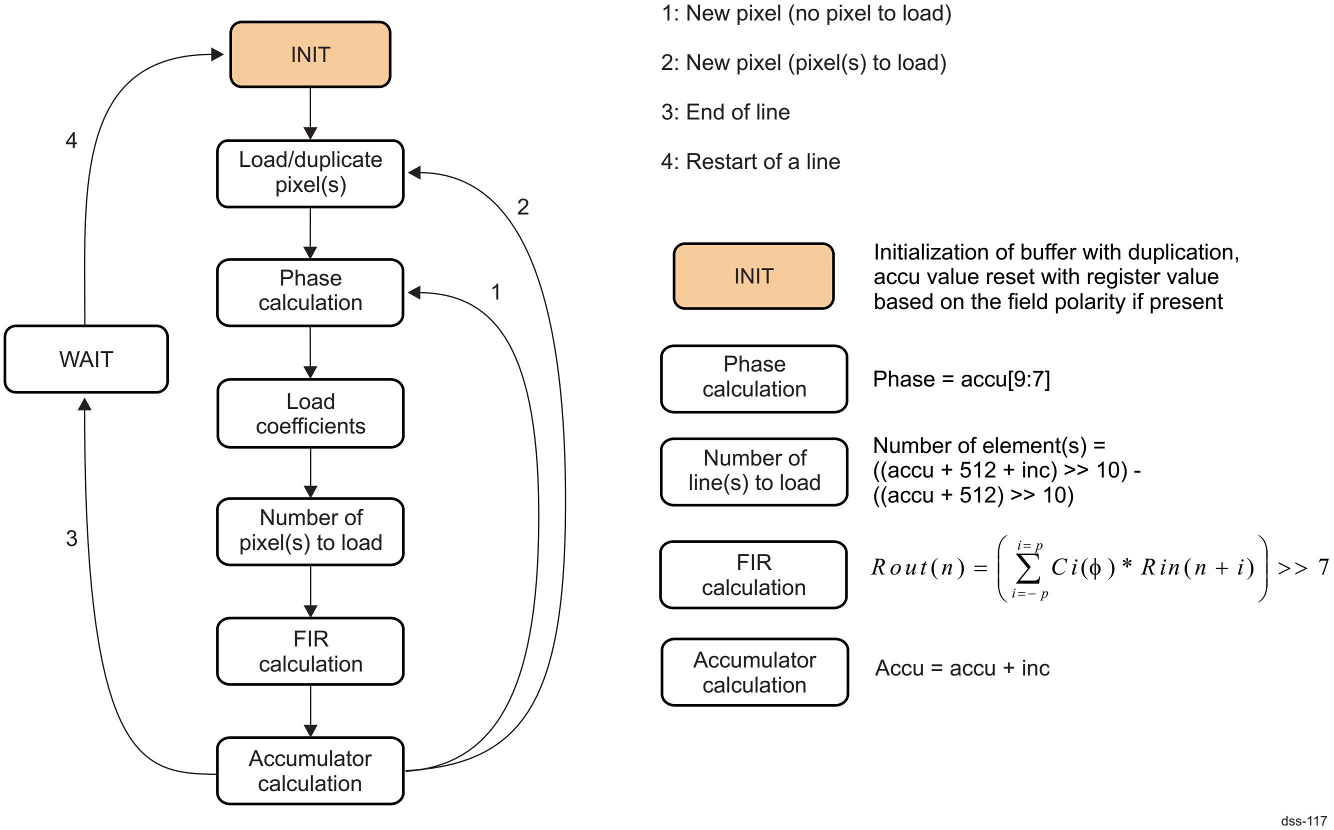


Figure 13-75 presents the horizontal up/downsampling FSM.

Figure 13-75. Horizontal Up-/Down-Sampling Algorithm



13.5.1.3 Scaling Settings

NOTE:

- In this section, the screen word refers to LCD panel or TV set.
- n indicates pipeline 0 or 1 because there are two video pipelines in the DISPC.

13.5.1.3.1 Register List

The following registers define the scaling registers for the video layer n configuration:

- DSS.DISPC_VIDn_BAj
- DSS.DISPC_VIDn_ATTRS
- DSS.DISPC_VIDn_FIR
- DSS.DISPC_VIDn_ACCUl
- DSS.DISPC_VIDn_FIR_COEF_H_i
- DSS.DISPC_VIDn_FIR_COEF_HV_i
- DSS.DISPC_VIDn_FIR_COEF_Vi

Table 13-31 lists the registers for programming the vertical FIR coefficients (3-tap configuration).

Table 13-31. Vertical FIR Coefficients Corresponding Table (3-Tap Configuration)

$C_x()$	FIR_VC$_x$()
$C_{-1}()$	FIR_VC$_2$()
$C_0()$	FIR_VC$_1$()

Table 13-31. Vertical FIR Coefficients Corresponding Table (3-Tap Configuration) (continued)

C$_{X}$()	**FIR_VC$_{X}$()**
C$_{1}$()	FIR_VC$_{0}$()

The corresponding registers for programming the vertical FIR coefficients (3-tap configuration) are:

- FIR_VC$_{2}$() = DSS.DISPC_VIDn_FIR_COEF_HV_i[31:24] FIR_VC2
- FIR_VC$_{1}$() = DSS.DISPC_VIDn_FIR_COEF_HV_i[23:16] FIR_VC1
- FIR_VC$_{0}$() = DSS.DISPC_VIDn_FIR_COEF_HV_i[15:8] FIR_VC0

Table 13-32 lists the registers for programming the vertical FIR coefficients (5-tap configuration).

Table 13-32. Vertical FIR Coefficients Corresponding Table (5-Tap Configuration)

C$_{X}$()	**FIR_VC$_{X}$()**
C$_{22}$()	FIR_VC$_{22}$()
C$_{-1}$()	FIR_VC$_{2}$()
C$_{0}$()	FIR_VC$_{1}$()
C$_{1}$()	FIR_VC$_{0}$()
C$_{00}$()	FIR_VC$_{00}$()

The corresponding registers for programming the vertical FIR coefficients (5-tap configuration) are:

- FIR_VC$_{22}$() = DSS.DISPC_VIDn_FIR_COEF_Vi[15:8] FIR_VC22
- FIR_VC$_{2}$() = DSS.DISPC_VIDn_FIR_COEF_HV_i[31:24] FIR_VC2
- FIR_VC$_{1}$() = DSS.DISPC_VIDn_FIR_COEF_HV_i[23:16] FIR_VC1
- FIR_VC$_{0}$() = DSS.DISPC_VIDn_FIR_COEF_HV_i[15:8] FIR_VC0
- FIR_VC$_{00}$() = DSS.DISPC_VIDn_FIR_COEF_Vi[7:0] FIR_VC00

Table 13-33 lists the registers for programming the horizontal FIR coefficients (5-tap configuration).

Table 13-33. Horizontal FIR Coefficients Corresponding Table (5-Tap Configuration)

C$_{X}$()	**FIR_HC$_{X}$()**
C$_{-2}$()	FIR_HC$_{4}$()
C$_{-1}$()	FIR_HC$_{3}$()
C$_{0}$()	FIR_HC$_{2}$()
C$_{1}$()	FIR_HC$_{1}$()
C$_{2}$()	FIR_HC$_{0}$()

The corresponding registers for programming the vertical FIR coefficients (3-tap configuration) are:

- FIR_HC$_{4}$() = DSS.DISPC_VIDn_FIR_COEF_HV_i[7:0] FIR_HC4
- FIR_HC$_{3}$() = DSS.DISPC_VIDn_FIR_COEF_H_i[31:24] FIR_HC3
- FIR_HC$_{2}$() = DSS.DISPC_VIDn_FIR_COEF_H_i[23:16] FIR_HC2
- FIR_HC$_{1}$() = DSS.DISPC_VIDn_FIR_COEF_H_i[15:8] FIR_HC1
- FIR_HC$_{0}$() = DSS.DISPC_VIDn_FIR_COEF_H_i[7:0] FIR_HC0

13.5.1.3.2 Enabling

The video pipeline #n is enabled/disabled by setting/resetting the DSS.DISPC_VIDn_ATTRS[0].EN bit. While the video pipeline is enabled/disabled, the video layer is visible/not visible on the screen (LCD panel or TV set).

The video up/downsampling block for the video pipeline #n is programmed by setting the DSS.DISPC_VIDn_ATTRS[6:5] RESIZE_EN bit field:

- When the RESIZE_EN[1] bit is set to 1, the video vertical up/downsampling block is enabled. When set to 0, the vertical resize processing is disabled.
- When the RESIZE_EN[0] bit is set to 1, the video horizontal up/downsampling block is enabled. When set to 0, the horizontal resize processing is disabled.
- When the RESIZE_EN[1:0] is set to 0x3, both horizontal and vertical resize processing are enabled.

NOTE:

- Set a valid configuration before enabling the video up/downsampling block.
- Vertical and horizontal downsampling are limited to a 0.25 resize factor. When processing a down-scaling with a vertical factor between 0.5 and 0.25, a 5-tap filter configuration must be used. See Section 13.5.1.3.5 for more information concerning the filter coefficients.

13.5.1.3.3 Factor

The following register bit fields define the increment value of the video up/downsampling block for video pipeline n:

- Vertical up/downsampling increment value (DSS.DISPC_VIDn_FIR[27:16] FIR_V_INC bit field, with n = 1 or 2): The unsigned integer value range is [1:4096]. The software calculates the value using the following equation:

$$\text{VIDFIRVINC[11:0]} = 1024 \text{ x } \frac{\text{VIDORGSIZEY[10:0]}}{\text{VIDSIZEY[10:0]}} \tag{15}$$

dss-E118

NOTE:

- If the FIR_V_INC[11:0] bit field value is greater than 4096, it is clipped to 4096. If VID_SIZE_Y[10:0] equals 0x1, VID_SIZE_Y[10:0] is replaced by 0x2 in the previous equation.
- The VID_ORG_SIZE_Y[10:0] and VID_SIZE_Y[10:0] bit field values must be programmed with the value desired minus 1.

- Horizontal up/downsampling increment value (the DSS.DISPC_VIDn_FIR[11:0] FIR_H_INC bit field, with n = 1 or 2): The unsigned integer value range is [1:4096]. The software calculates the value using the following equation:

$$\text{VIDFIRHINC[11:0]} = 1024 \text{ x } \frac{\text{VIDORGSIZEX[10:0]}}{\text{VIDSIZEX[10:0]}} \tag{16}$$

dss-E119

NOTE:

- If the FIR_H_INC[11:0] bit field value is greater than 4096, it is clipped to 4096. If VID_SIZE_X[10:0] equals 1, VID_SIZE_X[10:0] is replaced by 2 in the previous equation.
- The VID_ORG_SIZE_X[10:0] and VID_SIZE_X[10:0] bit field values must be programmed with the value desired minus 1.

13.5.1.3.4 Initial Phase

- Vertical up/downsampling accumulator value DSS.DISPC_VIDn_ACCUl[25:16] VERTICAL_ACCU bit fields

  The unsigned integer value range is [0:1023]. The accumulator value indicates on which phase the vertical filtering starts. The value 0 indicates that the phase 0 is the first phase used by the hardware to generate the first data.
- Vertical up/downsampling accumulator value DSS.DISPC_VIDn_ACCUl[9:0] VIDHORIZONTALACCU

bit fields

The unsigned integer value range is [0:1023]. The accumulator value indicates on which phase the horizontal filtering starts. The value 0 indicates that the phase 0 is the first phase used by the hardware to generate the first data

Table 13-34 lists the vertical/horizontal accumulator values and phases

Table 13-34. Vertical/Horizontal Accumulator Phase

Accumulator Value	Phases
0	0
128	1
256	2
384	3
512	4
640	5
768	6
896	7

NOTE: For LCD output, the initial phase is always 0 (horizontal and vertical.) For TV output, the vertical phases (odd and even) can be nonzero values.

13.5.1.3.5 Coefficients

- **Vertical up/downsampling coefficients (DSS.DISPC_VIDn_FIR_COEF_HV_i and DSS.DISPC_VIDn_FIR_COEF_V)**

  The 3-tap vertical up/downsampling coefficients are defined in DSS.DISPC_VIDn_FIR_COEF_HV_i registers. There are eight registers for the eight phases with three coefficients for each of them so a total of 24 programmable coefficients for the vertical up/downsampling block. Each register contains two 8-bit signed coefficients and one 8-bit unsigned coefficient (central one).

  In addition, there are 2-tap vertical up/downsampling coefficients defined in DSS.DISPC_VIDn_FIR_COEF_Vi registers. There are eight registers for the eight phases with two coefficients for each of them so a total of 16 programmable coefficients for the vertical up/downsampling block used in addition of the 3-tap registers defined above. Each register contains two 8-bit signed coefficients (C_{22}() and C_{00}()).

  In case of 5-tap configuration, both sets of registers, DSS.DISPC_VIDn_FIR_COEF_HV_i and DSS.DISPC_VIDn_FIR_COEF_V, are used. In case of 3-tap configuration, only one set of registers, DSS.DISPC_VIDn_FIR_COEF_HV, is used.

- **Horizontal up/downsampling coefficients (DSS.DISPC_VIDn_FIR_COEF_H_i and DSS.DISPC_VIDn_FIR_COEF_HV)**

  The 5-tap horizontal up/downsampling coefficients are defined in DSS.DISPC_VIDn_FIR_COEF_H_i and DSS.DISPC_VIDn_FIR_COEF_HV_i registers. There are eight registers for the eight phases with five coefficients for each register, for a total of 40 programmable coefficients for the horizontal up/downsampling block.

  Each DSS.DISPC_VIDn_FIR_COEF_H_i register contains three 8-bit signed coefficients and one 8-bit unsigned coefficient (central one), and each DSS.DISPC_VIDn_FIR_COEF_HV_i contains one 8-bit signed coefficient.

Table 13-35 through Table 13-40 give the programmable coefficients for the FIR up/downsampling filters (Max-Fauque-Berthier method).

13.5.1.3.5.1 Up-Sampling

Table 13-35 gives the 24 coefficients to program the vertical upsampling (3-tap configuration).

Table 13-35. Up-Sampling Vertical Filter Coefficients (Three Taps)

Phases	FIR_VC$_2$()	FIR_VC$_1$()	FIR_VC$_0$()
0	0	128	0
1	3	123	2
2	12	111	5
3	32	89	7
4	0	64	64
5	7	89	32
6	5	111	12
7	2	123	3

Table 13-36 gives the 40 coefficients to program the vertical upsampling (5-tap configuration).

Table 13-36. Up-Sampling Vertical Filter Coefficients (Five Taps)

Phases	FIR_VC$_{22}$()	FIR_VC$_2$()	FIR_VC$_1$()	FIR_VC$_0$()	FIR_VC$_{00}$()
0	0	0	128	0	0
1	-1	13	124	-8	0
2	-2	30	112	-11	-1
3	-5	51	95	-11	-2
4	0	-9	73	73	-9
5	-2	-11	95	51	-5
6	-1	-11	112	30	-2
7	0	-8	124	13	-1

Table 13-37 gives the 40 coefficients to program the horizontal upsampling (5-tap configuration).

Table 13-37. Up-Sampling Horizontal Filter Coefficients (Five Taps)

Phases	FIR_HC$_4$()	FIR_HC$_3$()	FIR_HC$_2$()	FIR_HC$_1$()	FIR_HC$_0$()
0	0	0	128	0	0
1	-1	13	124	-8	0
2	-2	30	112	-11	-1
3	-5	51	95	-11	-2
4	0	-9	73	73	-9
5	-2	-11	95	51	-5
6	-1	-11	112	30	-2
7	0	-8	124	13	-1

The upsampling coefficients register configuration (vertical three taps and horizontal five taps) is the following:

- DSS.DISPC_VIDn_FIR_COEF_H_0 = 0x00800000
- DSS.DISPC_VIDn_FIR_COEF_HV_0 = 0x00800000
- DSS.DISPC_VIDn_FIR_COEF_H_1 = 0x0D7CF800
- DSS.DISPC_VIDn_FIR_COEF_HV_1 = 0x037B02FF
- DSS.DISPC_VIDn_FIR_COEF_H_2 = 0x1E70F5FF
- DSS.DISPC_VIDn_FIR_COEF_HV_2 = 0x0C6F05FE
- DSS.DISPC_VIDn_FIR_COEF_H_3 = 0x335FF5FE
- DSS.DISPC_VIDn_FIR_COEF_HV_3 = 0x205907FB
- DSS.DISPC_VIDn_FIR_COEF_H_4 = 0xF74949F7
- DSS.DISPC_VIDn_FIR_COEF_HV_4 = 0x00404000

- DSS.DISPC_VIDn_FIR_COEF_H_5 = 0xF55F33FB
- DSS.DISPC_VIDn_FIR_COEF_HV_5 = 0x075920FE
- DSS.DISPC_VIDn_FIR_COEF_H_6 = 0xF5701EFE
- DSS.DISPC_VIDn_FIR_COEF_HV_6 = 0x056F0CFF
- DSS.DISPC_VIDn_FIR_COEF_H_7 = 0xF87C0DFF
- DSS.DISPC_VIDn_FIR_COEF_HV_7 = 0x027B0300

NOTE: In this case, the DSS.DISPC_VIDn_FIR_COEF_Vi registers are not used.

The upsampling coefficients register configuration (both vertical and horizontal five taps) is the following:

- DSS.DISPC_VIDn_FIR_COEF_H_0 = 0x00800000
- DSS.DISPC_VIDn_FIR_COEF_HV_0 = 0x00800000
- DSS.DISPC_VIDn_FIR_COEF_V0 = 0x00000000
- DSS.DISPC_VIDn_FIR_COEF_H_1 = 0x0D7CF800
- DSS.DISPC_VIDn_FIR_COEF_HV_1 = 0x0D7CF8FF
- DSS.DISPC_VIDn_FIR_COEF_V1 = 0x0000FF00
- DSS.DISPC_VIDn_FIR_COEF_H_2 = 0x1E70F5FF
- DSS.DISPC_VIDn_FIR_COEF_HV_2 = 0x1E70F5FE
- DSS.DISPC_VIDn_FIR_COEF_V2 = 0x0000FEFF
- DSS.DISPC_VIDn_FIR_COEF_H_3 = 0x335FF5FE
- DSS.DISPC_VIDn_FIR_COEF_HV_3 = 0x335FF5FB
- DSS.DISPC_VIDn_FIR_COEF_V3 = 0x0000FBFE
- DSS.DISPC_VIDn_FIR_COEF_H_4 = 0xF74949F7
- DSS.DISPC_VIDn_FIR_COEF_HV_4 = 0xF7404000
- DSS.DISPC_VIDn_FIR_COEF_V04 = 0x000000F7
- DSS.DISPC_VIDn_FIR_COEF_H_5 = 0xF55F33FB
- DSS.DISPC_VIDn_FIR_COEF_HV_5 = 0xF55F33FE
- DSS.DISPC_VIDn_FIR_COEF_V5 = 0x0000FEFB
- DSS.DISPC_VIDn_FIR_COEF_H_6 = 0xF5701EFE
- DSS.DISPC_VIDn_FIR_COEF_HV_6 = 0xF5701EFF
- DSS.DISPC_VIDn_FIR_COEF_V6 = 0x0000FFFE
- DSS.DISPC_VIDn_FIR_COEF_H_7 = 0xF87C0DFF
- DSS.DISPC_VIDn_FIR_COEF_HV_7 = 0xF87C0D00
- DSS.DISPC_VIDn_FIR_COEF_V7 = 0x000000FF

13.5.1.3.5.2 Down-Sampling

Table 13-38 gives the 24 coefficients to program the vertical downsampling (3-tap configuration).

Table 13-38. Down-Sampling Vertical Filter Coefficients (Three Taps)

Phases	FIR_VC$_2$()	FIR_VC$_1$()	FIR_VC$_0$()
0	36	56	36
1	40	57	31
2	45	56	27
3	50	55	23
4	18	55	55
5	23	55	50
6	27	56	45

Table 13-38. Down-Sampling Vertical Filter Coefficients (Three Taps) (continued)

Phases	$FIR\_VC_2()$	$FIR\_VC_1()$	$FIR\_VC_0()$
7	31	57	40

Table 13-39 gives the 40 coefficients to program the vertical downsampling (5-tap configuration).

Table 13-39. Down-Sampling Vertical Filter Coefficients (Five Taps)

Phases	$FIR\_VC_{22}()$	$FIR\_VC_2()$	$FIR\_VC_1()$	$FIR\_VC_0()$	$FIR\_VC_{00}()$
0	0	36	56	36	0
1	4	40	55	31	-2
2	8	44	54	27	-5
3	-12	48	53	22	-7
4	-9	17	52	51	17
5	-7	22	53	48	12
6	-5	27	54	44	8
7	-2	31	55	40	4

Table 13-40 gives the 40 coefficients to program the horizontal downsampling (5-tap configuration).

Table 13-40. Down-Sampling Horizontal Filter Coefficients (Five Taps)

Phases	$FIR\_HC_4()$	$FIR\_HC_3()$	$FIR\_HC_2()$	$FIR\_HC_1()$	$FIR\_HC_0()$
0	0	36	56	36	0
1	4	40	55	31	-2
2	8	44	54	27	-5
3	-12	48	53	22	-7
4	-9	17	52	51	17
5	-7	22	53	48	12
6	-5	27	54	44	8
7	-2	31	55	40	4

The downsampling coefficients register configuration (vertical three taps and horizontal five taps) is the following:

- DSS.DISPC_VIDn_FIR_COEF_H_0 = 0x24382400
- DSS.DISPC_VIDn_FIR_COEF_HV_0 = 0x24382400
- DSS.DISPC_VIDn_FIR_COEF_H_1 = 0x28371FFE
- DSS.DISPC_VIDn_FIR_COEF_HV_1 = 0x28391F04
- DSS.DISPC_VIDn_FIR_COEF_H_2 = 0x2C361BFB
- DSS.DISPC_VIDn_FIR_COEF_HV_2 = 0x2D381B08
- DSS.DISPC_VIDn_FIR_COEF_H_3 = 0x303516F9
- DSS.DISPC_VIDn_FIR_COEF_HV_3 = 0x3237170C
- DSS.DISPC_VIDn_FIR_COEF_H_4 = 0x11343311
- DSS.DISPC_VIDn_FIR_COEF_HV_4 = 0x123737F7
- DSS.DISPC_VIDn_FIR_COEF_H_5 = 0x1635300C
- DSS.DISPC_VIDn_FIR_COEF_HV_5 = 0x173732F9
- DSS.DISPC_VIDn_FIR_COEF_H_6 = 0x1B362C08
- DSS.DISPC_VIDn_FIR_COEF_HV_6 = 0x1B382DFB
- DSS.DISPC_VIDn_FIR_COEF_H_7 = 0x1F372804
- DSS.DISPC_VIDn_FIR_COEF_HV_7 = 0x1F3928FE

NOTE:

- In this case, the DSS.DISPC_VIDn_FIR_COEF_Vi registers are not used.
- In this case, the downsampling factor must be higher than 1/2.

The downsampling coefficients register configuration (both the vertical and the horizontal five taps) is the following:

- DSS.DISPC_VIDn_FIR_COEF_H_0 = 0x24382400
- DSS.DISPC_VIDn_FIR_COEF_HV_0 = 0x24382400
- DSS.DISPC_VIDn_FIR_COEF_V0 = 0x00000000
- DSS.DISPC_VIDn_FIR_COEF_H_1 = 0x28371FFE
- DSS.DISPC_VIDn_FIR_COEF_HV_1 = 0x28371F04
- DSS.DISPC_VIDn_FIR_COEF_V1 = 0x000004FE
- DSS.DISPC_VIDn_FIR_COEF_H_2 = 0x2C361BFB
- DSS.DISPC_VIDn_FIR_COEF_HV_2 = 0x2C361B08
- DSS.DISPC_VIDn_FIR_COEF_V2 = 0x000008FB
- DSS.DISPC_VIDn_FIR_COEF_H_3 = 0x303516F9
- DSS.DISPC_VIDn_FIR_COEF_HV_3 = 0x3035160C
- DSS.DISPC_VIDn_FIR_COEF_V3 = 0x00000CF9
- DSS.DISPC_VIDn_FIR_COEF_H_4 = 0x11343311
- DSS.DISPC_VIDn_FIR_COEF_HV_4 = 0x113433F7
- DSS.DISPC_VIDn_FIR_COEF_V4 = 0x0000F711
- DSS.DISPC_VIDn_FIR_COEF_H_5 = 0x1635300C
- DSS.DISPC_VIDn_FIR_COEF_HV_5 = 0x163530F9
- DSS.DISPC_VIDn_FIR_COEF_V5 = 0x0000F90C
- DSS.DISPC_VIDn_FIR_COEF_H_6 = 0x1B362C08
- DSS.DISPC_VIDn_FIR_COEF_HV_6 = 0x1B362CFB
- DSS.DISPC_VIDn_FIR_COEF_V6 = 0x0000FB08
- DSS.DISPC_VIDn_FIR_COEF_H_7 = 0x1F372804
- DSS.DISPC_VIDn_FIR_COEF_HV_7 = 0x1F3728FE
- DSS.DISPC_VIDn_FIR_COEF_V7 = 0x0000FE04

NOTE: This configuration must be used for vertical downsampling factors between 1/2 and 1/4

13.5.2 Display Low-Power Refresh Settings

This section describes the display low-power refresh application on the device. The display subsystem remains active while saving power by putting unused power domains and unused modules into idle mode. This process can be expanded to include the screen saver mode in which the MPU subsystem wakes up to update the frame buffer and then returns to idle mode. On the device platform, where power consumption is of high importance, the display modes must be configured properly to achieve optimal power savings

The display low-power refresh mode can be used in the following scenarios:

- During the period of time when there is no application running and the backlight turns off.
- Once the backlight turns off, the LCD display can be shut off or can be refreshed showing the time and date. The screen saver mode can be used to update the time every minute.

This section discusses the methodology for finding optimal power savings. These settings are detailed for a 16-bit, 240 x 320 pixel QVGA LCD.

13.5.2.1 Display Low-Power Refresh Overview

When the device is not in idle mode, meaning all clocks are on and the power is applied to all power domains, the following activity typically occurs with respect to the display subsystem:

- The MPU subsystem is processing
- The display subsystem DMA controller is moving data from the SDRAM frame buffer location to the display subsystem internal FIFO.
- The LCD data is being sent from the internal FIFO to the display panel.

When the MPU goes into idle mode, the following activity occurs:

- The display subsystem DMA controller remains active, moving data from the SDRAM frame buffer to the internal FIFO.
- The SDRAM will go in and out of self-refresh between transfers.
- The display subsystem internal FIFO will continue to send LCD data to the display panel.

This procedure is named as the display low-power refresh scenario.

NOTE: In the device, the display subsystem has its own power domain (the DSS power domain).

13.5.2.2 Display Subsystem Clock

13.5.2.2.1 Display Subsystem Clock Configuration

The display subsystem contains one functional clock source, which is the DSS functional clock 1 (DSS1_ALWON_FCLK):

- DSS1_ALWON_FCLK is sourced from DPLL4 (DPLL4_ALWON_FCLK), with several multipliers available and is configured in the PRCM.CM_CLKSEL_DSS[4:0] CLKSEL_DSS1 bit field.

The pixel clock is set as DSS1_ALWON_FCLK by configuring the DSS.DSS_CONTROL[0] DISPC_CLK_SWITCH bit.

The LCD logic clock is determined by the DSS.DISPC_DIVISOR[23:16] LCD bit field. This divisor is used on the DSS functional clock that is selected in the DSS_CONTROL register (either DSS1_ALWON_FCLK or DSS2_ALWON_FCLK). This LCD divisor selects the logical clock frequency which is used to clock the logic in the display subsystem. For some applications there is a required minimum logical clock frequency. The lower the logical clock frequency then the lower the power consumption.

The pixel clock is determined by setting the DSS.DISPC_DIVISOR[7:0] PCD bit field. This divisor is used on the LCD logic clock.

In the following example, the DPLL4 clock (DPLL4_ALWON_FCLK) is enabled and running at 266 MHz:

The PRCM.CM_CLKSEL2_PLL[19:8] PERIPH_DPLL_MULT bit field is set to 0x4 and the PRCM.CM_CLKSEL2_PLL[6:0] PERIPH_DPLL_DIV bit field is set to 0x1 (DPLL4 x 4/(1+1)):

DPLL4_ALWON_FCLKOUT= DPLL4_ALWON_FCLK(266MHz) x 4/2 = 532 MHz

NOTE: The DPLL4_ALWON_FCLOUT clock is an internal clock in DPLL4 module after the DPLL_MULT and DPLL_DIV stages. The DPLL4_M4_CLK clock is one of the DPLL4 output clocks and is the clock source for DSS1_ALWON_FCLK.

The DSS.DSS_CONTROL[0] DSS_CLK_SWITCH bit is set to 0x0 to select DSS1_ALWON_FCLK as the display subsystem functional clock:

DSS1_ALWON_FCLK = DPLL4_M4_CLK

The PRCM.CM_CLKSEL_DSS[4:0] CLKSEL_DSS1 bit field is set to 0x08:

$$\text{DSS1\_ALWON\_FCLK} = \frac{\text{DPLL4\_ALWON\_FCLKOUT (532MHz)}}{8} = 66.5\ \text{MHz} \tag{17}$$

dss-E120

The DSS.DISPC_DIVISOR[23:16] LCD bit field is set to 0x01:

$$\text{LogicClock} = \frac{\text{DSS1\_ALWON\_FCLK(66.5MHz)}}{1} = 66.5\ \text{Mhz} \tag{18}$$

dss-E121

The DSS.DISPC_DIVISOR[6:0] PCD bit field is set to 0x0C:

$$\text{PixelClock} = \frac{\text{LogicClock(66.5MHz)}}{12} = 5.54\ \text{Mhz} \tag{19}$$

dss-E122

13.5.2.2.1.1 Pixel Clock Frequency Settings to Reduce Power Consumption

Power consumption is reduced when a low pixel clock frequency is used. If the clock frequency is set too low, however, the frames-per-second (FPS) are reduced. This can result in visible flickering on the screen each time the screen is refreshed. To avoid electrical polarization problems, refer to the appropriate LCD panel datasheet to determine the maximum range of pixel clock frequency variation. To save power, therefore, the pixel clock frequency during low-power mode must be set as low as possible, but high enough to eliminate visible flickering

13.5.2.2.1.2 Display Subsystem Divider Settings to Reduce Power Consumption

The pixel clock is determined by the DSS.DISPC_DIVISOR[7:0] PCD and DSS.DISPC_DIVISOR[23:16] LCD settings. In most cases, the LCD[7:0] bit field is set to 0x1 and the PCD[7:0] bit field is used as the main divider. To reduce power consumption, software users should investigate if the DSS.DISPC_DIVISOR[23:16] LCD bit field can be set to a value other than 0x1 and then decrease DSS.DISPC_DIVISOR[7:0] PCD bit field value. For example, if the desired pixel clock is 1.625 MHz with a 13-MHz functional clock, then this pixel clock can be achieved by setting DSS.DISPC_DIVISOR[23:16] LCD to 0x1 and DSS.DISPC_DIVISOR[7:0] PCD to 0x8. The same pixel clock can be achieved by setting DSS.DISPC_DIVISOR[23:16] LCD to 0x2 and DSS.DISPC_DIVISOR[7:0] PCD to 0x4.

13.5.2.2.2 Display Subsystem Clock Enable

To take the DSS out of reset, all DSS-related clocks must be enabled, and the DPLL4 clock must be enabled. After taking the DSS out of reset, these clocks can be disabled if they are not used. The following clocks must be enabled before the DSS can come out of reset:

- PRCM.CM_FCLKEN_DSS[0] EN_DSS1 = 0x1
- PRCM.CM_FCLKEN_DSS[1] EN_DSS2 = 0x1
- PRCM.CM_FCLKEN_DSS[2] EN_TV = 0x1
- PRCM.CM_ICLKEN_DSS[0] EN_DSS = 0x1
- PRCM.CM_CLKEN_PLL[18:16] EN_PERIPH_DPLL = 0x7

Once these clocks are enabled, the display subsystem can be taken out of reset.

The following sections explain the display low-power mode configuration options, which are determined by product requirements (LCD panel type).

13.5.2.3 Autoidle and Smart Idle

13.5.2.3.1 Autoidle

To further save power consumption, the autoidle feature at the module can be enabled for the active modules. For example, the PRCM and the system control modules are active during this mode. By enabling the autoidle feature, the clocks at the module level are gated when they are not needed.

The RFBI, display controller, and L4 interfaces can internally gate their clocks to decrease power consumption if no transaction is present on the related bus. The following bits must be set to enable this functionality:

- DSS.DSS_SYSCONFIG[0] AUTOIDLE bit (1: Autoidle; 0: Clock free-running) for the display subsystem
- DSS.RFBI_SYSCONFIG[0] AUTOIDLE bit (1: Autoidle; 0: Clock free-running) for the RFBI

- DSS.DISPC_SYSCFG[0] AUTOIDLE bit (1: Autoidle; 0: Clock free-running) for the display controller
- DSS.DISPC_CFG[9] FUNC_GATED bit (1: Functional clocks gated enabled, 0: Functional clocks gated disabled) for the display controller

13.5.2.3.2 Smart-Idle

The smart-idle feature can be enabled to allow the module to enter idle when the clocks are not needed. The smart-idle feature can be enabled for the display subsystem submodules to further save power consumption:

- Display controller: DSS.DISPC_SYSCFG[4:3] SIDLEMODE
- RFBI: DSS.RFBI_SYSCONFIG[4:3] SIDLEMODE

13.5.2.4 FIFO Thresholds

The display subsystem internal FIFO is used to move data to the LCD panel. This FIFO is filled by the display subsystem DMA controller. The DMA controller is triggered to start and stop based on two thresholds:

- DSS.DISPC_GFX_FIFO_THR[11:0] GFX_FIFO_LOW_THR
- DSS.DISPC_GFX_FIFO_THR[27:16] GFX_FIFO_HIGH_THR

When the level of the FIFO reaches the low threshold, the internal DMA controller begins to fill the FIFO With the data in the frame buffer. Once the amount of pixel data reaches the high threshold, the internal DMA controller stops.

13.5.2.4.1 FIFO Threshold Settings to Reduce Power Consumption

Power consumption is reduced by increasing the difference between the high and low FIFO threshold levels, thereby leaving the SDRAM in self-refresh for a longer period of time. To perform this reduction, consider the following:

- The low FIFO threshold level must be as low as possible, but not low enough to cause any underflow.
- The high FIFO threshold level must not exceed the FIFO size minus one burst. A value above this limit results in the DMA controller trying to fill the FIFO to a level that cannot be reached, which will increase power consumption.
- The difference between high and low FIFO threshold levels must not be less than one burst size. These settings do not reduce power consumption because the SDRAM never goes into self-refresh, but they will avoid underflow.

13.5.2.5 Vertical and Horizontal Timings

The vertical and horizontal timings and the pixel clock speed determine the number of frames updated per second. Figure 13-76 shows the timings for a 240 x 320 pixel QVGA LCD panel. If the pulse width (also called blanking) and the front porch parameters are increased, more setup time is added before the data is transferred. This additional time is beneficial for delaying the data transfer if the data is not ready because of bandwidth limitations. Care must be taken to determine the fps when modifying these parameters.

Use the following formula to determine the fps for a 240 x 320 QVGA LCD:

$$fps = \frac{1}{[(Hsw + 1) + (Hfp + 1) + 240 + (Hbp + 1)] \times [(Vsw + 1) + Vfp + 320 + Vbp)] \times (PCLK)} \qquad (20)$$

dss-E123

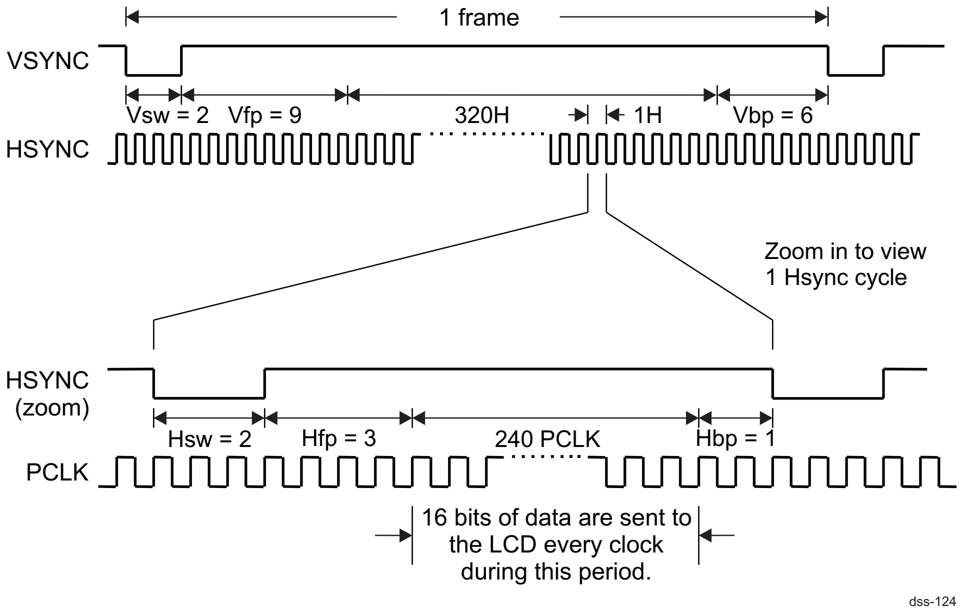
With:

- Hsw: DSS.DISPC_TIMING_H[7:0] HSW bit field value
- Hfp: DSS.DISPC_TIMING_H[19:8] HFP bit field value
- Hbp: DSS.DISPC_TIMING_H[31:20] HBP bit field value
- Vsw: DSS.DISPC_TIMING_V[7:0] VSW bit field value
- Vfp: DSS.DISPC_TIMING_V[19:8] VFP bit field value

- Vbp: DSS.DISPC_TIMING_V[31:20] VBP bit field value
- PCLK: Pixel clock period

The horizontal (Hsw) and vertical (Vsw) pulse widths and the horizontal front (Hfp) and back (Hbp) porches are increased by 1 because the value is programmed as the desired value minus 1.

Figure 13-76. QVGA LCD Timings



The fps for the example of 6-MHz pixel clock with the setting shown in Figure 13-76 is as follows:

$$fps = \frac{1}{[(2 + 1) + 4 + 240 + 2] \times [(2 + 1) + 9 + 320 + 6] \times 166.67 \times 10^{-9}}$$

dss-E125

$$fps = 71.57Hz \qquad (21)$$

13.5.2.5.1 Horizontal and Vertical Timing Settings to Reduce Power Consumption

The number of fps that the screen is refreshed is also determined by the vertical and horizontal timings. Consequently, longer timings between frames (blanking periods) reduce the fps and reduce average power consumption. Shorter blanking periods increase fps and increases power consumption. If the blanking between frames is too small, a FIFO underflow may occur.

13.6 Registers

13.6.1 DSS_DISPC Registers

Table 13-41 lists the memory-mapped registers for the DSS_DISPC. All register offset addresses not listed in Table 13-41 should be considered as reserved locations and the register contents should not be modified.

Table 13-41. DSS_DISPC Registers

Offset	Acronym	Register Name	Section
0h	DISPC_REVISION		Section 13.6.1.1
10h	DISPC_SYSCFG		Section 13.6.1.2
14h	DISPC_SYSSTS		Section 13.6.1.3
18h	DISPC_IRQSTS		Section 13.6.1.4
1Ch	DISPC_IRQEN		Section 13.6.1.5

Table 13-41. DSS_DISPC Registers (continued)

Offset	Acronym	Register Name	Section
40h	DISPC_CTRL		Section 13.6.1.6
44h	DISPC_CFG		Section 13.6.1.7
4Ch	DISPC_DEFAULT_COLOR_0		Section 13.6.1.8
50h	DISPC_DEFAULT_COLOR_1		Section 13.6.1.8
54h	DISPC_TRANS_COLOR_0		Section 13.6.1.9
58h	DISPC_TRANS_COLOR_1		Section 13.6.1.9
5Ch	DISPC_LINE_STS		Section 13.6.1.10
60h	DISPC_LINE_NUMBER		Section 13.6.1.11
64h	DISPC_TIMING_H		Section 13.6.1.12
68h	DISPC_TIMING_V		Section 13.6.1.13
6Ch	DISPC_POL_FREQ		Section 13.6.1.14
70h	DISPC_DIVISOR		Section 13.6.1.15
74h	DISPC_GLOBAL_ALPHA		Section 13.6.1.16
78h	DISPC_SIZE_DIG		Section 13.6.1.17
7Ch	DISPC_SIZE_LCD		Section 13.6.1.18
80h	DISPC_GFX_BA_0		Section 13.6.1.19
84h	DISPC_GFX_BA_1		Section 13.6.1.19
88h	DISPC_GFX_POSITION		Section 13.6.1.20
8Ch	DISPC_GFX_SIZE		Section 13.6.1.21
A0h	DISPC_GFX_ATTRS		Section 13.6.1.22
A4h	DISPC_GFX_FIFO_THR		Section 13.6.1.23
A8h	DISPC_GFX_FIFO_SIZE_STS		Section 13.6.1.24
ACh	DISPC_GFX_ROW_INC		Section 13.6.1.25
B0h	DISPC_GFX_PIXEL_INC		Section 13.6.1.26
B4h	DISPC_GFX_WINDOW_SKIP		Section 13.6.1.27
B8h	DISPC_GFX_TBL_BA		Section 13.6.1.28
BCh to C0h	DISPC_VID1_BA_0 to DISPC_VID1_BA_1		Section 13.6.1.29
C4h	DISPC_VID1_POSITION		Section 13.6.1.30
C8h	DISPC_VID1_SIZE		Section 13.6.1.31
CCh	DISPC_VID1_ATTRS		Section 13.6.1.32
D0h	DISPC_VID1_FIFO_THR		Section 13.6.1.33
D4h	DISPC_VID1_FIFO_SIZE_STS		Section 13.6.1.34
D8h	DISPC_VID1_ROW_INC		Section 13.6.1.35
DCh	DISPC_VID1_PIXEL_INC		Section 13.6.1.36
E0h	DISPC_VID1_FIR		Section 13.6.1.37
E4h	DISPC_VID1_PICTURE_SIZE		Section 13.6.1.38
E8h to ECh	DISPC_VID1_ACCU_0 to DISPC_VID1_ACCU_1		Section 13.6.1.39
F0h	DISPC_VID1_FIR_COEF_H_0		Section 13.6.1.40
F4h	DISPC_VID1_FIR_COEF_HV_0		Section 13.6.1.41
F8h	DISPC_VID1_FIR_COEF_H_1		Section 13.6.1.40
FCh	DISPC_VID1_FIR_COEF_HV_1		Section 13.6.1.41
100h	DISPC_VID1_FIR_COEF_H_2		Section 13.6.1.40
104h	DISPC_VID1_FIR_COEF_HV_2		Section 13.6.1.41
108h	DISPC_VID1_FIR_COEF_H_3		Section 13.6.1.40
10Ch	DISPC_VID1_FIR_COEF_HV_3		Section 13.6.1.41
110h	DISPC_VID1_FIR_COEF_H_4		Section 13.6.1.40

Table 13-41. DSS_DISPC Registers (continued)

Offset	Acronym	Register Name	Section
114h	DISPC_VID1_FIR_COEF_HV_4		Section 13.6.1.41
118h	DISPC_VID1_FIR_COEF_H_5		Section 13.6.1.40
11Ch	DISPC_VID1_FIR_COEF_HV_5		Section 13.6.1.41
120h	DISPC_VID1_FIR_COEF_H_6		Section 13.6.1.40
124h	DISPC_VID1_FIR_COEF_HV_6		Section 13.6.1.41
128h	DISPC_VID1_FIR_COEF_H_7		Section 13.6.1.40
12Ch	DISPC_VID1_FIR_COEF_HV_7		Section 13.6.1.41
130h	DISPC_VID1_CONV_COEF0		Section 13.6.1.42
134h	DISPC_VID1_CONV_COEF1		Section 13.6.1.43
138h	DISPC_VID1_CONV_COEF2		Section 13.6.1.44
13Ch	DISPC_VID1_CONV_COEF3		Section 13.6.1.45
140h	DISPC_VID1_CONV_COEF4		Section 13.6.1.46
14Ch to 150h	DISPC_VID2_BA_0 to DISPC_VID2_BA_1		Section 13.6.1.47
154h	DISPC_VID2_POSITION		Section 13.6.1.48
158h	DISPC_VID2_SIZE		Section 13.6.1.49
15Ch	DISPC_VID2_ATTRS		Section 13.6.1.50
160h	DISPC_VID2_FIFO_THR		Section 13.6.1.51
164h	DISPC_VID2_FIFO_SIZE_STS		Section 13.6.1.52
168h	DISPC_VID2_ROW_INC		Section 13.6.1.53
16Ch	DISPC_VID2_PIXEL_INC		Section 13.6.1.54
170h	DISPC_VID2_FIR		Section 13.6.1.55
174h	DISPC_VID2_PICTURE_SIZE		Section 13.6.1.56
178h to 17Ch	DISPC_VID2_ACCU_0 to DISPC_VID2_ACCU_1		Section 13.6.1.57
180h	DISPC_VID2_FIR_COEF_H_0		Section 13.6.1.58
184h	DISPC_VID2_FIR_COEF_HV_0		Section 13.6.1.59
188h	DISPC_VID2_FIR_COEF_H_1		Section 13.6.1.58
18Ch	DISPC_VID2_FIR_COEF_HV_1		Section 13.6.1.59
190h	DISPC_VID2_FIR_COEF_H_2		Section 13.6.1.58
194h	DISPC_VID2_FIR_COEF_HV_2		Section 13.6.1.59
198h	DISPC_VID2_FIR_COEF_H_3		Section 13.6.1.58
19Ch	DISPC_VID2_FIR_COEF_HV_3		Section 13.6.1.59
1A0h	DISPC_VID2_FIR_COEF_H_4		Section 13.6.1.58
1A4h	DISPC_VID2_FIR_COEF_HV_4		Section 13.6.1.59
1A8h	DISPC_VID2_FIR_COEF_H_5		Section 13.6.1.58
1ACh	DISPC_VID2_FIR_COEF_HV_5		Section 13.6.1.59
1B0h	DISPC_VID2_FIR_COEF_H_6		Section 13.6.1.58
1B4h	DISPC_VID2_FIR_COEF_HV_6		Section 13.6.1.59
1B8h	DISPC_VID2_FIR_COEF_H_7		Section 13.6.1.58
1BCh	DISPC_VID2_FIR_COEF_HV_7		Section 13.6.1.59
1C0h	DISPC_VID2_CONV_COEF0		Section 13.6.1.60
1C4h	DISPC_VID2_CONV_COEF1		Section 13.6.1.61
1C8h	DISPC_VID2_CONV_COEF2		Section 13.6.1.62
1CCh	DISPC_VID2_CONV_COEF3		Section 13.6.1.63
1D0h	DISPC_VID2_CONV_COEF4		Section 13.6.1.64
1D4h	DISPC_DATA_CYCLE_0		Section 13.6.1.65
1D8h	DISPC_DATA_CYCLE_1		Section 13.6.1.65

Table 13-41. DSS_DISPC Registers (continued)

Offset	Acronym	Register Name	Section
1DCh	DISPC_DATA_CYCLE_2		Section 13.6.1.65
1E0h to 1FCh	DISPC_VID1_FIR_COEF_V_0 to DISPC_VID1_FIR_COEF_V_7		Section 13.6.1.66
200h to 21Ch	DISPC_VID2_FIR_COEF_V_0 to DISPC_VID2_FIR_COEF_V_7		Section 13.6.1.67
220h	DISPC_CPR_COEF_R		Section 13.6.1.68
224h	DISPC_CPR_COEF_G		Section 13.6.1.69
228h	DISPC_CPR_COEF_B		Section 13.6.1.70
22Ch	DISPC_GFX_PRELOAD		Section 13.6.1.71
230h	DISPC_VID1_PRELOAD		Section 13.6.1.72
234h	DISPC_VID2_PRELOAD		Section 13.6.1.73

13.6.1.1 DISPC_REVISION Register (offset = 0h) [reset = 0h]

DISPC_REVISION is shown in Figure 13-77 and described in Table 13-42.

This register contains the IP revision code.

Figure 13-77. DISPC_REVISION Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								REV							
R-0h																								R-0h							

Table 13-42. DISPC_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	REV	R	0h	IP revision [7:4] Major revision [3:0] Minor revision

13.6.1.2 DISPC_SYSCFG Register (offset = 10h) [reset = 1h]

DISPC_SYSCFG is shown in Figure 13-78 and described in Table 13-43.

This register allows the control of various parameters of the interconnect interface

Figure 13-78. DISPC_SYSCFG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="2">MIDLEMODE</td><td colspan="2">RESERVED</td><td colspan="2">CLOCK_ACTIVITY</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">RESERVED</td><td colspan="2">SIDLEMODE</td><td>ENWAKEUP</td><td>SOFTRESET</td><td>AUTOIDLE</td></tr>
<tr><td colspan="3">R/W-0h</td><td colspan="2">R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-1h</td></tr>
</table>

Table 13-43. DISPC_SYSCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R/W	0h	
13-12	MIDLEMODE	R/W	0h	Master interface power management, standby/waitcontrol 0h (R/W) = MStandby is asserted only when the module is disabled 1h (R/W) = MStandby is never asserted 2h (R/W) = MStandby is asserted based on the internal activity of the module 3h (R/W) = 3
11-10	RESERVED	R/W	0h	
9-8	CLOCK_ACTIVITY	R/W	0h	Clock activity during wakeup mode period 0h (R/W) = interface and functional clocks can be switched off. 1h (R/W) = Functional clocks can be switched off and interface clocks are maintained during wakeup period 2h (R/W) = Interface clocks can be switched off and functional clocks are maintained during wakeup period 3h (R/W) = Interface and functional clocks are maintained during wakeup period
7-5	RESERVED	R/W	0h	
4-3	SIDLEMODE	R/W	0h	Slave interface power management, idle req/ack control 0h (R/W) = An idle request is acknowledged unconditionally 1h (R/W) = An idle request is never acknowledged 2h (R/W) = Idle request is acknowledged based on the internal activity of the module 3h (R/W) = Reserved
2	ENWAKEUP	R/W	0h	Wakeup feature control 0h (R/W) = Wakeup is disabled 1h (R/W) = Wakeup is enabled

Table 13-43. DISPC_SYSCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	SOFTRESET	R/W	0h	Software reset. Set this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads, it always returns 0h (R/W) = Normal mode 1h (R/W) = The module is reset.
0	AUTOIDLE	R/W	1h	Internal interface clock gating strategy 0h (R/W) = Interface clock is free-running 1h (R/W) = Automatic L3 and L4 interface clock gating strategy is applied based on interface activity

13.6.1.3 DISPC_SYSSTS Register (offset = 14h) [reset = 1h]

DISPC_SYSSTS is shown in Figure 13-79 and described in Table 13-44.

This register provides status information about the module, excluding interrupt status information.

Figure 13-79. DISPC_SYSSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							RESETDONE
R-0h							R-1h

Table 13-44. DISPC_SYSSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-1	RESERVED	R	0h	
0	RESETDONE	R	1h	Internal reset monitoring 0h (R) = Internal module reset is ongoing 1h (R) = Reset complete

13.6.1.4 DISPC_IRQSTS Register (offset = 18h) [reset = 0h]

DISPC_IRQSTS is shown in Figure 13-80 and described in Table 13-45.

This register regroups all the status of module internal events that generate an interrupt. A write of 1 to a given bit
resets the bit

Figure 13-80. DISPC_IRQSTS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="7">RESERVED</td><td>WAKEUP</td></tr>
<tr><td colspan="7">R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>SYNC_LOST_DIGITAL</td><td>SYNC_LOST</td><td>VID2_END_WINDOW</td><td>VID2_FIFO_UFLOW</td><td>VID1_END_WINDOW</td><td>VID1_FIFO_UFLOW</td><td>OCP_ERROR</td><td>PALLETE_GAMMA_LOADING</td></tr>
<tr><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>GFXEND_WINDOW</td><td>GFXFIFO_UFLOW</td><td>PGM_LINE_NO</td><td>ACBIAS_CNT_STS</td><td>EVSYNC_ODD</td><td>EVSYNC_EVEN</td><td>VSYNC</td><td>FRMDONE</td></tr>
<tr><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 13-45. DISPC_IRQSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	
16	WAKEUP	R/W	0h	Wakeup 0h (W) = Wakeup status bit unchanged 0h (R) = Wakeup is false 1h (W) = Wakeup status bit reset 1h (R) = Wakeup is true (pending)
15	SYNC_LOST_DIGITAL	R/W	0h	SyncLostDigital 0h (W) = SyncLostDigital status bit unchanged 0h (R) = SyncLostDigital is false 1h (W) = SyncLostDigital status bit reset 1h (R) = SyncLostDigital is true (pending)
14	SYNC_LOST	R/W	0h	SyncLost 0h (W) = SyncLost status bit unchanged 0h (R) = SyncLost is false 1h (W) = SyncLost status bit reset 1h (R) = SyncLost is true (pending)
13	VID2_END_WINDOW	R/W	0h	Vid2EndWindow 0h (W) = Vid2EndWindow status bit unchanged 0h (R) = Vid2EndWindow is false 1h (W) = Vid2EndWindow status bit reset 1h (R) = Vid2EndWindow is true (pending)
12	VID2_FIFO_UFLOW	R/W	0h	Vid2FIFOUnderflow 0h (W) = Vid2FIFOUnderflow status bit unchanged 0h (R) = Vid2FIFOUnderflow 1h (W) = Vid2FIFOUnderflow status bit reset 1h (R) = Vid2FIFOUnderflow is true (pending)

Table 13-45. DISPC_IRQSTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	VID1_END_WINDOW	R/W	0h	Vid1EndWindow 0h (W) = Vid1EndWindow status bit unchanged 0h (R) = Vid1EndWindow is false 1h (W) = Vid1EndWindow status bit reset 1h (R) = Vid1EndWindow is true (pending).
10	VID1_FIFO_UFLOW	R/W	0h	Vid1FIFOUnderflow 0h (W) = Vid1FIFOUnderflow status bit unchanged 0h (R) = Vid1FIFOUnderflow is false 1h (W) = Vid1FIFOUnderflow status bit reset 1h (R) = Vid1FIFOUnderflow is true (pending).
9	OCP_ERROR	R/W	0h	OCPError 0h (W) = OCPError status bit unchanged 0h (R) = OCPError is false 1h (W) = OCPError status bit reset 1h (R) = OCPError is true (pending).
8	PALLETE_GAMMA_LOADING	R/W	0h	PaletteGammaLoading 0h (W) = PaletteGammaLoading status bit unchanged 0h (R) = PaletteGammaLoading is False 1h (W) = PaletteGammaLoading status bit reset 1h (R) = PaletteGammaLoading is true (pending)
7	GFXEND_WINDOW	R/W	0h	GfxEndWindow 0h (W) = GfxEndWindow status bit unchanged 0h (R) = GfxEndWindow is false. 1h (W) = GfxEndWindow status bit reset 1h (R) = GfxEndWindow is true (pending).
6	GFXFIFO_UFLOW	R/W	0h	GfxFIFOUnderflow 0h (W) = GfxFIFOUnderflow status bit unchanged 0h (R) = GfxFIFOUnderflow is false 1h (W) = GfxFIFOUnderflow status bit reset 1h (R) = GfxFIFOUnderflow is true (pending)
5	PGM_LINE_NO	R/W	0h	ProgrammedLineNumber 0h (W) = ProgrammedLineNumber status bit unchanged 0h (R) = ProgrammedLineNumber is false 1h (W) = ProgrammedLineNumber status bit reset 1h (R) = ProgrammedLineNumber is true (pending)
4	ACBIAS_CNT_STS	R/W	0h	ACBiasCountStatus 0h (W) = ACBiasCountStatus status bit unchanged 0h (R) = ACBiasCountStatus is false. 1h (W) = ACBiasCountStatus status bit reset 1h (R) = ACBiasCountStatus is true (pending)
3	EVSYNC_ODD	R/W	0h	EVSYNC_ODD 0h (W) = EVSYNC_ODD status bit unchanged 0h (R) = EVSYNC_ODD is false 1h (W) = EVSYNC_ODD status bit reset 1h (R) = EVSYNC_ODD is true (pending)
2	EVSYNC_EVEN	R/W	0h	EVSYNC_EVEN 0h (W) = EVSYNC_EVEN status bit unchanged 0h (R) = EVSYNC_EVEN is false. 1h (W) = EVSYNC_EVEN status bit reset 1h (R) = EVSYNC_EVEN is true (pending)

Table 13-45. DISPC_IRQSTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	VSYNC	R/W	0h	VSYNC 0h (W) = VSYNC status bit unchanged 0h (R) = VSYNC is false 1h (W) = VSYNC status bit reset 1h (R) = VSYNC is true (pending)
0	FRMDONE	R/W	0h	FrameDone 0h (W) = FrameDone status bit unchanged 0h (R) = FrameDone is false 1h (W) = FrameDone status bit reset 1h (R) = FrameDone is true (pending).

13.6.1.5 DISPC_IRQEN Register (offset = 1Ch) [reset = 0h]

DISPC_IRQEN is shown in Figure 13-81 and described in Table 13-46.

This register allows the masking/unmasking of module internal interrupt sources, on an event-by-event basis.

Figure 13-81. DISPC_IRQEN Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							WAKEUP
R/W-0h							R/W-0h
15	14	13	12	11	10	9	8
SYNC_LOST_DIGITAL	SYNC_LOST	VID2_END_WINDOW	VID2_FIFO_UFLOW	VID1_END_WINDOW	VID1_FIFO_UFLOW	OCP_ERROR	PALLETE_GAMMA_LOADING
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GFXEND_WINDOW	GFXFIFO_UFLOW	PGM_LINE_NO	ACBIAS_COUNT_STS	EVSYNC_ODD	EVSYNC_EVEN	VSYNC	FRM_DONE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 13-46. DISPC_IRQEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	
16	WAKEUP	R/W	0h	Wakeup mask 0h (R/W) = Wakeup is masked 1h (R) = Wakeup is masked
15	SYNC_LOST_DIGITAL	R/W	0h	SyncLostDigital 0h (R/W) = SyncLostDigital is masked 1h (R) = SyncLostDigital generates an interrupt when it occurs
14	SYNC_LOST	R/W	0h	SyncLost 0h (R/W) = SyncLost is masked 1h (R) = SyncLost is masked
13	VID2_END_WINDOW	R/W	0h	Vid2EndWindow 0h (R/W) = Vid2EndWindow is masked 1h (R) = Vid2EndWindow generates an interrupt when it occurs
12	VID2_FIFO_UFLOW	R/W	0h	Vid2FIFOUnderflow 0h (R/W) = Vid2FIFOUnderflow is masked 1h (R) = Vid2FIFOUnderflow generates an interrupt when it occurs
11	VID1_END_WINDOW	R/W	0h	EndVid1Window 0h (R/W) = EndVid1Window is masked 1h (R) = EndVid1Window is masked
10	VID1_FIFO_UFLOW	R/W	0h	Vid1FIFOUnderflow 0h (R/W) = Vid1FIFOUnderflow is masked 1h (R) = Vid1FIFOUnderflow generates an interrupt when it occurs
9	OCP_ERROR	R/W	0h	OCPError 0h (R/W) = OCPError 1h (R) = OCPError generates an interrupt when it occurs

Table 13-46. DISPC_IRQEN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	PALLETE_GAMMA_LOADING	R/W	0h	PaletteGammaMask 0h (R/W) = PaletteGammaMask is masked 1h (R) = PaletteGammaMask generates an interrupt when it occurs
7	GFXEND_WINDOW	R/W	0h	GfxEndWindow 0h (R/W) = GfxEndWindow is masked 1h (R) = GfxEndWindow generates an interrupt when it occurs
6	GFXFIFO_UFLOW	R/W	0h	GfxFIFOUnderflow 0h (R/W) = GfxFIFOUnderflow is masked 1h (R) = GfxFIFOUnderflow generates an interrupt when it occurs
5	PGM_LINE_NO	R/W	0h	ProgrammedLineNumber 0h (R/W) = ProgrammedLineNumber is masked. 1h (R) = ProgrammedLineNumber generates an interrupt when it occurs
4	ACBIAS_COUNT_STS	R/W	0h	ACBiasCountStatus 0h (R/W) = ACBiasCountStatus is masked 1h (R) = ACBiasCountStatus generates an interrupt when it occurs.
3	EVSYNC_ODD	R/W	0h	EVSYNC_ODD 0h (R/W) = EVSYNC_ODD is masked 1h (R) = EVSYNC_ODD generates an interrupt when it occurs
2	EVSYNC_EVEN	R/W	0h	EVSYNC_EVEN 0h (R/W) = EVSYNC_EVEN is masked 1h (R) = EVSYNC_EVEN generates an interrupt when it occurs
1	VSYNC	R/W	0h	VSYNC 0h (R/W) = VSYNC is masked 1h (R) = VSYNC generates an interrupt when it occurs.
0	FRM_DONE	R/W	0h	FrameMask 0h (R/W) = FrameMask is masked 1h (R) = FrameMask generates an interrupt when it occurs

13.6.1.6 DISPC_CTRL Register (offset = 40h) [reset = 0h]

DISPC_CTRL is shown in Figure 13-82 and described in Table 13-47.

The control register configures the display controller module

Figure 13-82. DISPC_CTRL Register

31	30	29	28	27	26	25	24
SPATIAL_TEMPORAL_DITHER		LCD_EN_POL	LCD_EN_SIGNAL	PCK_FREE_EN	TDM_UNUSED_BITS		TDM_CYCLE_FMT
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h

23	22	21	20	19	18	17	16
TDM_CYCLE_FMT	TDM_PARALLEL_MODE		TDM_EN	HT			GPOUT1
R/W-0h	R/W-0h		R/W-0h	R/W-0h			R/W-0h

15	14	13	12	11	10	9	8
GPOUT0	GPIN1	GPIN0	OVLY_OPT	STALL_MODE	RESERVED	TFT_DATA_LINES	
R/W-0h	R-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

7	6	5	4	3	2	1	0
ST_DITHER_EN	GO_DIGITAL	GO_LCD	M8B	STNTFT	MONO_COLOR	DIGITAL_EN	LCD_EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 13-47. DISPC_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SPATIAL_TEMPORAL_DITHER	R/W	0h	Spatial/Temporal dithering number of frames 0h (R/W) = Spatial only 1h (R/W) = Spatial and temporal over two frames 2h (R/W) = Spatial and temporal over four frames 3h (R/W) = Reserved
29	LCD_EN_POL	R/W	0h	LCD Enable Signal Polarity 0h (R/W) = Active low 1h (R/W) = Active high
28	LCD_EN_SIGNAL	R/W	0h	LCD Enable Signal: LCD interface active/inactive 0h (R/W) = Signal disabled 1h (R/W) = Signal enabled
27	PCK_FREE_EN	R/W	0h	Pixel clock free-running enabled/disabled 0h (R/W) = Clock disabled 1h (R/W) = Clock enabled
26-25	TDM_UNUSED_BITS	R/W	0h	State of unused bits (TDM mode only) 0h (R/W) = Low level (0) 1h (R/W) = High level (1) 2h (R/W) = Unchanged from previous state 3h (R/W) = Reserved
24-23	TDM_CYCLE_FMT	R/W	0h	Cycle format (TDM mode only) 0h (R/W) = 1 cycle for 1 pixel 1h (R/W) = 2 cycles for 1 pixel 2h (R/W) = 3 cycles for 1 pixel 3h (R/W) = 3 cycles for 2 pixels
22-21	TDM_PARALLEL_MODE	R/W	0h	Output Interface width (TDM mode only) 0h (R/W) = 8-bit parallel output interface selected 1h (R/W) = 9-bit parallel output interface selected 2h (R/W) = 12-bit parallel output interface selected 3h (R/W) = 16-bit parallel output interface selected

Table 13-47. DISPC_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	TDM_EN	R/W	0h	Enable the multiple cycle format (TDM mode used only for Active Matrix mode with the RFBI enable bit off). 0h (R/W) = TDM disabled 1h (R/W) = TDM enabled
19-17	HT	R/W	0h	Hold Time for digital output WR: EVSYNC Encoded value (from 0 to 7) holds time for digital output. The data will be held for (HT + 1) external digital clock periods.
16	GPOUT1	R/W	0h	General Purpose Output Signal 0h (R/W) = The GPout1 is reset. 1h (R/W) = The GPout1 is set
15	GPOUT0	R/W	0h	General Purpose Output Signal 0h (R/W) = The GPout0 is reset 1h (R/W) = The GPout0 is set
14	GPIN1	R	0h	General Purpose Input Signal 0h (R) = The GPin1 has been reset 1h (R) = The GPin1 has been set
13	GPIN0	R	0h	General Purpose Input Signal 0h (R) = The GPin0 has been reset 1h (R) = The GPin0 has been set
12	OVLY_OPT	R/W	0h	Overlay Optimization (available when graphics format is NOT is 1, 2, and 4-BPP) 0h (R/W) = Graphics data below video1 window fetched from memory or no overlap between graphics and video1 windows 1h (R/W) = Graphics data below video1 window not fetched from memory.
11	STALL_MODE	R/W	0h	Stall mode for the LCD output 0h (R/W) = Normal mode selected 1h (R/W) = Stall mode selected. The Display Controller sends the data without considering the VSYNC/HSYNC. The LCD output is disabled at the end of the transfer of the frame. The S/W has to re-enable the LCD output to generate a new frame. The stall mode is used in RFBI command modes
10	RESERVED	R/W	0h	
9-8	TFT_DATA_LINES	R/W	0h	Number of lines of the LCD interface 0h (R/W) = 12-bit output aligned on the LSB of the pixel data interface 1h (R/W) = 16-bit output aligned on the LSB of the pixel data interface 2h (R/W) = 18-bit output aligned on the LSB of the pixel data interface 3h (R/W) = 24-bit output aligned on the LSB of the pixel data interface
7	ST_DITHER_EN	R/W	0h	Spatial temporal dithering enable 0h (R/W) = Spatial/temporal dithering logic disabled 1h (R/W) = Spatial/temporal dithering logic enabled

Table 13-47. DISPC_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	GO_DIGITAL	R/W	0h	Digital GO Command 0h (R/W) = The hardware has finished updating the internal shadow registers of the pipeline(s) associated with the digital output using the user values. The hardware resets the bit when the update is completed 1h (R/W) = Users have finished programming the shadow registers of the pipeline(s) associated with the digital output and the hardware can update the internal registers at the external VSYNC
5	GO_LCD	R/W	0h	LCD GO Command 0h (R/W) = The hardware has finished updating the internal shadow registers of the pipeline(s) connected to the LCD output using the user values. The hardware resets the bit when the update is completed 1h (R/W) = Users have finished programming the shadow registers of the pipeline(s) associated with the LCD output and the hardware can update the internal registers at the VFP start period.
4	M8B	R/W	0h	Mono 8-bit mode 0h (R/W) = Pixel data [3:0] is used to output four pixel values to the panel at each pixel clock transition (only in Passive Mono 8-bit mode) 1h (R/W) = Pixel data [7:0] is used to output eight pixel values to the panel each pixel clock transition (only in Passive Mono 8-bit mode)
3	STNTFT	R/W	0h	LCD display type 0h (R/W) = Passive or Passive Matrix display operation enabled. Passive Matrix dither logic enabled 1h (R/W) = Active Matrix display operation enabled. Passive Matrix Dither logic and output FIFO bypassed
2	MONO_COLOR	R/W	0h	Monochrome/Color 0h (R/W) = Color operation enabled (Passive Matrix mode only) 1h (R/W) = Monochrome operation enabled (Passive Matrix mode only)
1	DIGITAL_EN	R/W	0h	Digital enable 0h (R/W) = Digital output disabled (at the end of the current field if interlace output when the bit is reset) 1h (R/W) = Digital output enabled
0	LCD_EN	R/W	0h	LCD enable 0h (R/W) = LCD output disabled (at the end of the frame when the bit is reset) 1h (R/W) = LCD output disabled (at the end of the frame when the bit is reset)

13.6.1.7 DISPC_CFG Register (offset = 44h) [reset = 0h]

DISPC_CFG is shown in Figure 13-83 and described in Table 13-48.

This control register configures the display controller module.
Shadow register, updated on VFP start period or EVSYNC

Figure 13-83. DISPC_CFG Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				TV_ALPHA_BLDR_EN	LCD_APLHABLDR_EN	FIFO_FILLING	FIFO_HAND_CHECK
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CPR	FIFO_MERGE	TCK_DIG_SELECTION	TCK_DIG_EN	TCK_LCD_SELECTION	TCK_LCD_EN	FUNC_GATED	ACBIAS_GATED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VSYNC_GATED	HSYNC_GATED	PIXEL_CLK_GATED	PIXEL_DATA_GATED	PALETTEGAMMA_TBL	LOAD_MODE		PIXEL_GATED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h

Table 13-48. DISPC_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	TV_ALPHA_BLDR_EN	R/W	0h	Selects the alpha blender (TV output) 0h (R/W) = Alpha blender is disabled 1h (R/W) = The alpha blender is enabled
18	LCD_APLHABLDR_EN	R/W	0h	Selects the alpha blender (LCD output) 0h (R/W) = Alpha blender is disabled. 1h (R/W) = The alpha blender is enabled
17	FIFO_FILLING	R/W	0h	Controls if the FIFO are refilled only when the LOW threshold is reached or if all FIFO are refilled when at least one of them reaches the LOW threshold. 0h (R/W) = Each FIFO is refilled when it reaches LOW threshold 1h (R/W) = All FIFOs are refilled up to high threshold when at least one of them reaches the LOW threshold. (only active FIFOs should be considered and when reaching the end of the frame the FIFO goes to empty condition so no need to fill it again)
16	FIFO_HAND_CHECK	R/W	0h	Controls the handshake between FIFO and RFBI STALL to prevent from underflow. The bit should be set to 0 when the module is not in STALL mode 0h (R/W) = Only the STALL signal from RFBI is used regardless of the FIFO fullness information to provide data to the RFBI module 1h (R/W) = The STALL signal from RFBI is used in combination with the FIFO fullness information to provide data to the RFBI module only when it does not generated FIFO underflow.

Table 13-48. DISPC_CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	CPR	R/W	0h	Color phase rotation control wr: VFP 0h (R/W) = Color phase rotation disabled 1h (R/W) = Color phase rotation enabled
14	FIFO_MERGE	R/W	0h	FIFO merge control 0h (R/W) = FIFO merge disabled Each FIFO is dedicated to one pipeline 1h (R/W) = FIFO merge enabled All the FIFOS are merged into a single one to be used by the single active pipeline
13	TCK_DIG_SELECTION	R/W	0h	Transparency color key selection (digital output) 0h (R/W) = Graphics destination transparency color key selected in normal mode or graphics source transparency color key selected in alpha mode 1h (R/W) = Video source transparency color key selected in normal mode
12	TCK_DIG_EN	R/W	0h	Transparency color key enabled (digital output) 0h (R/W) = Disable the transparency color key for digital output 1h (R/W) = Enable the transparency color key for digital output
11	TCK_LCD_SELECTION	R/W	0h	Transparency color key selection (LCD output) 0h (R/W) = Graphics destination transparency color key selected in normal mode or graphics source transparency color key selected in alpha mode 1h (R/W) = Video source transparency color key selected in normal mode
10	TCK_LCD_EN	R/W	0h	Transparency color key enabled (LCD output) 0h (R/W) = Disable the transparency color key for the LCD 1h (R/W) = Enable the transparency color key for the LCD
9	FUNC_GATED	R/W	0h	Functional clocks gated enabled 0h (R/W) = Functional clocks gated disabled 1h (R/W) = Functional clocks gated enabled
8	ACBIAS_GATED	R/W	0h	ACBias Gated Enabled 0h (R/W) = ACBias Gated Disabled 1h (R/W) = ACBias Gated Enabled
7	VSYNC_GATED	R/W	0h	VSYNC Gated Enabled 0h (R/W) = VSYNC Gated Disabled 1h (R/W) = VSYNC Gated Enabled
6	HSYNC_GATED	R/W	0h	HSYNC Gated Enabled 0h (R/W) = HSYNC Gated Disabled 1h (R/W) = HSYNC Gated Enabled
5	PIXEL_CLK_GATED	R/W	0h	Pixel Clock Gated Enabled 0h (R/W) = Pixel Clock Gated Disabled 1h (R/W) = Pixel Clock Gated Enabled
4	PIXEL_DATA_GATED	R/W	0h	Pixel Data Gated Enabled 0h (R/W) = Pixel Data Gated Disabled 1h (R/W) = Pixel Data Gated Enabled

Table 13-48. DISPC_CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	PALETTEGAMMA_TBL	R/W	0h	Palette/Gamma Table selection 0h (R/W) = LUT used as palette (only if graphics format is BITMAP1, 2, 4, and 8) 1h (R/W) = LUT used as gamma table (only if graphics format is NOT BITMAP1, 2, 4, and 8 or no graphics window present)
2-1	LOAD_MODE	R/W	0h	Loading Mode for the Palette/Gamma Table 0h (R/W) = Palette/Gamma Table and data are loaded every frame 1h (R/W) = Palette/Gamma Table to be loaded. Users set the bit when the palette/gamma table has to be loaded. H/W resets the bit when table has been loaded. (DISPC_GFX_ATTRIBUTES. GfxEnable has to be set to 1) 2h (R/W) = Frame data only loaded every frame 3h (R/W) = Palette/Gamma Table and frame data loaded on first frame then switch to 10 (H/W)
0	PIXEL_GATED	R/W	0h	Pixel Gated Enable (only for Active Matrix Display) 0h (R/W) = Pixel Gated Enable (only for Active Matrix Display) 1h (R/W) = Pixel clock only toggles when there is valid data to display. (only in Active Matrix mode)

13.6.1.8 DISPC_DEFAULT_COLOR_0 Register (offset = 4Ch + [i * 4h]) [reset = 0h]

DISPC_DEFAULT_COLOR_0 is shown in Figure 13-84 and described in Table 13-49.

The control register allows to configure the default solid background color for the LCD (DISPC_DEFAULT_COLOR_0) and for 24-bit digital output (DISPC_DEFAULT_COLOR_1). Shadow register, updated on VFP start period for DISPC_DEFAULT_COLOR_0 and EVSYNC for DISPC_DEFAULT_COLOR_1

Figure 13-84. DISPC_DEFAULT_COLOR_0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="24">DEFAULT_COLOR</td></tr>
<tr><td colspan="8">R/W-0h</td><td colspan="24">R/W-0h</td></tr>
</table>

Table 13-49. DISPC_DEFAULT_COLOR_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-0	DEFAULT_COLOR	R/W	0h	24-bit RGB color value to specify the default solid color to display RW 0x000000 when there is no data from the overlays

13.6.1.9 DISPC_TRANS_COLOR_0 Register (offset = 54h + [i * 4h]) [reset = 0h]

DISPC_TRANS_COLOR_0 is shown in Figure 13-85 and described in Table 13-50.

The register sets the transparency color value for the video/graphics overlays for the LCD output (DISPC_TRANS_COLOR_0) for 24-bit digital output(DISPC_TRANS_COLOR_1).
Shadow register, updated on VFP start period for DISPC_TRANS_COLOR_0 and EVSYNC for DISPC_TRANS_COLOR_1

Figure 13-85. DISPC_TRANS_COLOR_0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="24">TRANS_COLOR_KEY</td></tr>
<tr><td colspan="8">R/W-0h</td><td colspan="24">R/W-0h</td></tr>
</table>

Table 13-50. DISPC_TRANS_COLOR_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-0	TRANS_COLOR_KEY	R/W	0h	Transparency Color Key Value in RGB format [0] BITMAP 1 (CLUT), [23,1] set to 0s [1:0] BITMAP 2 (CLUT), [23,2] set to 0s [3:0] BITMAP 4 (CLUT), [23,4] set to 0s [7:0] BITMAP 8 (CLUT), [23,8] set to 0s [11:0] RGB 12, [23,12] set to 0s [15:0] RGB 16, [23,16] set to 0s [23:0] RGB 24

13.6.1.10 DISPC_LINE_STS Register (offset = 5Ch) [reset = 7FFh]

DISPC_LINE_STS is shown in Figure 13-86 and described in Table 13-51.

The control register indicates the current LCD panel display line number

Figure 13-86. DISPC_LINE_STS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="21">RESERVED</td><td colspan="11">LINE_NUMBER</td></tr>
<tr><td colspan="21">R-0h</td><td colspan="11">R-7FFh</td></tr>
</table>

Table 13-51. DISPC_LINE_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10-0	LINE_NUMBER	R	7FFh	Current LCD panel line number Current display line number. The first active line has the value 0. During blanking lines the line number is not incremented

13.6.1.11 DISPC_LINE_NUMBER Register (offset = 60h) [reset = 0h]

DISPC_LINE_NUMBER is shown in Figure 13-87 and described in Table 13-52.

The control register indicates the LCD panel display line number for the interrupt and the DMA request. Shadow register, updated on VFP start period

Figure 13-87. DISPC_LINE_NUMBER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="21">RESERVED</td><td colspan="11">LINE_NUMBER</td></tr>
<tr><td colspan="21">R/W-0h</td><td colspan="11">R/W-0h</td></tr>
</table>

Table 13-52. DISPC_LINE_NUMBER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-0	LINE_NUMBER	R/W	0h	LCD panel line number programming LCD line number defines the line on which the programmable interrupt is generated and the DMA request occurs

13.6.1.12 DISPC_TIMING_H Register (offset = 64h) [reset = 0h]

DISPC_TIMING_H is shown in Figure 13-88 and described in Table 13-53.

The register configures the timing logic for the HSYNC signal.
Shadow register, updated on VFP start period

Figure 13-88. DISPC_TIMING_H Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="12">HBP</td><td colspan="12">HFP</td><td colspan="8">HSW</td></tr>
<tr><td colspan="12">R/W-0h</td><td colspan="12">R/W-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 13-53. DISPC_TIMING_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	HBP	R/W	0h	Horizontal Back Porch. Encoded value (from 1 to 4096) to specify the number of pixel clock periods to add to the beginning of a line transmission before the first set of pixels is output to the display (program to value minus 1)
19-8	HFP	R/W	0h	Horizontal front porch. Encoded value (from 1 to 4096) to specify the number of pixel clock periods to add to the end of a line transmission before line clock is asserted (program to value minus 1)
7-0	HSW	R/W	0h	Horizontal synchronization pulse width Encoded value (from 1 to 256) to specify the number of pixel clock periods to pulse the line clock at the end of each line (program to value minus 1).

13.6.1.13 DISPC_TIMING_V Register (offset = 68h) [reset = 0h]

DISPC_TIMING_V is shown in Figure 13-89 and described in Table 13-54.

The register configures the timing logic for the VSYNC signal.
Shadow register, updated on VFP start period

Figure 13-89. DISPC_TIMING_V Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="12">VBP</td><td colspan="12">VFP</td><td colspan="8">VSW</td></tr>
<tr><td colspan="12">R/W-0h</td><td colspan="12">R/W-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 13-54. DISPC_TIMING_V Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	VBP	R/W	0h	Vertical back porch Encoded value (from 0 to 4095) to specify the number of line clock periods to add to the beginning of a frame before the first set of pixels is output to the display
19-8	VFP	R/W	0h	Vertical front porch Encoded value (from 0 to 4095) to specify the number of line clock periods to add to the end of each frame
7-0	VSW	R/W	0h	Vertical synchronization pulse width In active mode, encoded value (from 1 to 256) to specify the number of line clock periods (program to value minus one) to pulse the frame clock (VSYNC) pin at the end of each frame after the end of frame wait (VFP) period elapses. Frame clock uses as VSYNC signal in active mode. In passive mode, encoded value (from 1 to 256) to specify the number of extra line clock periods (program to value minus one) to insert after the vertical front porch (VFP) period has elapsed.

13.6.1.14 DISPC_POL_FREQ Register (offset = 6Ch) [reset = 0h]

DISPC_POL_FREQ is shown in Figure 13-90 and described in Table 13-55.

The register configures the signal configuration.
Shadow register, updated on VFP start period

Figure 13-90. DISPC_POL_FREQ Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED						ONOFF	RF
R/W-0h						R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
IEO	IPC	IHS	IVS	ACBI			
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ACB							
R/W-0h							

Table 13-55. DISPC_POL_FREQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17	ONOFF	R/W	0h	HSYNC/VSYNC Pixel clock Control On/Off 0h (R/W) = HSYNC and VSYNC are driven on opposite edges of pixel clock than pixel data 1h (R/W) = HSYNC and VSYNC are driven according to bit 16
16	RF	R/W	0h	Program HSYNC/VSYNC Rise or Fall 0h (R/W) = HSYNC and VSYNC are driven on falling edge of pixel clock (if bit 17 set to 1) 1h (R/W) = HSYNC and VSYNC are driven on the rising edge of pixel clock (if bit 17 set to 1)
15	IEO	R/W	0h	Invert output enable 0h (R/W) = Ac-bias is active high (active display mode) 1h (R/W) = Ac-bias is active low (active display mode)
14	IPC	R/W	0h	Invert pixel clock 0h (R/W) = Data is driven on the LCD data lines on the rising-edge of the pixel clock 1h (R/W) = Data is driven on the LCD data lines on the falling-edge of the pixel clock
13	IHS	R/W	0h	Invert HSYNC 0h (R/W) = Line clock pin is active high and inactive low 1h (R/W) = Line clock pin is active low and inactive high
12	IVS	R/W	0h	Invert VSYNC 0h (R/W) = Frame clock pin is active high and inactive low 1h (R/W) = Frame clock pin is active low and inactive high

Table 13-55. DISPC_POL_FREQ Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-8	ACBI	R/W	0h	AC-bias pin transitions per interrupt Value (from 0 to 15) used to specify the number of AC Bias pin transitions
7-0	ACB	R/W	0h	AC-bias pin frequency Value (from 0 to 255) used to specify the number of line clocks to count before transitioning the ac-bias pin. This pin is used to periodically invert the polarity of the power supply to prevent DC charge build-up within the display.

13.6.1.15 DISPC_DIVISOR Register (offset = 70h) [reset = 00010002h]

DISPC_DIVISOR is shown in Figure 13-91 and described in Table 13-56.

The register configures the divisors.
Shadow register, updated on VFP start period

Figure 13-91. DISPC_DIVISOR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">LCD</td><td colspan="8">RESERVED</td><td colspan="8">PCD</td></tr>
<tr><td colspan="8">R/W-0h</td><td colspan="8">R/W-1h</td><td colspan="8">R/W-0h</td><td colspan="8">R/W-2h</td></tr>
</table>

Table 13-56. DISPC_DIVISOR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-16	LCD	R/W	1h	Display Controller Logic Clock Divisor Value (from 1 to 255) to specify the frequency of the display controller logic clock based on the function clock. The value 0 is invalid.
15-8	RESERVED	R/W	0h	
7-0	PCD	R/W	2h	Pixel Clock Divisor Value (from 1 to 255) to specify the frequency of the pixel clock based on the Logic clock which is the functional clock divided by LCD. The values 0 and 1 are invalid.

13.6.1.16 DISPC_GLOBAL_ALPHA Register (offset = 74h) [reset = 0h]

DISPC_GLOBAL_ALPHA is shown in Figure 13-92 and described in Table 13-57.

The register defines the global alpha value for the graphics and video 2 pipelines. Shadow register, updated on VFP start period or EVSYNC for each bit field depending on the association of the each pipeline with the LCD or TV output.

Figure 13-92. DISPC_GLOBAL_ALPHA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								VID2_GLOBAL_ALPHA							
R/W-0h								R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED								GFX_GLOBAL_ALPHA							
R/W-0h								R/W-0h							

Table 13-57. DISPC_GLOBAL_ALPHA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-16	VID2_GLOBAL_ALPHA	R/W	0h	Global alpha value from 0 to 255. 0 corresponds to fully transparent and 255 to fully opaque
15-8	RESERVED	R/W	0h	
7-0	GFX_GLOBAL_ALPHA	R/W	0h	Global alpha value from 0 to 255. 0 corresponds to fully transparent and 255 to fully opaque

13.6.1.17 DISPC_SIZE_DIG Register (offset = 78h) [reset = 0h]

DISPC_SIZE_DIG is shown in Figure 13-93 and described in Table 13-58.

The register configures the size of the digital output field (interlace), frame (progressive) (horizontal and vertical).
Shadow register, updated on EVSYNC

Figure 13-93. DISPC_SIZE_DIG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					LPP										
R/W-0h					R/W-0h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PPL										
R/W-0h					R/W-0h										

Table 13-58. DISPC_SIZE_DIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	LPP	R/W	0h	Lines per panel Encoded value (from 1 to 2048) to specify the number of lines per panel (program to value minus one)
15-11	RESERVED	R/W	0h	
10-0	PPL	R/W	0h	Pixels per line Encoded value (from 1 to 2048) to specify the number of pixels contained within each line on the display (program to value minus one)

13.6.1.18 DISPC_SIZE_LCD Register (offset = 7Ch) [reset = 0h]

DISPC_SIZE_LCD is shown in Figure 13-94 and described in Table 13-59.

The register configures the panel size (horizontal and vertical).
Shadow register, updated on VFP start period

Figure 13-94. DISPC_SIZE_LCD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					LPP										
R/W-0h					R/W-0h										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PPL										
R/W-0h					R/W-0h										

Table 13-59. DISPC_SIZE_LCD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	LPP	R/W	0h	Lines per panel Encoded value (from 1 to 2048) to specify the number of lines per panel (program to value minus one)
15-11	RESERVED	R/W	0h	
10-0	PPL	R/W	0h	Pixels per line Encoded value (from 1 to 2048) to specify the number of pixels contains within each line on the display (program to value minus one). When running in normal mode (stall mode is bypassed by setting DSS.DISPC_CONTROL[11] STALLMODE =0) the line width must be set to a value multiple of 8 pixels (ex: PPL=0x7)

13.6.1.19 DISPC_GFX_BA_0 Register (offset = 80h + [i * 4h]) [reset = 0h]

DISPC_GFX_BA_0 is shown in Figure 13-95 and described in Table 13-60.

The register configures the base address of the graphics buffer displayed in the graphics window (0 1 :for ping-pong mechanism with external trigger, based on the field polarity, 0 only used when graphics pipeline on the LCD
output and 0 1 when on the 24-bit digital output).
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-95. DISPC_GFX_BA_0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">GFX_BA</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 13-60. DISPC_GFX_BA_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GFX_BA	R/W	0h	Graphics base address Base address of the graphics buffer (aligned on pixel size boundary) (in case 1-, 2-, and 4-BPP, byte alignment is required)

13.6.1.20 DISPC_GFX_POSITION Register (offset = 88h) [reset = 0h]

DISPC_GFX_POSITION is shown in Figure 13-96 and described in Table 13-61.

The register configures the position of the graphics window.
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-96. DISPC_GFX_POSITION Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					GFX_POSY										
R/W-0h					R/W-0h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					GFX_POSX										
R/W-0h					R/W-0h										

Table 13-61. DISPC_GFX_POSITION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	GFX_POSY	R/W	0h	Y position of the graphics window. Encoded value (from 0 to 2047) to specify the Y position of the graphics window on the screen. The line at the top has the Y-position 0.
15-11	RESERVED	R/W	0h	
10-0	GFX_POSX	R/W	0h	X position of the graphics window. Encoded value (from 0 to 2047) to specify the X position of the graphics window on the screen. The first pixel on the left of the screen has the X-position 0.

13.6.1.21 DISPC_GFX_SIZE Register (offset = 8Ch) [reset = 0h]

DISPC_GFX_SIZE is shown in Figure 13-97 and described in Table 13-62.

The register configures the size of the graphics window.
Shadow register, updated on VFP start period or EVSYNC

Figure 13-97. DISPC_GFX_SIZE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					GFX_SIZEY										
R/W-0h					R/W-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					GFX_SIZEX										
R/W-0h					R/W-0h										

Table 13-62. DISPC_GFX_SIZE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	GFX_SIZEY	R/W	0h	Number of lines of the graphics window. Encoded value (from 1 to 2048) to specify the number of lines of the graphics window (program to value minus one)
15-11	RESERVED	R/W	0h	
10-0	GFX_SIZEX	R/W	0h	Number of pixels of the graphics window. Encoded value (from 1 to 2048) to specify the number of pixels per line of the graphics window (program to value minus one)

13.6.1.22 DISPC_GFX_ATTRS Register (offset = A0h) [reset = 0h]

DISPC_GFX_ATTRS is shown in Figure 13-98 and described in Table 13-63.

The register configures the graphics attributes.
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-98. DISPC_GFX_ATTRS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="3">RESERVED</td><td>PRE_MULTIPLY_ALPHA</td><td colspan="4">RESERVED</td></tr>
<tr><td colspan="3">R/W-0h</td><td>R/W-0h</td><td colspan="4">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>GFX_SELF_REFRESH</td><td>GFX_ARBITRATION</td><td colspan="2">GFX_ROTATION</td><td>GFX_FIFO_PRELOAD</td><td>GFX_ENDIAN</td><td>GFX_NIBBLE_MODE</td><td>GFX_CHANNEL_OUT</td></tr>
<tr><td>R/W-0h</td><td>R/W-0h</td><td colspan="2">R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">GFX_BURST_SIZE</td><td>GFX_REPLICATION_EN</td><td colspan="4">GFX_FMT</td><td>GFX_EN</td></tr>
<tr><td colspan="2">R/W-0h</td><td>R/W-0h</td><td colspan="4">R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 13-63. DISPC_GFX_ATTRS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	PRE_MULTIPLY_ALPHA	R/W	0h	The field configures the DISPC GFX to process incoming data as pre-multiplied alpha data or non premultiplied alpha data. Default setting is non pre-multiplied alpha data 0h (R/W) = Non pre-multiplyalpha data color component 1h (R/W) = Pre-multiplyalpha data color component
27-16	RESERVED	R/W	0h	
15	GFX_SELF_REFRESH	R/W	0h	Enables the self refresh of the graphics window from its own FIFO only 0h (R/W) = The graphics pipeline accesses the interconnect to fetch data from the system memory 1h (R/W) = The graphics pipeline does not need anymore to fetch data from memory. Only the graphics FIFO is used. It takes effect after the frame has been loaded in the FIFO
14	GFX_ARBITRATION	R/W	0h	Determines the priority of the graphics pipeline. The graphics pipeline is one of the high priority pipeline. The arbitration wheel gives always the priority first to the high priority pipelines using round-robin between them. When there is only normal priority pipelines sending requests, the round-robin applies between them 0h (R/W) = The graphics pipeline is one of the normal priority pipeline 1h (R/W) = The graphics pipeline is one of the high priority pipeline
13-12	GFX_ROTATION	R/W	0h	Graphics rotation flag (used only in case of RGB24 packed format) 0h (R/W) = No rotation 1h (R/W) = Rotation by 90 degrees 2h (R/W) = Rotation by 180 degrees 3h (R/W) = Rotation by 270 degrees

Table 13-63. DISPC_GFX_ATTRS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	GFX_FIFO_PRELOAD	R/W	0h	Graphics preload value 0h (R/W) = H/W prefetches pixels up to the preload value defined in the preload register. 1h (R/W) = H/W prefetches pixels up to high threshold value
10	GFX_ENDIAN	R/W	0h	Graphics endianness 0h (R/W) = Little endian operation is selected 1h (R/W) = Big endian operation is selected
9	GFX_NIBBLE_MODE	R/W	0h	Graphics Nibble Mode (only for 1, 2 and 4 BPP) 0h (R/W) = Nibble mode is disabled 1h (R/W) = Nibble mode is enabled
8	GFX_CHANNEL_OUT	R/W	0h	Graphics Channel Out configuration 0h (R/W) = LCD output selected 1h (R/W) = 24-bit output selected
7-6	GFX_BURST_SIZE	R/W	0h	Graphics DMA Burst Size 0h (R/W) = 4x32bit bursts 1h (R/W) = 8x32bit bursts 2h (R/W) = 16x32bit bursts 3h (R/W) = 3
5	GFX_REPLICATION_EN	R/W	0h	GfxReplicationEnable 0h (R/W) = Disable Graphics replication logic 1h (R/W) = Enable Graphics replication logic
4-1	GFX_FMT	R/W	0h	Graphics format Other enums: Reserved (0x7, 0xA, 0xB and 0xF) 0h (R/W) = BITMAP 1 (CLUT) 1h (R/W) = BITMAP 2 (CLUT) 2h (R/W) = BITMAP 4 (CLUT) 3h (R/W) = BITMAP 8 (CLUT) 4h (R/W) = RGB 12 (un-packed in 16-bit container) 5h (R/W) = ARGB16 6h (R/W) = RGB 16 8h (R/W) = RGB 24 (un-packed in 32-bit container) 9h (R/W) = RGB 24 (packed in 24-bit container) Ch (R/W) = ARGB32 Dh (R/W) = RGBA32 Eh (R/W) = RGBx 32 (24-bit RGB aligned on MSB of the 32-bit container)
0	GFX_EN	R/W	0h	GfxEnable 0h (R/W) = Graphics disabled (graphics pipeline inactive and graphics window not present) 1h (R/W) = Graphics enabled (graphics pipeline active and graphics window present on the screen)

13.6.1.23 DISPC_GFX_FIFO_THR Register (offset = A4h) [reset = 3FF03C0h]

DISPC_GFX_FIFO_THR is shown in Figure 13-99 and described in Table 13-64.

The register configures the graphics FIFO.
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-99. DISPC_GFX_FIFO_THR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				GFX_FIFO_HIGH_THR											
R/W-0h				R/W-3FFh											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				GFX_FIFO_LOW_THR											
R/W-0h				R/W-3C0h											

Table 13-64. DISPC_GFX_FIFO_THR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-16	GFX_FIFO_HIGH_THR	R/W	3FFh	Graphics FIFO High Threshold Number of bytes defining the threshold value.
15-12	RESERVED	R/W	0h	
11-0	GFX_FIFO_LOW_THR	R/W	3C0h	Graphics FIFO Low Threshold Number of bytes defining the threshold value

13.6.1.24 DISPC_GFX_FIFO_SIZE_STS Register (offset = A8h) [reset = 400h]

DISPC_GFX_FIFO_SIZE_STS is shown in Figure 13-100 and described in Table 13-65.

This register defines the graphics FIFO size.

Figure 13-100. DISPC_GFX_FIFO_SIZE_STS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">RESERVED</td><td colspan="11">GFX_FIFO_SIZE</td></tr>
<tr><td colspan="5">R-0h</td><td colspan="11">R-400h</td></tr>
</table>

Table 13-65. DISPC_GFX_FIFO_SIZE_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10-0	GFX_FIFO_SIZE	R	400h	Graphics FIFO Size Number of bytes defining the FIFO value.

13.6.1.25 DISPC_GFX_ROW_INC Register (offset = ACh) [reset = 1h]

DISPC_GFX_ROW_INC is shown in Figure 13-101 and described in Table 13-66.

The register configures the number of bytes to increment at the end of the row.
Shadow register, updated on VFP start period or EVSYNC

Figure 13-101. DISPC_GFX_ROW_INC Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">GFX_ROW_INC</td></tr>
<tr><td colspan="32">R/W-1h</td></tr>
</table>

Table 13-66. DISPC_GFX_ROW_INC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GFX_ROW_INC	R/W	1h	Number of bytes to increment at the end of the row Encoded signed value (from (-2^31)-1 to 2^31) to specify the number of bytes to increment at the end of the row in the graphics buffer. The value 0 is invalid. The value 1 means next pixel. The value 1+n*BPP means increment of n pixels. The value 1- (n+1)*BPP means decrement of n pixels.

13.6.1.26 DISPC_GFX_PIXEL_INC Register (offset = B0h) [reset = 0h]

DISPC_GFX_PIXEL_INC is shown in Figure 13-102 and described in Table 13-67.

The register configures the number of bytes to increment between two pixels.
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-102. DISPC_GFX_PIXEL_INC Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">GFX_PIXEL_INC</td></tr>
<tr><td colspan="16">R/W-0h</td><td colspan="16">R/W-0h</td></tr>
</table>

Table 13-67. DISPC_GFX_PIXEL_INC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	GFX_PIXEL_INC	R/W	0h	Number of bytes to increment between two pixels Encoded signed value (from (-2^15)-1 to 2^15) to specify the number of bytes between two pixels in the graphics buffer. The value 0 is invalid. The value 1 means next pixel. The value 1+n*BPP means increment of n pixels. The value 1-(n+1)*BPP means decrement of n pixels.

13.6.1.27 DISPC_GFX_WINDOW_SKIP Register (offset = B4h) [reset = 0h]

DISPC_GFX_WINDOW_SKIP is shown in Figure 13-103 and described in Table 13-68.

The register configures the number of bytes to skip during video window display.
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-103. DISPC_GFX_WINDOW_SKIP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GFX_WINDOW_SKIP																															
R/W-0h																															

Table 13-68. DISPC_GFX_WINDOW_SKIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GFX_WINDOW_SKIP	R/W	0h	Number of bytes to skip during video window #1. Encoded signed value (from (-2^31)-1 to 2^31) to specify the number of bytes to skip in the graphics buffer when video window #1 is displayed on top of the graphics and no transparency color is enabled.

13.6.1.28 DISPC_GFX_TBL_BA Register (offset = B8h) [reset = 0h]

DISPC_GFX_TBL_BA is shown in Figure 13-104 and described in Table 13-69.

The register configures the base address of the palette buffer or the gamma table buffer.
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-104. DISPC_GFX_TBL_BA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GFX_TBL_BA																															
R/W-0h																															

Table 13-69. DISPC_GFX_TBL_BA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GFX_TBL_BA	R/W	0h	Base address of the palette/gamma table buffer (24-bit entries in 32-bit containers, aligned on 32-bit boundary).

13.6.1.29 DISPC_VID1_BA_0 to DISPC_VID1_BA_1 Register (offset = BCh to C0h) [reset = 0h]

DISPC_VID1_BA_0 to DISPC_VID1_BA_1 is shown in Figure 13-105 and described in Table 13-70.

The register configures the base address of the video buffer for video window #n(#j for ping-pong mechanism with
external trigger, based on the field polarity: 0 for even field and 1 for odd field).
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-105. DISPC_VID1_BA_0 to DISPC_VID1_BA_1 Register

31-0
BA
R/W-0h

Table 13-70. DISPC_VID1_BA_0 to DISPC_VID1_BA_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BA	R/W	0h	Video base address Base address of the video buffer (aligned on pixel size boundary)

13.6.1.30 DISPC_VID1_POSITION Register (offset = C4h) [reset = 0h]

DISPC_VID1_POSITION is shown in Figure 13-106 and described in Table 13-71.

The register configures the position of video window #n.
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-106. DISPC_VID1_POSITION Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					POSY										
R/W-0h					R/W-0h										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					POSX										
R/W-0h					R/W-0h										

Table 13-71. DISPC_VID1_POSITION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	POSY	R/W	0h	Y position of video window #n Encoded value (from 0 to 2047) to specify the Y position of video window #n. The line at the top has the Y-position 0.
15-11	RESERVED	R/W	0h	
10-0	POSX	R/W	0h	X position of video window #n Encoded value (from 0 to 2047) to specify the X position of video window #n. The first pixel on the left of the display screen has the X-position 0.

13.6.1.31 DISPC_VID1_SIZE Register (offset = C8h) [reset = 0h]

DISPC_VID1_SIZE is shown in Figure 13-107 and described in Table 13-72.

The register configures the size of video window #n.
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-107. DISPC_VID1_SIZE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					VID_SIZE_Y										
R/W-0h					R/W-0h										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					VID_SIZE_X										
R/W-0h					R/W-0h										

Table 13-72. DISPC_VID1_SIZE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	VID_SIZE_Y	R/W	0h	Number of lines of video #n Encoded value (from 1 to 2048) to specify the number of lines of video window #n (program to value minus one).
15-11	RESERVED	R/W	0h	
10-0	VID_SIZE_X	R/W	0h	Number of pixels of video window #n Encoded value (from 1 to 2048) to specify the number of pixels of video window #n (program to value minus one).

13.6.1.32 DISPC_VID1_ATTRS Register (offset = CCh) [reset = 0h]

DISPC_VID1_ATTRS is shown in Figure 13-108 and described in Table 13-73.

Figure 13-108. DISPC_VID1_ATTRS Register

31	30	29	28	27	26	25	24
RESERVED			PRE_MULTIPLY_ALPHA	RESERVED			SELF_REFRESH
R/W-0h			R/W-0h	R/W-0h			R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ARBITRATION	LINE_BUFFER_SPLIT	VERTICAL_TAPS	OPTIMIZATION	FIFO_PRELOAD	ROW_REPEAT_EN	ENDIAN	CHANNEL_OUT
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BURST_SIZE		ROTATION		FULL_RANGE	REPLICATION_EN	COLOR_CONV_EN	VRESIZE_CONF
R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
HRESIZE_CONF	RESIZE_EN		FMT				EN
R/W-0h	R/W-0h		R/W-0h				R/W-0h

Table 13-73. DISPC_VID1_ATTRS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	PRE_MULTIPLY_ALPHA	R/W	0h	The field configures the DISPC VID2 to process incoming data as pre-multiplied alpha data or non pre-multiplied alpha data. Default setting is non pre-multiplied alpha data. 0h (R/W) = Non pre-multiplyalpha data color component 1h (R/W) = Premultiplyalpha data color component
27-25	RESERVED	R/W	0h	
24	SELF_REFRESH	R/W	0h	Enables the self refresh of the video window from its own FIFO only. 0h (R/W) = The video pipeline accesses the interconnect to fetch data from the system memory 1h (R/W) = The video pipeline does not need anymore to fetch data from memory. Only the video FIFO is used. It takes effect after the frame has been loaded in the FIFO
23	ARBITRATION	R/W	0h	Determines the priority of the video pipeline. The video pipeline is one of the high priority pipeline. The arbitration wheel gives always the priority first to the high priority pipelines using round-robin between them. When there is only normal priority pipelines sending requests, the round-robin applies between them. 0h (R/W) = The video pipeline is one of the normal priority pipeline 1h (R/W) = The video pipeline is one of the high priority pipeline
22	LINE_BUFFER_SPLIT	R/W	0h	Video vertical line buffer split 0h (R/W) = Vertical line buffers are not split 1h (R/W) = Vertical line buffers are split into two
21	VERTICAL_TAPS	R/W	0h	Video vertical resize tap number 0h (R/W) = Three taps are used for the vertical filtering logic. The other two taps are not used 1h (R/W) = Five taps are used for the vertical filtering logic

Table 13-73. DISPC_VID1_ATTRS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	OPTIMIZATION	R/W	0h	Video optimization in case of 0h (R/W) = The DMA engine fetches one pixel for each 32-bit OCP request (RGB16 and YUV422) while doing 90- and 270-degree rotation (accessing on-chip memory and off-chip memory). 1h (R/W) = The DMA engine fetches two pixels for each 32-bit OCP request (RGB16 and YUV422) while doing 90- and 270-degree rotation (accessing on-chip memory and off-chip memory). The bit field [21] VIDVERTICALTAPS shall be set to 0x1, bit field [22] VIDLINEBUFFERSPLIT to 0x1, and all scaler registers shall be configured even for 1:1 ratio. Even width is required for the input picture when 5 taps are used.
19	FIFO_PRELOAD	R/W	0h	Video preload value 0h (R/W) = H/W prefetches pixels up to the preload value defined in the preload register 1h (R/W) = H/W prefetches pixels up to the high threshold value.
18	ROW_REPEAT_EN	R/W	0h	Video Row Repeat (YUV case only when rotating 90 or 270-degree) 0h (R/W) = Row of VIDn won't be read twice 1h (R/W) = The Row data are fetched twice to extract both the Y components
17	ENDIAN	R/W	0h	Video Endianness 0h (R/W) = Little endian operation is selected 1h (R/W) = Big endian operation is selected.
16	CHANNEL_OUT	R/W	0h	Video Channel Out configuration 0h (R/W) = LCD output selected 1h (R/W) = 24 bit output selected
15-14	BURST_SIZE	R/W	0h	Video DMA Burst Size 0h (R/W) = 4x32bit bursts 1h (R/W) = 8x32bit bursts 2h (R/W) = 16x32bit bursts 3h (R/W) = 3
13-12	ROTATION	R/W	0h	Video Rotation Flag 0h (R/W) = No rotation or VidFormat is RGB 1h (R/W) = Rotation by 90 degrees 2h (R/W) = Rotation by 180 degrees 3h (R/W) = Rotation by 270 degrees
11	FULL_RANGE	R/W	0h	VidFullRange 0h (R/W) = Limited range selected: 16 subtracted from Y before color space conversion 1h (R/W) = Full range selected: Y is not modified before the color space conversion
10	REPLICATION_EN	R/W	0h	VidReplicationEnable 0h (R/W) = Disable Video replication logic 1h (R/W) = Enable Video replication logic
9	COLOR_CONV_EN	R/W	0h	VidColorConvEnable 0h (R/W) = Disable Color Space Conversion CbYCr to RGB 1h (R/W) = Enable Color Space Conversion CbYCr to RGB

Table 13-73. DISPC_VID1_ATTRS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	VRESIZE_CONF	R/W	0h	Video Vertical Resize Configuration 0h (R/W) = Up-sampling selected 1h (R/W) = Down-sampling selected
7	HRESIZE_CONF	R/W	0h	Video Horizontal Resize Configuration 0h (R/W) = Up-sampling selected 1h (R/W) = Down-sampling selected
6-5	RESIZE_EN	R/W	0h	Video Resize Enable 0h (R/W) = Disable the resize processing 1h (R/W) = Enable the horizontal resize processing 2h (R/W) = Enable the vertical resize processing 3h (R/W) = Enable both horizontal and vertical resize processing
4-1	FMT	R/W	0h	Video2 channel Format Other enums: Reserved (all other values: 0x0 and 0x3, 0x7, and 0xF) 0h (R/W) = RESERVED : 0 1h (R/W) = RESERVED : 1 2h (R/W) = RESERVED : 2 3h (R/W) = RESERVED : 3 4h (R/W) = RGB 12 (16-bit container) 5h (R/W) = ARGB 16 6h (R/W) = RGB 16 7h (R/W) = RESERVED : 7 8h (R/W) = RGB 24 (un-packed in 32-bit container) 9h (R/W) = RGB 24 (packed in 24-bit container) Ah (R/W) = YUV2 4:2:2 co-sited Bh (R/W) = UYVY 4:2:2 co-sited Ch (R/W) = ARGB 32 Dh (R/W) = RGBA 32 Eh (R/W) = RGBx 32 (24-bit RGB aligned on MSB of the 32-bit container) Fh (R/W) = RESERVED : 15
0	EN	R/W	0h	VidEnable 0h (R/W) = Video disabled (video pipeline inactive and window not present) 1h (R/W) = Video enabled (video pipeline active and window present on the screen)

13.6.1.33 DISPC_VID1_FIFO_THR Register (offset = D0h) [reset = 3C003FFh]

DISPC_VID1_FIFO_THR is shown in Figure 13-109 and described in Table 13-74.

The register configures the video FIFO associated with video pipeline #n.
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-109. DISPC_VID1_FIFO_THR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				FIFO_HIGH_THR											
R/W-0h				R/W-3C0h											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				FIFO_LOW_THR											
R/W-0h				R/W-3FFh											

Table 13-74. DISPC_VID1_FIFO_THR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-16	FIFO_HIGH_THR	R/W	3C0h	Video FIFO high threshold Number of bytes defining the threshold value
15-12	RESERVED	R/W	0h	
11-0	FIFO_LOW_THR	R/W	3FFh	Video FIFO low threshold Number of bytes defining the threshold value

13.6.1.34 DISPC_VID1_FIFO_SIZE_STS Register (offset = D4h) [reset = 400h]

DISPC_VID1_FIFO_SIZE_STS is shown in Figure 13-110 and described in Table 13-75.

The register defines the video FIFO size for video pipeline #n.

Figure 13-110. DISPC_VID1_FIFO_SIZE_STS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="21">RESERVED</td><td colspan="11">FIFO_SIZE</td></tr>
<tr><td colspan="21">R/W-0h</td><td colspan="11">R/W-400h</td></tr>
</table>

Table 13-75. DISPC_VID1_FIFO_SIZE_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-0	FIFO_SIZE	R/W	400h	Video FIFO Size Number of bytes defining the FIFO value

13.6.1.35 DISPC_VID1_ROW_INC Register (offset = D8h) [reset = 1h]

DISPC_VID1_ROW_INC is shown in Figure 13-111 and described in Table 13-76.

The register configures the number of bytes to increment at the end of the row for the buffer associated with video window #n.
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-111. DISPC_VID1_ROW_INC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROW_INC																															
R/W-1h																															

Table 13-76. DISPC_VID1_ROW_INC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ROW_INC	R/W	1h	Number of bytes to increment at the end of the row Encoded signed value (from (-2^31)-1 to 2^31) to specify the number of bytes to increment at the end of the row in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value 1+n*BPP means increment of n pixels. The value 1- (n+1)*BPP means decrement of n pixels.

13.6.1.36 DISPC_VID1_PIXEL_INC Register (offset = DCh) [reset = 1h]

DISPC_VID1_PIXEL_INC is shown in Figure 13-112 and described in Table 13-77.

The register configures the number of bytes to increment between two pixels for the buffer associated with video
window #n.
Shadow register, updated on VFP start period or EVSYNC

Figure 13-112. DISPC_VID1_PIXEL_INC Register

31-16	15-0
RESERVED	PIXEL_INC
R/W-0h	R/W-1h

Table 13-77. DISPC_VID1_PIXEL_INC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	PIXEL_INC	R/W	1h	Number of bytes to increment at the end of the row Encoded signed value (from (-2^15)-1 to 2^15) to specify the number of bytes between two pixels in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value 1+n*BPP means increment of n pixels. The value 1-(n+1)*BPP means decrement of n pixels

13.6.1.37 DISPC_VID1_FIR Register (offset = E0h) [reset = 0h]

DISPC_VID1_FIR is shown in Figure 13-113 and described in Table 13-78.

The register configures the resize factors for horizontal and vertical up-/down-sampling of video window #n.
Shadow register, updated on VFP start period or EVSYNC

Figure 13-113. DISPC_VID1_FIR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED			FIR_V_INC												
R/W-0h			R/W-0h												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			FIR_H_INC												
R/W-0h			R/W-0h												

Table 13-78. DISPC_VID1_FIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28-16	FIR_V_INC	R/W	0h	Vertical increment of the up-/down-sampling filter Encoded value (from 1 to 4096). The value 0 is invalid. Values greater than 4096 are invalid.
15-13	RESERVED	R/W	0h	
12-0	FIR_H_INC	R/W	0h	Horizontal increment of the up-/down-sampling filter Encoded value (from 1 to 4096). The value 0 is invalid. Values greater than 4096 are invalid.

13.6.1.38 DISPC_VID1_PICTURE_SIZE Register (offset = E4h) [reset = 0h]

DISPC_VID1_PICTURE_SIZE is shown in Figure 13-114 and described in Table 13-79.

The register configures the size of the video picture associated with video layer #nbefore up-/down-scaling.
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-114. DISPC_VID1_PICTURE_SIZE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					VID_ORG_SIZE_Y										
R/W-0h					R/W-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					VID_ORG_SIZE_X										
R/W-0h					R/W-0h										

Table 13-79. DISPC_VID1_PICTURE_SIZE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	VID_ORG_SIZE_Y	R/W	0h	Number of lines of the video picture Encoded value (from 1 to 2048) to specify the number of lines of the video picture in memory (program to value minus one)
15-11	RESERVED	R/W	0h	
10-0	VID_ORG_SIZE_X	R/W	0h	Number of pixels of the video picture Encoded value (from 1 to 2048) to specify the number of pixels of the video picture in memory (program to value minus one). The size is limited to the size of the line buffer of the vertical sampling block in case the video picture is processed by the vertical filtering unit.

13.6.1.39 DISPC_VID1_ACCU_0 to DISPC_VID1_ACCU_1 Register (offset = E8h to ECh) [reset = 0h]

DISPC_VID1_ACCU_0 to DISPC_VID1_ACCU_1 is shown in Figure 13-115 and described in Table 13-80.

The register configures the resize accumulator init values for horizontal and vertical up-/down-sampling of video
window #n (#I for ping-pong mechanism with external trigger, based on the field polarity)
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-115. DISPC_VID1_ACCU_0 to DISPC_VID1_ACCU_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						VERTICAL_ACCU									
R/W-0h						R/W-0h									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						HORIZONTAL_ACCU									
R/W-0h						R/W-0h									

Table 13-80. DISPC_VID1_ACCU_0 to DISPC_VID1_ACCU_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R/W	0h	
25-16	VERTICAL_ACCU	R/W	0h	Vertical initialization accu value. Encoded value (from 0 to 1023).
15-10	RESERVED	R/W	0h	
9-0	HORIZONTAL_ACCU	R/W	0h	Horizontal initialization accu value. Encoded value (from 0 to 1023).

13.6.1.40 DISPC_VID1_FIR_COEF_H_0 Register (offset = F0h + [i * 8h]) [reset = 0h]

DISPC_VID1_FIR_COEF_H_0 is shown in Figure 13-116 and described in Table 13-81.

The bank of registers configure the up-/down-scaling coefficients for the vertical and horizontal resize of the video
picture associated with video window #n for the phases from 0 to 7.
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-116. DISPC_VID1_FIR_COEF_H_0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">FIR_HC3</td><td colspan="8">FIR_HC2</td><td colspan="8">FIR_HC1</td><td colspan="8">FIR_HC0</td></tr>
<tr><td colspan="8">R/W-0h</td><td colspan="8">R/W-0h</td><td colspan="8">R/W-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 13-81. DISPC_VID1_FIR_COEF_H_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	FIR_HC3	R/W	0h	Signed coefficient C3 for the horizontal up-/down-scaling with the phase n
23-16	FIR_HC2	R/W	0h	Unsigned coefficient C2 for the horizontal up-/down-scaling with the phase n
15-8	FIR_HC1	R/W	0h	Signed coefficient C1 for the horizontal up-/down-scaling with the phase n
7-0	FIR_HC0	R/W	0h	Signed coefficient C0 for the horizontal up-/down-scaling with the phase

13.6.1.41 DISPC_VID1_FIR_COEF_HV_0 Register (offset = F4h + [i * 8h]) [reset = 0h]

DISPC_VID1_FIR_COEF_HV_0 is shown in Figure 13-117 and described in Table 13-82.

The bank of registers configure the down/up-/down-scaling coefficients for the vertical and horizontal resize of the
video picture associated with video window #n for the phases from 0 to 7.
Shadow register, updated on VFP start period or EVSYNC

Figure 13-117. DISPC_VID1_FIR_COEF_HV_0 Register

31-24	23-16	15-8	7-0
FIR_VC2	FIR_VC1	FIR_VC0	FIR_HC4
R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 13-82. DISPC_VID1_FIR_COEF_HV_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	FIR_VC2	R/W	0h	Signed coefficient C2 for the vertical up-/down-scaling with the phase n
23-16	FIR_VC1	R/W	0h	Unsigned coefficient C1 for the vertical up-/down-scaling with the phase n
15-8	FIR_VC0	R/W	0h	Signed coefficient C0 for the vertical up-/down-scaling with the phase n
7-0	FIR_HC4	R/W	0h	Signed coefficient C4 for the horizontal up-/down-scaling with the phase n

13.6.1.42 DISPC_VID1_CONV_COEF0 Register (offset = 130h) [reset = 0h]

DISPC_VID1_CONV_COEF0 is shown in Figure 13-118 and described in Table 13-83.

The register configures the color space conversion matrix coefficients for video pipeline #n. Shadow register, updated on VFP start period or EVSYNC.

Figure 13-118. DISPC_VID1_CONV_COEF0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					RCR										
R/W-0h					R/W-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					RY										
R/W-0h					R/W-0h										

Table 13-83. DISPC_VID1_CONV_COEF0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	RCR	R/W	0h	RCr Coefficient Encoded signed value (from -1024 to 1023).
15-11	RESERVED	R/W	0h	
10-0	RY	R/W	0h	RY Coefficient Encoded signed value (from -1024 to 1023).

13.6.1.43 DISPC_VID1_CONV_COEF1 Register (offset = 134h) [reset = 0h]

DISPC_VID1_CONV_COEF1 is shown in Figure 13-119 and described in Table 13-84.

The register configures the color space conversion matrix coefficients for video pipeline #n.
Shadow register, updated on VFP start period or EVSYNC

Figure 13-119. DISPC_VID1_CONV_COEF1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					GY										
R/W-0h					R/W-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					RCB										
R/W-0h					R/W-0h										

Table 13-84. DISPC_VID1_CONV_COEF1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	GY	R/W	0h	GY Coefficient Encoded signed value (from -1024 to 1023).
15-11	RESERVED	R/W	0h	
10-0	RCB	R/W	0h	RCb Coefficient Encoded signed value (from -1024 to 1023)

13.6.1.44 DISPC_VID1_CONV_COEF2 Register (offset = 138h) [reset = 0h]

DISPC_VID1_CONV_COEF2 is shown in Figure 13-120 and described in Table 13-85.

The register configures the color space conversion matrix coefficients for video pipeline #n. Shadow register, updated on VFP start period or EVSYNC.

Figure 13-120. DISPC_VID1_CONV_COEF2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					GCB										
R/W-0h					R/W-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					GCR										
R/W-0h					R/W-0h										

Table 13-85. DISPC_VID1_CONV_COEF2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	GCB	R/W	0h	GCb Coefficient Encoded signed value (from -1024 to 1023).
15-11	RESERVED	R/W	0h	
10-0	GCR	R/W	0h	GCr Coefficient Encoded signed value (from -1024 to 1023).

13.6.1.45 DISPC_VID1_CONV_COEF3 Register (offset = 13Ch) [reset = 0h]

DISPC_VID1_CONV_COEF3 is shown in Figure 13-121 and described in Table 13-86.

The register configures the color space conversion matrix coefficients for video pipeline #n. Shadow register, updated on VFP start period or EVSYNC

Figure 13-121. DISPC_VID1_CONV_COEF3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					BCR										
R/W-0h					R/W-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					BY										
R/W-0h					R/W-0h										

Table 13-86. DISPC_VID1_CONV_COEF3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	BCR	R/W	0h	BCr coefficient Encoded signed value (from -1024 to 1023).
15-11	RESERVED	R/W	0h	
10-0	BY	R/W	0h	BY coefficient Encoded signed value (from -1024 to 1023).

13.6.1.46 DISPC_VID1_CONV_COEF4 Register (offset = 140h) [reset = 0h]

DISPC_VID1_CONV_COEF4 is shown in Figure 13-122 and described in Table 13-87.

The register configures the color space conversion matrix coefficients for video pipeline #n. Shadow register, updated on VFP start period or EVSYNC.

Figure 13-122. DISPC_VID1_CONV_COEF4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					BCB										
R/W-0h																					R/W-0h										

Table 13-87. DISPC_VID1_CONV_COEF4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-0	BCB	R/W	0h	BCb Coefficient Encoded signed value (from -1024 to 1023).

13.6.1.47 DISPC_VID2_BA_0 to DISPC_VID2_BA_1 Register (offset = 14Ch to 150h) [reset = 0h]

DISPC_VID2_BA_0 to DISPC_VID2_BA_1 is shown in Figure 13-123 and described in Table 13-88.

The register configures the base address of the video buffer for video window #n(#j for ping-pong mechanism with
external trigger, based on the field polarity: 0 for even field and 1 for odd field).
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-123. DISPC_VID2_BA_0 to DISPC_VID2_BA_1 Register

31-0
BA
R/W-0h

Table 13-88. DISPC_VID2_BA_0 to DISPC_VID2_BA_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BA	R/W	0h	Video base address Base address of the video buffer (aligned on pixel size boundary)

13.6.1.48 DISPC_VID2_POSITION Register (offset = 154h) [reset = 0h]

DISPC_VID2_POSITION is shown in Figure 13-124 and described in Table 13-89.

The register configures the position of video window #n.
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-124. DISPC_VID2_POSITION Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					POSY										
R/W-0h					R/W-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					POSX										
R/W-0h					R/W-0h										

Table 13-89. DISPC_VID2_POSITION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	POSY	R/W	0h	Y position of video window #n Encoded value (from 0 to 2047) to specify the Y position of video window #n. The line at the top has the Y-position 0.
15-11	RESERVED	R/W	0h	
10-0	POSX	R/W	0h	X position of video window #n Encoded value (from 0 to 2047) to specify the X position of video window #n. The first pixel on the left of the display screen has the X-position 0.

13.6.1.49 DISPC_VID2_SIZE Register (offset = 158h) [reset = 0h]

DISPC_VID2_SIZE is shown in Figure 13-125 and described in Table 13-90.

The register configures the size of video window #n.
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-125. DISPC_VID2_SIZE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					VID_SIZE_Y										
R/W-0h					R/W-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					VID_SIZE_X										
R/W-0h					R/W-0h										

Table 13-90. DISPC_VID2_SIZE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	VID_SIZE_Y	R/W	0h	Number of lines of video #n Encoded value (from 1 to 2048) to specify the number of lines of video window #n (program to value minus one).
15-11	RESERVED	R/W	0h	
10-0	VID_SIZE_X	R/W	0h	Number of pixels of video window #n Encoded value (from 1 to 2048) to specify the number of pixels of video window #n (program to value minus one).

13.6.1.50 DISPC_VID2_ATTRS Register (offset = 15Ch) [reset = 0h]

DISPC_VID2_ATTRS is shown in Figure 13-126 and described in Table 13-91.

Figure 13-126. DISPC_VID2_ATTRS Register

31	30	29	28	27	26	25	24
RESERVED			PRE_MULTIPLY_ALPHA	RESERVED			SELF_REFRESH
R/W-0h			R/W-0h	R/W-0h			R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ARBITRATION	LINE_BUFFER_SPLIT	VERTICAL_TAPS	OPTIMIZATION	FIFO_PRELOAD	ROW_REPEAT_EN	ENDIAN	CHANNEL_OUT
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BURST_SIZE		ROTATION		FULL_RANGE	REPLICATION_EN	COLOR_CONV_EN	VRESIZE_CONF
R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
HRESIZE_CONF	RESIZE_EN		FMT				EN
R/W-0h	R/W-0h		R/W-0h				R/W-0h

Table 13-91. DISPC_VID2_ATTRS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	PRE_MULTIPLY_ALPHA	R/W	0h	The field configures the DISPC VID2 to process incoming data as pre-multiplied alpha data or non pre-multiplied alpha data. Default setting is non pre-multiplied alpha data. 0h (R/W) = Non pre-multiplyalpha data color component 1h (R/W) = Premultiplyalpha data color component
27-25	RESERVED	R/W	0h	
24	SELF_REFRESH	R/W	0h	Enables the self refresh of the video window from its own FIFO only. 0h (R/W) = The video pipeline accesses the interconnect to fetch data from the system memory 1h (R/W) = The video pipeline does not need anymore to fetch data from memory. Only the video FIFO is used. It takes effect after the frame has been loaded in the FIFO
23	ARBITRATION	R/W	0h	Determines the priority of the video pipeline. The video pipeline is one of the high priority pipeline. The arbitration wheel gives always the priority first to the high priority pipelines using round-robin between them. When there is only normal priority pipelines sending requests, the round-robin applies between them. 0h (R/W) = The video pipeline is one of the normal priority pipeline 1h (R/W) = The video pipeline is one of the high priority pipeline
22	LINE_BUFFER_SPLIT	R/W	0h	Video vertical line buffer split 0h (R/W) = Vertical line buffers are not split 1h (R/W) = Vertical line buffers are split into two
21	VERTICAL_TAPS	R/W	0h	Video vertical resize tap number 0h (R/W) = Three taps are used for the vertical filtering logic. The other two taps are not used 1h (R/W) = Five taps are used for the vertical filtering logic

Table 13-91. DISPC_VID2_ATTRS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	OPTIMIZATION	R/W	0h	Video optimization in case of 0h (R/W) = The DMA engine fetches one pixel for each 32-bit OCP request (RGB16 and YUV422) while doing 90- and 270-degree rotation (accessing on-chip memory and off-chip memory). 1h (R/W) = The DMA engine fetches two pixels for each 32-bit OCP request (RGB16 and YUV422) while doing 90- and 270-degree rotation (accessing on-chip memory and off-chip memory). The bit field [21] VIDVERTICALTAPS shall be set to 0x1, bit field [22] VIDLINEBUFFERSPLIT to 0x1, and all scaler registers shall be configured even for 1:1 ratio. Even width is required for the input picture when 5 taps are used.
19	FIFO_PRELOAD	R/W	0h	Video preload value 0h (R/W) = H/W prefetches pixels up to the preload value defined in the preload register 1h (R/W) = H/W prefetches pixels up to the high threshold value.
18	ROW_REPEAT_EN	R/W	0h	Video Row Repeat (YUV case only when rotating 90 or 270-degree) 0h (R/W) = Row of VIDn won't be read twice 1h (R/W) = The Row data are fetched twice to extract both the Y components
17	ENDIAN	R/W	0h	Video Endianness 0h (R/W) = Little endian operation is selected 1h (R/W) = Big endian operation is selected.
16	CHANNEL_OUT	R/W	0h	Video Channel Out configuration 0h (R/W) = LCD output selected 1h (R/W) = 24 bit output selected
15-14	BURST_SIZE	R/W	0h	Video DMA Burst Size 0h (R/W) = 4x32bit bursts 1h (R/W) = 8x32bit bursts 2h (R/W) = 16x32bit bursts 3h (R/W) = 3
13-12	ROTATION	R/W	0h	Video Rotation Flag 0h (R/W) = No rotation or VidFormat is RGB 1h (R/W) = Rotation by 90 degrees 2h (R/W) = Rotation by 180 degrees 3h (R/W) = Rotation by 270 degrees
11	FULL_RANGE	R/W	0h	VidFullRange 0h (R/W) = Limited range selected: 16 subtracted from Y before color space conversion 1h (R/W) = Full range selected: Y is not modified before the color space conversion
10	REPLICATION_EN	R/W	0h	VidReplicationEnable 0h (R/W) = Disable Video replication logic 1h (R/W) = Enable Video replication logic
9	COLOR_CONV_EN	R/W	0h	VidColorConvEnable 0h (R/W) = Disable Color Space Conversion CbYCr to RGB 1h (R/W) = Enable Color Space Conversion CbYCr to RGB

Table 13-91. DISPC_VID2_ATTRS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	VRESIZE_CONF	R/W	0h	Video Vertical Resize Configuration 0h (R/W) = Up-sampling selected 1h (R/W) = Down-sampling selected
7	HRESIZE_CONF	R/W	0h	Video Horizontal Resize Configuration 0h (R/W) = Up-sampling selected 1h (R/W) = Down-sampling selected
6-5	RESIZE_EN	R/W	0h	Video Resize Enable 0h (R/W) = Disable the resize processing 1h (R/W) = Enable the horizontal resize processing 2h (R/W) = Enable the vertical resize processing 3h (R/W) = Enable both horizontal and vertical resize processing
4-1	FMT	R/W	0h	Video2 channel Format Other enums: Reserved (all other values: 0x0 and 0x3, 0x7, and 0xF) 0h (R/W) = RESERVED : 0 1h (R/W) = RESERVED : 1 2h (R/W) = RESERVED : 2 3h (R/W) = RESERVED : 3 4h (R/W) = RGB 12 (16-bit container) 5h (R/W) = ARGB 16 6h (R/W) = RGB 16 7h (R/W) = RESERVED : 7 8h (R/W) = RGB 24 (un-packed in 32-bit container) 9h (R/W) = RGB 24 (packed in 24-bit container) Ah (R/W) = YUV2 4:2:2 co-sited Bh (R/W) = UYVY 4:2:2 co-sited Ch (R/W) = ARGB 32 Dh (R/W) = RGBA 32 Eh (R/W) = RGBx 32 (24-bit RGB aligned on MSB of the 32-bit container) Fh (R/W) = RESERVED : 15
0	EN	R/W	0h	VidEnable 0h (R/W) = Video disabled (video pipeline inactive and window not present) 1h (R/W) = Video enabled (video pipeline active and window present on the screen)

13.6.1.51 DISPC_VID2_FIFO_THR Register (offset = 160h) [reset = 3C003FFh]

DISPC_VID2_FIFO_THR is shown in Figure 13-127 and described in Table 13-92.

The register configures the video FIFO associated with video pipeline #n.
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-127. DISPC_VID2_FIFO_THR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				FIFO_HIGH_THR											
R/W-0h				R/W-3C0h											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				FIFO_LOW_THR											
R/W-0h				R/W-3FFh											

Table 13-92. DISPC_VID2_FIFO_THR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-16	FIFO_HIGH_THR	R/W	3C0h	Video FIFO high threshold Number of bytes defining the threshold value
15-12	RESERVED	R/W	0h	
11-0	FIFO_LOW_THR	R/W	3FFh	Video FIFO low threshold Number of bytes defining the threshold value

13.6.1.52 DISPC_VID2_FIFO_SIZE_STS Register (offset = 164h) [reset = 400h]

DISPC_VID2_FIFO_SIZE_STS is shown in Figure 13-128 and described in Table 13-93.

The register defines the video FIFO size for video pipeline #n.

Figure 13-128. DISPC_VID2_FIFO_SIZE_STS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="21">RESERVED</td><td colspan="11">FIFO_SIZE</td></tr>
<tr><td colspan="21">R/W-0h</td><td colspan="11">R/W-400h</td></tr>
</table>

Table 13-93. DISPC_VID2_FIFO_SIZE_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-0	FIFO_SIZE	R/W	400h	Video FIFO Size Number of bytes defining the FIFO value

13.6.1.53 DISPC_VID2_ROW_INC Register (offset = 168h) [reset = 1h]

DISPC_VID2_ROW_INC is shown in Figure 13-129 and described in Table 13-94.

The register configures the number of bytes to increment at the end of the row for the buffer associated with video
window #n.
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-129. DISPC_VID2_ROW_INC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROW_INC																															
R/W-1h																															

Table 13-94. DISPC_VID2_ROW_INC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ROW_INC	R/W	1h	Number of bytes to increment at the end of the row Encoded signed value (from (-2^31)-1 to 2^31) to specify the number of bytes to increment at the end of the row in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value 1+n*BPP means increment of n pixels. The value 1- (n+1)*BPP means decrement of n pixels.

13.6.1.54 DISPC_VID2_PIXEL_INC Register (offset = 16Ch) [reset = 1h]

DISPC_VID2_PIXEL_INC is shown in Figure 13-130 and described in Table 13-95.

The register configures the number of bytes to increment between two pixels for the buffer associated with video
window #n.
Shadow register, updated on VFP start period or EVSYNC

Figure 13-130. DISPC_VID2_PIXEL_INC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PIXEL_INC															
R/W-0h																R/W-1h															

Table 13-95. DISPC_VID2_PIXEL_INC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	PIXEL_INC	R/W	1h	Number of bytes to increment at the end of the row Encoded signed value (from (-2^15)-1 to 2^15) to specify the number of bytes between two pixels in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value 1+n*BPP means increment of n pixels. The value 1-(n+1)*BPP means decrement of n pixels

13.6.1.55 DISPC_VID2_FIR Register (offset = 170h) [reset = 0h]

DISPC_VID2_FIR is shown in Figure 13-131 and described in Table 13-96.

The register configures the resize factors for horizontal and vertical up-/down-sampling of video window #n.
Shadow register, updated on VFP start period or EVSYNC

Figure 13-131. DISPC_VID2_FIR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="3">RESERVED</td><td colspan="13">FIR_V_INC</td></tr>
<tr><td colspan="3">R/W-0h</td><td colspan="13">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">RESERVED</td><td colspan="13">FIR_H_INC</td></tr>
<tr><td colspan="3">R/W-0h</td><td colspan="13">R/W-0h</td></tr>
</table>

Table 13-96. DISPC_VID2_FIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28-16	FIR_V_INC	R/W	0h	Vertical increment of the up-/down-sampling filter Encoded value (from 1 to 4096). The value 0 is invalid. Values greater than 4096 are invalid.
15-13	RESERVED	R/W	0h	
12-0	FIR_H_INC	R/W	0h	Horizontal increment of the up-/down-sampling filter Encoded value (from 1 to 4096). The value 0 is invalid. Values greater than 4096 are invalid.

13.6.1.56 DISPC_VID2_PICTURE_SIZE Register (offset = 174h) [reset = 0h]

DISPC_VID2_PICTURE_SIZE is shown in Figure 13-132 and described in Table 13-97.

The register configures the size of the video picture associated with video layer #nbefore up-/down-scaling.
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-132. DISPC_VID2_PICTURE_SIZE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					VID_ORG_SIZE_Y										
R/W-0h					R/W-0h										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					VID_ORG_SIZE_X										
R/W-0h					R/W-0h										

Table 13-97. DISPC_VID2_PICTURE_SIZE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	VID_ORG_SIZE_Y	R/W	0h	Number of lines of the video picture Encoded value (from 1 to 2048) to specify the number of lines of the video picture in memory (program to value minus one)
15-11	RESERVED	R/W	0h	
10-0	VID_ORG_SIZE_X	R/W	0h	Number of pixels of the video picture Encoded value (from 1 to 2048) to specify the number of pixels of the video picture in memory (program to value minus one). The size is limited to the size of the line buffer of the vertical sampling block in case the video picture is processed by the vertical filtering unit.

13.6.1.57 DISPC_VID2_ACCU_0 to DISPC_VID2_ACCU_1 Register (offset = 178h to 17Ch) [reset = 0h]

DISPC_VID2_ACCU_0 to DISPC_VID2_ACCU_1 is shown in Figure 13-133 and described in Table 13-98.

The register configures the resize accumulator init values for horizontal and vertical up-/down-sampling of video
window #n (#I for ping-pong mechanism with external trigger, based on the field polarity)
Shadow register, updated on VFP start period or EVSYNC.

Figure 13-133. DISPC_VID2_ACCU_0 to DISPC_VID2_ACCU_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						VERTICAL_ACCU									
R/W-0h						R/W-0h									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						HORIZONTAL_ACCU									
R/W-0h						R/W-0h									

Table 13-98. DISPC_VID2_ACCU_0 to DISPC_VID2_ACCU_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R/W	0h	
25-16	VERTICAL_ACCU	R/W	0h	Vertical initialization accu value. Encoded value (from 0 to 1023).
15-10	RESERVED	R/W	0h	
9-0	HORIZONTAL_ACCU	R/W	0h	Horizontal initialization accu value. Encoded value (from 0 to 1023).

13.6.1.58 DISPC_VID2_FIR_COEF_H_0 Register (offset = 180h + [i * 8h]) [reset = 0h]

DISPC_VID2_FIR_COEF_H_0 is shown in Figure 13-134 and described in Table 13-99.

The bank of registers configure the up-/down-scaling coefficients for the vertical and horizontal resize of the video
picture associated with video window #n for the phases from 0 to 7.
Shadow register, updated on VFP start period or EVSYNC

Figure 13-134. DISPC_VID2_FIR_COEF_H_0 Register

31-24	23-16	15-8	7-0
FIR_HC3	FIR_HC2	FIR_HC1	FIR_HC0
R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 13-99. DISPC_VID2_FIR_COEF_H_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	FIR_HC3	R/W	0h	Signed coefficient C3 for the horizontal up-/down-scaling with the phase n
23-16	FIR_HC2	R/W	0h	Unsigned coefficient C2 for the horizontal up-/down-scaling with the phase n
15-8	FIR_HC1	R/W	0h	Signed coefficient C1 for the horizontal up-/down-scaling with the phase n
7-0	FIR_HC0	R/W	0h	Signed coefficient C0 for the horizontal up-/down-scaling with the phase n

13.6.1.59 DISPC_VID2_FIR_COEF_HV_0 Register (offset = 184h + [i * 8h]) [reset = 0h]

DISPC_VID2_FIR_COEF_HV_0 is shown in Figure 13-135 and described in Table 13-100.

The bank of registers configure the down/up-/down-scaling coefficients for the vertical and horizontal resize of the
video picture associated with video window #n for the phases from 0 to 7.
Shadow register, updated on VFP start period or EVSYNC

Figure 13-135. DISPC_VID2_FIR_COEF_HV_0 Register

31-24	23-16	15-8	7-0
FIR_VC2	FIR_VC1	FIR_VC0	FIR_HC4
R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 13-100. DISPC_VID2_FIR_COEF_HV_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	FIR_VC2	R/W	0h	Signed coefficient C2 for the vertical up-/down-scaling with the phase n
23-16	FIR_VC1	R/W	0h	Unsigned coefficient C1 for the vertical up-/down-scaling with the phase n
15-8	FIR_VC0	R/W	0h	Signed coefficient C0 for the vertical up-/down-scaling with the phase n
7-0	FIR_HC4	R/W	0h	Signed coefficient C4 for the horizontal up-/down-scaling with the phase n

13.6.1.60 DISPC_VID2_CONV_COEF0 Register (offset = 1C0h) [reset = 0h]

DISPC_VID2_CONV_COEF0 is shown in Figure 13-136 and described in Table 13-101.

The register configures the color space conversion matrix coefficients for video pipeline #n. Shadow register, updated on VFP start period or EVSYNC.

Figure 13-136. DISPC_VID2_CONV_COEF0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					RCR										
R/W-0h					R/W-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					RY										
R/W-0h					R/W-0h										

Table 13-101. DISPC_VID2_CONV_COEF0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	RCR	R/W	0h	RCr Coefficient Encoded signed value (from -1024 to 1023).
15-11	RESERVED	R/W	0h	
10-0	RY	R/W	0h	RY Coefficient Encoded signed value (from -1024 to 1023).

13.6.1.61 DISPC_VID2_CONV_COEF1 Register (offset = 1C4h) [reset = 0h]

DISPC_VID2_CONV_COEF1 is shown in Figure 13-137 and described in Table 13-102.

The register configures the color space conversion matrix coefficients for video pipeline #n.
Shadow register, updated on VFP start period or EVSYNC

Figure 13-137. DISPC_VID2_CONV_COEF1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					GY										
R/W-0h					R/W-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					RCB										
R/W-0h					R/W-0h										

Table 13-102. DISPC_VID2_CONV_COEF1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	GY	R/W	0h	GY Coefficient Encoded signed value (from -1024 to 1023).
15-11	RESERVED	R/W	0h	
10-0	RCB	R/W	0h	RCb Coefficient Encoded signed value (from -1024 to 1023)

13.6.1.62 DISPC_VID2_CONV_COEF2 Register (offset = 1C8h) [reset = 0h]

DISPC_VID2_CONV_COEF2 is shown in Figure 13-138 and described in Table 13-103.

The register configures the color space conversion matrix coefficients for video pipeline #n. Shadow register, updated on VFP start period or EVSYNC.

Figure 13-138. DISPC_VID2_CONV_COEF2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					GCB										
R/W-0h					R/W-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					GCR										
R/W-0h					R/W-0h										

Table 13-103. DISPC_VID2_CONV_COEF2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	GCB	R/W	0h	GCb Coefficient Encoded signed value (from -1024 to 1023).
15-11	RESERVED	R/W	0h	
10-0	GCR	R/W	0h	GCr Coefficient Encoded signed value (from -1024 to 1023).

13.6.1.63 DISPC_VID2_CONV_COEF3 Register (offset = 1CCh) [reset = 0h]

DISPC_VID2_CONV_COEF3 is shown in Figure 13-139 and described in Table 13-104.

The register configures the color space conversion matrix coefficients for video pipeline #n.
Shadow register, updated on VFP start period or EVSYNC

Figure 13-139. DISPC_VID2_CONV_COEF3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					BCR										
R/W-0h					R/W-0h										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					BY										
R/W-0h					R/W-0h										

Table 13-104. DISPC_VID2_CONV_COEF3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	BCR	R/W	0h	BCr coefficient Encoded signed value (from -1024 to 1023).
15-11	RESERVED	R/W	0h	
10-0	BY	R/W	0h	BY coefficient Encoded signed value (from -1024 to 1023).

13.6.1.64 DISPC_VID2_CONV_COEF4 Register (offset = 1D0h) [reset = 0h]

DISPC_VID2_CONV_COEF4 is shown in Figure 13-140 and described in Table 13-105.

The register configures the color space conversion matrix coefficients for video pipeline #n. Shadow register, updated on VFP start period or EVSYNC.

Figure 13-140. DISPC_VID2_CONV_COEF4 Register

31-11	10-0
RESERVED	BCB
R/W-0h	R/W-0h

Table 13-105. DISPC_VID2_CONV_COEF4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-0	BCB	R/W	0h	BCb Coefficient Encoded signed value (from -1024 to 1023).

13.6.1.65 DISPC_DATA_CYCLE_0 Register (offset = 1D4h + [i * 4h]) [reset = 0h]

DISPC_DATA_CYCLE_0 is shown in Figure 13-141 and described in Table 13-106.

The control register configures the output data format for ith (1st, 2nd or 3rd) cycle. Shadow register, updated on VFP start period.

Figure 13-141. DISPC_DATA_CYCLE_0 Register

31	30	29	28	27	26	25	24
RESERVED				BIT_ALIGNMENT_PIXEL2			
R/W-0h				R/W-0h			
23	22	21	20	19	18	17	16
RESERVED			NB_BITS_PIXEL2				
R/W-0h			R/W-0h				
15	14	13	12	11	10	9	8
RESERVED				BIT_ALIGNMENT_PIXEL1			
R/W-0h				R/W-0h			
7	6	5	4	3	2	1	0
RESERVED			NB_BITS_PIXEL1				
R/W-0h			R/W-0h				

Table 13-106. DISPC_DATA_CYCLE_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	BIT_ALIGNMENT_PIXEL2	R/W	0h	Bit alignment PIXEL2 Alignment of the bits from pixel#2 on the output interface
23-21	RESERVED	R/W	0h	
20-16	NB_BITS_PIXEL2	R/W	0h	Number of bits Number of bits from the pixel #2 (value from 0 to 16 bits). The values from 17 to 31 are invalid.
15-12	RESERVED	R/W	0h	
11-8	BIT_ALIGNMENT_PIXEL1	R/W	0h	Bit alignment Alignment of the bits from pixel#1 on the output interface
7-5	RESERVED	R/W	0h	
4-0	NB_BITS_PIXEL1	R/W	0h	Number of bits Number of bits from the pixel #1 (value from 0 to 16 bits). The values from 17 to 31 are invalid.

13.6.1.66 DISPC_VID1_FIR_COEF_V_0 to DISPC_VID1_FIR_COEF_V_7 Register (offset = 1E0h to 1FCh) [reset = 0h]

DISPC_VID1_FIR_COEF_V_0 to DISPC_VID1_FIR_COEF_V_7 is shown in Figure 13-142 and described in Table 13-107.

This bank of registers configures the down/up/down-scaling coefficients for the vertical resize of the video picture
associated with video window #n for phases 0 to 7. Shadow register, updated on VFP start period or EVSYNC.

Figure 13-142. DISPC_VID1_FIR_COEF_V_0 to DISPC_VID1_FIR_COEF_V_7 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="8">FIR_VC22</td><td colspan="8">FIR_VC00</td></tr>
<tr><td colspan="16">R/W-0h</td><td colspan="8">R/W-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 13-107. DISPC_VID1_FIR_COEF_V_0 to DISPC_VID1_FIR_COEF_V_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-8	FIR_VC22	R/W	0h	Signed coefficient C22 for vertical up/down-scaling with phase n
7-0	FIR_VC00	R/W	0h	Signed coefficient C00 for vertical up/down-scaling with phase n

13.6.1.67 DISPC_VID2_FIR_COEF_V_0 to DISPC_VID2_FIR_COEF_V_7 Register (offset = 200h to 21Ch) [reset = 0h]

DISPC_VID2_FIR_COEF_V_0 to DISPC_VID2_FIR_COEF_V_7 is shown in Figure 13-143 and described in Table 13-108.

This bank of registers configures the down/up/down-scaling coefficients for the vertical resize of the video picture
associated with video window #n for phases 0 to 7. Shadow register, updated on VFP start period or EVSYNC.

Figure 13-143. DISPC_VID2_FIR_COEF_V_0 to DISPC_VID2_FIR_COEF_V_7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FIR_VC22								FIR_VC00							
R/W-0h																R/W-0h								R/W-0h							

Table 13-108. DISPC_VID2_FIR_COEF_V_0 to DISPC_VID2_FIR_COEF_V_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-8	FIR_VC22	R/W	0h	Signed coefficient C22 for vertical up/down-scaling with phase n
7-0	FIR_VC00	R/W	0h	Signed coefficient C00 for vertical up/down-scaling with phase n

13.6.1.68 DISPC_CPR_COEF_R Register (offset = 220h) [reset = 0h]

DISPC_CPR_COEF_R is shown in Figure 13-144 and described in Table 13-109.

This register configures the color phase rotation matrix coefficients for the red component. Shadow register,
updated on VFP start period.

Figure 13-144. DISPC_CPR_COEF_R Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RR</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="2">RR</td><td>RESERVED</td><td colspan="5">RG</td></tr>
<tr><td colspan="2">R/W-0h</td><td>R/W-0h</td><td colspan="5">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="5">RG</td><td>RESERVED</td><td colspan="2">RB</td></tr>
<tr><td colspan="5">R/W-0h</td><td>R/W-0h</td><td colspan="2">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RB</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 13-109. DISPC_CPR_COEF_R Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RR	R/W	0h	RR coefficient Encoded signed value (from -512 to 511)
21	RESERVED	R/W	0h	
20-11	RG	R/W	0h	RG coefficient Encoded signed value (from -512 to 511)
10	RESERVED	R/W	0h	
9-0	RB	R/W	0h	RB coefficient Encoded signed value (from -512 to 511)

13.6.1.69 DISPC_CPR_COEF_G Register (offset = 224h) [reset = 0h]

DISPC_CPR_COEF_G is shown in Figure 13-145 and described in Table 13-110.

This register configures the color phase rotation matrix coefficients for the green component. Shadow register,
updated on VFP start period

Figure 13-145. DISPC_CPR_COEF_G Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">GR</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="2">GR</td><td>RESERVED</td><td colspan="5">GG</td></tr>
<tr><td colspan="2">R/W-0h</td><td>R/W-0h</td><td colspan="5">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="5">GG</td><td>RESERVED</td><td colspan="2">GB</td></tr>
<tr><td colspan="5">R/W-0h</td><td>R/W-0h</td><td colspan="2">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">GB</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 13-110. DISPC_CPR_COEF_G Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	GR	R/W	0h	GR coefficient Encoded signed value (from -512 to 511)
21	RESERVED	R/W	0h	
20-11	GG	R/W	0h	GG coefficient Encoded signed value (from -512 to 511)
10	RESERVED	R/W	0h	
9-0	GB	R/W	0h	GB coefficient Encoded signed value (from -512 to 511)

13.6.1.70 DISPC_CPR_COEF_B Register (offset = 228h) [reset = 0h]

DISPC_CPR_COEF_B is shown in Figure 13-146 and described in Table 13-111.

This register configures the color phase rotation matrix coefficients for the blue component. Shadow register,
updated on VFP start period.

Figure 13-146. DISPC_CPR_COEF_B Register

31	30	29	28	27	26	25	24
BR							
R/W-0h							
23	22	21	20	19	18	17	16
BR		RESERVED	BG				
R/W-0h		R/W-0h	R/W-0h				
15	14	13	12	11	10	9	8
BG					RESERVED	BB	
R/W-0h					R/W-0h	R/W-0h	
7	6	5	4	3	2	1	0
BB							
R/W-0h							

Table 13-111. DISPC_CPR_COEF_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	BR	R/W	0h	BR coefficient Encoded signed value (from -512 to 511)
21	RESERVED	R/W	0h	
20-11	BG	R/W	0h	BG coefficient Encoded signed value (from -512 to 511)
10	RESERVED	R/W	0h	
9-0	BB	R/W	0h	BB coefficient Encoded signed value (from -512 to 511)

13.6.1.71 DISPC_GFX_PRELOAD Register (offset = 22Ch) [reset = 100h]

DISPC_GFX_PRELOAD is shown in Figure 13-147 and described in Table 13-112.

This register configures the graphics FIFO. Shadow register, updated on VFP start period or EVSYNC.

Figure 13-147. DISPC_GFX_PRELOAD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				PRELOAD											
R/W-0h																				R/W-100h											

Table 13-112. DISPC_GFX_PRELOAD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	PRELOAD	R/W	100h	Graphics preload value: Number of bytes defining the preload value. Constraint: Maximum value is (FIFO size - DMA burst size - 8) bytes

13.6.1.72 DISPC_VID1_PRELOAD Register (offset = 230h) [reset = 100h]

DISPC_VID1_PRELOAD is shown in Figure 13-148 and described in Table 13-113.

This register configures the video FIFO. Shadow register, updated on VFP start period or EVSYNC.

Figure 13-148. DISPC_VID1_PRELOAD Register

31-12	11-0
RESERVED	PRELOAD
R/W-0h	R/W-100h

Table 13-113. DISPC_VID1_PRELOAD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	PRELOAD	R/W	100h	Video preload value: Number of bytes defining the preload value. Constraint: Maximum value is (FIFO size - DMA burst size - 8) bytes

13.6.1.73 DISPC_VID2_PRELOAD Register (offset = 234h) [reset = 100h]

DISPC_VID2_PRELOAD is shown in Figure 13-149 and described in Table 13-114.

This register configures the video FIFO. Shadow register, updated on VFP start period or EVSYNC.

Figure 13-149. DISPC_VID2_PRELOAD Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="20">RESERVED</td><td colspan="12">PRELOAD</td></tr>
<tr><td colspan="20">R/W-0h</td><td colspan="12">R/W-100h</td></tr>
</table>

Table 13-114. DISPC_VID2_PRELOAD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	PRELOAD	R/W	100h	Video preload value: Number of bytes defining the preload value. Constraint: Maximum value is (FIFO size - DMA burst size - 8) bytes

13.6.2 DSS_TOP Registers

Table 13-115 lists the memory-mapped registers for the DSS_TOP. All register offset addresses not listed in Table 13-115 should be considered as reserved locations and the register contents should not be modified.

Table 13-115. DSS_TOP Registers

Offset	Acronym	Register Name	Section
0h	DSS_REVISIONNUMBER		Section 13.6.2.1
10h	DSS_SYSCONFIG		Section 13.6.2.2
14h	DSS_SYSSTS		Section 13.6.2.3
18h	DSS_IRQSTS		Section 13.6.2.4
40h	DSS_CTRL		Section 13.6.2.5
5Ch	DSS_CLK_STS		Section 13.6.2.6

13.6.2.1 DSS_REVISIONNUMBER Register (offset = 0h) [reset = 00000040h]

DSS_REVISIONNUMBER is shown in Figure 13-150 and described in Table 13-116.

This register contains the DisplaySubSystem revision number

Figure 13-150. DSS_REVISIONNUMBER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">RESERVED</td><td colspan="8">REV</td></tr>
<tr><td colspan="24">R-0h</td><td colspan="8">R-40h</td></tr>
</table>

Table 13-116. DSS_REVISIONNUMBER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reads returns 0
7-0	REV	R	40h	[7:4] Major Revision [3:0] Minor revision

13.6.2.2 DSS_SYSCONFIG Register (offset = 10h) [reset = 1h]

DSS_SYSCONFIG is shown in Figure 13-151 and described in Table 13-117.

Figure 13-151. DSS_SYSCONFIG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">RESERVED</td><td colspan="2">RESERVED</td><td>RESERVED</td><td>SOFTRESET</td><td>AUTOIDLE</td></tr>
<tr><td colspan="3">R/W-0h</td><td colspan="2">R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-1h</td></tr>
</table>

Table 13-117. DSS_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4-3	RESERVED	R/W	0h	
2	RESERVED	R/W	0h	
1	SOFTRESET	R/W	0h	Software Reset. Set this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads, it alwatys returns 0. 0h (R/W) = Normal Mode. 1h (R/W) = The module is reset.
0	AUTOIDLE	R/W	1h	Enable Power management capability 0h (R/W) = OCP clock is free running. 1h (R/W) = Automatic OCP clock gating strategy is applied on the OCP interface activity.

13.6.2.3 DSS_SYSSTS Register (offset = 14h) [reset = 00000001h]

DSS_SYSSTS is shown in Figure 13-152 and described in Table 13-118.

Figure 13-152. DSS_SYSSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							RESETDONE
R-0h							R-1h

Table 13-118. DSS_SYSSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	RESETDONE	R	1h	Internal Reset monitoring. 0h (R) = Internal module reset is ongoing. 1h (R) = Reset Completed

13.6.2.4 DSS_IRQSTS Register (offset = 18h) [reset = 0h]

DSS_IRQSTS is shown in Figure 13-153 and described in Table 13-119.

This register indicates the source of the interrupt and the status of the interrupt line.

Figure 13-153. DSS_IRQSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	DISPC_IRQ
R-0h						R-0h	R-0h

Table 13-119. DSS_IRQSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reads return 0s.
1	RESERVED	R	0h	Reserved
0	DISPC_IRQ	R	0h	DISPC interrupt status 0h (R) = Interrupt inactive 1h (R) = Interrupt active

13.6.2.5 DSS_CTRL Register (offset = 40h) [reset = 0h]

DSS_CTRL is shown in Figure 13-154 and described in Table 13-120.

Figure 13-154. DSS_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED	RFBI_SWITCH	RESERVED	LCD2_CLK_SWITCH	RESERVED	RESERVED	FCK_CLK_SWITCH	
R-0h	R/W-0h	R-0h	R/W-0h	R-0h	R-0h	R/W-0h	
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	LCD1_CLK_SWITCH
R/W-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R/W-0h

Table 13-120. DSS_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	Write 0's for suture compatibility.
15	RESERVED	R	0h	Reserved
14	RFBI_SWITCH	R/W	0h	Selects the Video port from DISPC between Video port #1 and Video port #2 (mux #10) 0h (R/W) = Video Port #1 (also named primary LCD output or LCD1) is selected (backward compatibility mode) 1h (R/W) = Video Port #2 (also named secondary LCD output or LCD2) is selected.
13	RESERVED	R	0h	Reserved
12	LCD2_CLK_SWITCH	R/W	0h	DSS_CLK/PLL2_CLK1 clock switch (mux #3). Selects the clock source for the DISPC LCD2_CLK clock. 0h (R/W) = DSS_CLK selected (from PRCM) 1h (R/W) = PLL2_CLK1 selected.
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9-8	FCK_CLK_SWITCH	R/W	0h	Selects the clock source for the DISPC functional clock. 0h (R/W) = DSS_CLK selected (from PRCM) 1h (R/W) = PLL1_CLK1 selected (from DSI1_PLL) 2h (R/W) = PLL2_CLK1 selected (from DSI2 PLL) 3h (R/W) = PLL3_CLK1 selected (from HDMI PLL)
7	RESERVED	R/W	0h	Write 0's for suture compatibility.
6	RESERVED	R	0h	Reserved
5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	RESERVED	R	0h	Reserved
2	RESERVED	R	0h	Reserved
1	RESERVED	R	0h	Reserved
0	LCD1_CLK_SWITCH	R/W	0h	DSS_CLK/PLL1_CLK1 clock switch (mux #2). Selects the clock source for the DISPC LCD1_CLK clock. 0h (R/W) = DSS_CLK selected (from PRCM) 1h (R/W) = PLL1_CLK1 selected (from DSI1_PLL)

13.6.2.6 DSS_CLK_STS Register (offset = 5Ch) [reset = 0000AA81h]

DSS_CLK_STS is shown in Figure 13-155 and described in Table 13-121.

Figure 13-155. DSS_CLK_STS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED		RFBI_STS	RESERVED		FCK_CLK_STS		
R-0h		R-0h	R-0h		R-1h		
15	14	13	12	11	10	9	8
FCK_CLK_STS	RESERVED		LCD2_CLK_STS		RESERVED		RESERVED
R-1h	R-0h		R-1h		R-0h		R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED		RESERVED			LCD1_CLK_STS	
R-0h	R-0h		R-0h			R-1h	

Table 13-121. DSS_CLK_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	Read returns 0.
21	RFBI_STS	R	0h	Video port selection status (mux #11) Indiactes if video port #1 or video port #2 from DISPC is used to provide data to the RFBI. 0h (R) = Video Port #1 (named also as primary LCD output or LCD1) used to provide data to RFBI 1h (R) = Video Port #2 (named also secondary LCD output or LCD2) used to provide data to RFBI
20-19	RESERVED	R	0h	Reserved
18-15	FCK_CLK_STS	R	1h	FCK_CLK clock selection status (mux #1) indicates which clock is used by the glitch free mux selecting the source of FCK_CLK. It is required to have the current clock and the new selected clock being running in order to be able to switch. Both clocks are used at the same time while the switch is on going. 0h (R) = DSS_CLK clock switch is ongoing 1h (R) = DSS_CLK is used by DISPC as FCK_CLK clock 2h (R) = PLL1_CLK1 is used by DISPC as FCK_CLK clock 4h (R) = PLL2_CLK1 is used by DISPC as FCK_CLK clock 8h (R) = PLL3_CLK1 (TV_CLK) is used by DISPC as FCK_CLK clock
14-13	RESERVED	R	0h	Reserved
12-11	LCD2_CLK_STS	R	1h	LCD2_CLK clock selection status (mux #3) indicates which clock is used by the glitch free mux selecting the source of LCD2_CLK. It is required for the current clock and the new selected clock being running in order to be able to switch. Both clocks are are used at the same time while the switch is on going. 0h (R) = LCD2_CLK clock switch is ongoing 1h (R) = DSS_CLK is used by DSI1 as LCD2_CLK clock 2h (R) = PLL2_CLK2 is used by DISPC as LCD2_CLK clock
10-9	RESERVED	R	0h	Reserved
8-7	RESERVED	R	0h	Reserved
6-5	RESERVED	R	0h	Reserved
4-2	RESERVED	R	0h	Read returns 0.

Table 13-121. DSS_CLK_STS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	LCD1_CLK_STS	R	1h	LCD1_CLK clock selection status (mux #2) indicates which clock is used by the glitch free mux selecting the source of LCD1_CLK. It is required for the current clock and the new selected clock being running in order to be able to switch. Both clocks are are used at the same time while the switch is on going. 0h (R) = LCD1_CLK clock switching is ongoing 1h (R) = DSS_CLK is used as LCD1_CLK 2h (R) = PLI1_CLK1 is used by DISPC as LCD1_CLK

13.6.3 DSS_RFBI Registers

Table 13-122 lists the memory-mapped registers for the DSS_RFBI. All register offset addresses not listed in Table 13-122 should be considered as reserved locations and the register contents should not be modified.

Table 13-122. DSS_RFBI REGISTERS

Offset	Acronym	Register Name	Section
0h	RFBI_REVISION		Section 13.6.3.1
10h	RFBI_SYSCONFIG		Section 13.6.3.2
14h	RFBI_SYSSTS		Section 13.6.3.3
40h	RFBI_CTRL		Section 13.6.3.4
44h	RFBI_PIXEL_CNT		Section 13.6.3.5
48h	RFBI_LINE_NUMBER		Section 13.6.3.6
4Ch	RFBI_CMD		Section 13.6.3.7
50h	RFBI_PARAM		Section 13.6.3.8
54h	RFBI_DATA		Section 13.6.3.9
58h	RFBI_READ		Section 13.6.3.10
5Ch	RFBI_STS		Section 13.6.3.11
60h	RFBI_CONFIG_0		Section 13.6.3.12
64h	RFBI_ONOFF_TIME_0		Section 13.6.3.13
68h	RFBI_CYCLE_TIME_0		Section 13.6.3.14
6Ch	RFBI_DATA_CYCLE1_0		Section 13.6.3.15
70h	RFBI_DATA_CYCLE2_0		Section 13.6.3.16
74h	RFBI_DATA_CYCLE3_0		Section 13.6.3.17
78h	RFBI_CONFIG_1		Section 13.6.3.12
7Ch	RFBI_ONOFF_TIME_1		Section 13.6.3.13
80h	RFBI_CYCLE_TIME_1		Section 13.6.3.14
84h	RFBI_DATA_CYCLE1_1		Section 13.6.3.15
88h	RFBI_DATA_CYCLE2_1		Section 13.6.3.16
8Ch	RFBI_DATA_CYCLE3_1		Section 13.6.3.17
90h	RFBI_VSYNC_WIDTH		Section 13.6.3.18
94h	RFBI_HSYNC_WIDTH		Section 13.6.3.19

13.6.3.1 RFBI_REVISION Register (offset = 0h) [reset = 0h]

RFBI_REVISION is shown in Figure 13-156 and described in Table 13-123.

This Register contains the IP revision code

Figure 13-156. RFBI_REVISION Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								REV							
R-0h																								R-0h							

Table 13-123. RFBI_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	REV	R	0h	IP Revision

13.6.3.2 RFBI_SYSCONFIG Register (offset = 10h) [reset = 1h]

RFBI_SYSCONFIG is shown in Figure 13-157 and described in Table 13-124.

This Register allows control of various parameters of the Interconnect Interface

Figure 13-157. RFBI_SYSCONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							

23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED			SIDLE_MODE		RESERVED	SOFT_RESET	AUTO_IDLE
R/W-0h			R/W-0h		R/W-0h	R/W-0h	R/W-1h

Table 13-124. RFBI_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4-3	SIDLE_MODE	R/W	0h	Slave interface power management, Idle req/ack control 0h (R/W) = Idle request is acknowledged unconditionally 1h (R/W) = An idle request is never acknowledged 2h (R/W) = Idle request is acknowledged based on the internal activity of the module 3h (R/W) = 3
2	RESERVED	R/W	0h	
1	SOFT_RESET	R/W	0h	Software reset Sets this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads, it always returns 0 0h (R/W) = Normal mode 1h (R/W) = The module is reset
0	AUTO_IDLE	R/W	1h	Internal clock gating strategy (interconnectL4 and display controller clock) 0h (R/W) = Interconnect L4 clock and display controller clock are free-running. 1h (R/W) = Automatic clock gating strategy is applied for the interconnect L4 clock and display controller clock, based on the interconnect interface and internal activity

13.6.3.3 RFBI_SYSSTS Register (offset = 14h) [reset = 1h]

RFBI_SYSSTS is shown in Figure 13-158 and described in Table 13-125.

This register provides status information about the module, excluding the interrupt status information.

Figure 13-158. RFBI_SYSSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED						BUSY_RFBI_DATA	BUSY
R-0h						R-0h	R-0h

7	6	5	4	3	2	1	0
RESERVED							RESET_DONE
R-0h							R-1h

Table 13-125. RFBI_SYSSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9	BUSY_RFBI_DATA	R	0h	Data are pending to be processed from interconnect FIFO. 0h (R/W) = No Data Pending 1h (R/W) = Data Pending
8	BUSY	R	0h	L4 Interface busy status bit 0h (R/W) = The access to the following register is not stalled: RFBI_CMD, RFBI_DATA, RFBI_STS, RFBI_PARAM, RFBI_READ 1h (R/W) = The access to any of the following registers is stalled: RFBI_CMD, RFBI_DATA, RFBI_STS, RFBI_PARAM, RFBI_READ.
7-1	RESERVED	R	0h	
0	RESET_DONE	R	1h	Internal reset monitoring 0h (R/W) = Internal module reset is on-going 1h (R/W) = Reset Completed

13.6.3.4 RFBI_CTRL Register (offset = 40h) [reset = 2h]

RFBI_CTRL is shown in Figure 13-159 and described in Table 13-126.

The control register allows configuration of the RFBI module

Figure 13-159. RFBI_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							

23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							

15	14	13	12	11	10	9	8
RESERVED							SMART_DMA_REQ
R/W-0h							R/W-0h

7	6	5	4	3	2	1	0
DISABLE_DMA_REQ	HIGH_THR		ITE	CONFIG_SELECT		BYPASS_MODE	EN
R/W-0h	R/W-0h		R/W-0h	R/W-0h		R/W-1h	R/W-0h

Table 13-126. RFBI_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	
8	SMART_DMA_REQ	R/W	0h	Smart DMA request 0h (R/W) = The dmareq is asserted and de-asserted depending on the interconnect FIFO space even if MIdlereq is high in smart idle/no-idle mode and the entire burst gets error responses from the module. 1h (R/W) = The dmareq is de-asserted after 2 clk cycles if it has been asserted for more than or equal to 2 clk cycles and MIdlereq is high in smart idle or no idle mode. No more burst requests will be given even if the space is available in the interconnect FIFO.
7	DISABLE_DMA_REQ	R/W	0h	Disable DMA request 0h (R/W) = The dmareq is enabled and the signal is generated based on the space available and the request coming into the data register. 1h (R/W) = The dmareq is disabled and the signal is not generated at all based on space in the interconnect FIFO. It stays high until the DISABLE DMAREQ is high even if there is space in the interconnect FIFO to take requests.
6-5	HIGH_THR	R/W	0h	Defines the interconnect FIFO high threshold used by HW to assert DMA request. Used only if data written to RFBI_DATA are sent using system DMA 0h (R/W) = Size of the transfer of 4 words of 32-bit wide 1h (R/W) = Size of the transfer of 8 words of 32-bit wide 2h (R/W) = Size of the transfer of 16 words of 32-bit wide
4	ITE	R/W	0h	Internal Trigger 0h (R/W) = H/W waits for ITE bit to be set if in internal trigger mode for the configuration in use 1h (R/W) = User sets the ITE bit to start the transfer, when H/W takes into account the bit, the H/W resets it

Table 13-126. RFBI_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	CONFIG_SELECT	R/W	0h	Select the CS and configuration 0h (R/W) = No CS selected 1h (R/W) = CS0 selected and Configuration #0 2h (R/W) = CS1 selected and configuration #1 3h (R/W) = CS0 and CS1 both selected (only the configuration for CS0 isused)
1	BYPASS_MODE	R/W	1h	Bypass Mode 0h (R/W) = The bypass mode is not selected 1h (R/W) = The bypass mode is selected
0	EN	R/W	0h	Enable/Disable flag 0h (R/W) = Disable RFBI 1h (R/W) = Enable RFBI

13.6.3.5 RFBI_PIXEL_CNT Register (offset = 44h) [reset = 0h]

RFBI_PIXEL_CNT is shown in Figure 13-160 and described in Table 13-127.

The control register configures the RFBI pixel count value.

Figure 13-160. RFBI_PIXEL_CNT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">PIXEL_CNT</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 13-127. RFBI_PIXEL_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PIXEL_CNT	R/W	0h	Pixel counter value The S/W indicates the number of pixels to transfer to the LCD panel frame buffer. The value is set when the module is disabled. During the transfer the HW decrements the register when a pixel has been sent to the RFB.

13.6.3.6 RFBI_LINE_NUMBER Register (offset = 48h) [reset = 0h]

RFBI_LINE_NUMBER is shown in Figure 13-161 and described in Table 13-128.

The control register configures the number of lines to synchronize the beginning of the transfer.

Figure 13-161. RFBI_LINE_NUMBER Register

31-11	10-0
RESERVED	LINE_NUMBER
R/W-0h	R/W-0h

Table 13-128. RFBI_LINE_NUMBER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-0	LINE_NUMBER	R/W	0h	Programmable line number Line number from 0 to 2^ 11-1. Number of HSYNC after the VSYNC occurs before the beginning of the transfer.

13.6.3.7 RFBI_CMD Register (offset = 4Ch) [reset = 0h]

RFBI_CMD is shown in Figure 13-162 and described in Table 13-129.

The control register configures the RFBI command

Figure 13-162. RFBI_CMD Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
RESERVED	CMD
R/W-0h	R/W-0h

Table 13-129. RFBI_CMD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	CMD	R/W	0h	Command Value 8/9/12/16 bit value depending on the parallel mode

13.6.3.8 RFBI_PARAM Register (offset = 50h) [reset = 0h]

RFBI_PARAM is shown in Figure 13-163 and described in Table 13-130.

The control register configures the RFBI parameter.

Figure 13-163. RFBI_PARAM Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
RESERVED	PARAM
R/W-0h	R/W-0h

Table 13-130. RFBI_PARAM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	PARAM	R/W	0h	Param Value 8/9/12/16 bit value depending on the parallel mode

13.6.3.9 RFBI_DATA Register (offset = 54h) [reset = 0h]

RFBI_DATA is shown in Figure 13-164 and described in Table 13-131.

The control register configures the RFBI data.

Figure 13-164. RFBI_DATA Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">DATA</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 13-131. RFBI_DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R/W	0h	Data value 12/16/18/24/2x16 bit value depending on the Data type

13.6.3.10 RFBI_READ Register (offset = 58h) [reset = 0h]

RFBI_READ is shown in Figure 13-165 and described in Table 13-132.

The control register configures the RFBI read

Figure 13-165. RFBI_READ Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																READ															
R/W-0h																R/W-0h															

Table 13-132. RFBI_READ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	READ	R/W	0h	Read Value 8/9/12/16 bit value depending on the parallel mode

13.6.3.11 RFBI_STS Register (offset = 5Ch) [reset = 0h]

RFBI_STS is shown in Figure 13-166 and described in Table 13-133.

The control register configures the RFBI status.

Figure 13-166. RFBI_STS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STS															
R/W-0h																R/W-0h															

Table 13-133. RFBI_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	STS	R/W	0h	Status value 8/9/12/16 bit value depending on the parallel mode

13.6.3.12 RFBI_CONFIG_0 Register (offset = 60h + [i * 18h]) [reset = 310000h]

RFBI_CONFIG_0 is shown in Figure 13-167 and described in Table 13-134.

The control register allows configuration #I of the RFBI module

Figure 13-167. RFBI_CONFIG_0 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							

23	22	21	20	19	18	17	16
RESERVED		HS_SYNC_POLARITY	TE_VSYNC_POLARITY	CS_POLARITY	WE_POLARITY	RE_POLARITY	A0_POLARITY
R/W-0h		R/W-1h	R/W-1h	R/W-0h	R/W-0h	R/W-0h	R/W-1h

15	14	13	12	11	10	9	8
RESERVED			UNUSED_BITS		CYCLE_FMT		L4_FMT
R/W-0h			R/W-0h		R/W-0h		R/W-0h

7	6	5	4	3	2	1	0
L4_FMT	DATA_TYPE		TIME_GRANULARITY	TRIGGER_MODE		PARALLEL_MODE	
R/W-0h	R/W-0h		R/W-0h	R/W-0h		R/W-0h	

Table 13-134. RFBI_CONFIG_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R/W	0h	
21	HS_SYNC_POLARITY	R/W	1h	HSYNC polarity 0h (R/W) = HSYNC Active Low 1h (R/W) = HSYNC Active High
20	TE_VSYNC_POLARITY	R/W	1h	TE or VSYNC Polarity 0h (R/W) = TE or VSYNC Active Low 1h (R/W) = TE or VSYNC Active High
19	CS_POLARITY	R/W	0h	CS Polarity 0h (R/W) = CS Active Low defined at reset time 1h (R/W) = CS Active High defined at reset time
18	WE_POLARITY	R/W	0h	WE Polarity 0h (R/W) = WE Active Low 1h (R/W) = WE Active High
17	RE_POLARITY	R/W	0h	RE Polarity 0h (R/W) = RE Active Low 1h (R/W) = RE Active High
16	A0_POLARITY	R/W	1h	A0 Polarity 0h (R/W) = A0 Active Low 1h (R/W) = A0 Active High
15-13	RESERVED	R/W	0h	
12-11	UNUSED_BITS	R/W	0h	State of unused bits 0h (R/W) = Low level (0) 1h (R/W) = High level (1) 2h (R/W) = Unchanged from previous state 3h (R/W) = 3

Table 13-134. RFBI_CONFIG_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-9	CYCLE_FMT	R/W	0h	Cycle format 0h (R/W) = 1 cycle for 1 pixel 1h (R/W) = 2 cycle for 1 pixel 2h (R/W) = 3 cycle for 1 pixel 3h (R/W) = 3 cycle for 2 pixels
8-7	L4_FMT	R/W	0h	L4 Write Access format 0h (R/W) = 1 pixel per L4 access to the register data 1h (R/W) = 1 2h (R/W) = 2 pixels per L4 access to the register data with 1st pixel at the position [15:0] 3h (R/W) = 2 pixels per L4 access to the register data with 1st pixel at the position [31:16]
6-5	DATA_TYPE	R/W	0h	Data type from the display controller and L4 0h (R/W) = 12-bit 1h (R/W) = 16-bit 2h (R/W) = 18-bit 3h (R/W) = 24-bit
4	TIME_GRANULARITY	R/W	0h	Multiplies signal timing latencies by two 0h (R/W) = x2 latencies disabled 1h (R/W) = x2 latencies enabled
3-2	TRIGGER_MODE	R/W	0h	Trigger Mode 0h (R/W) = Internal trigger mode (ITE bit mode) 1h (R/W) = External trigger mode (TE signal) 2h (R/W) = External trigger mode (VSYNC/HSYNC signals) 3h (R/W) = 3
1-0	PARALLEL_MODE	R/W	0h	Parallel Mode 0h (R/W) = 8-bit parallel output interface selected 1h (R/W) = 9-bit parallel output interface selected 2h (R/W) = 12-bit parallel output interface selected 3h (R/W) = 16-bit parallel output interface selected

13.6.3.13 RFBI_ONOFF_TIME_0 Register (offset = 64h + [i * 18h]) [reset = 0h]

RFBI_ONOFF_TIME_0 is shown in Figure 13-168 and described in Table 13-135.

The control register allows configuration of the RFBI timing.

Figure 13-168. RFBI_ONOFF_TIME_0 Register

31	30	29	28	27	26	25	24
RESERVED		RE_OFFTIME					
R/W-0h		R/W-0h					

23	22	21	20	19	18	17	16
RE_ONTIME				WE_OFFTIME			
R/W-0h				R/W-0h			

15	14	13	12	11	10	9	8
WE_OFFTIME		WE_ONTIME				CS_OFFTIME	
R/W-0h		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
CS_OFFTIME				CS_ONTIME			
R/W-0h				R/W-0h			

Table 13-135. RFBI_ONOFF_TIME_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	RE_OFFTIME	R/W	0h	Read Enable assertion time from start access time Number of L4Clk cycles
23-20	RE_ONTIME	R/W	0h	Read Enable assertion time from start access time Number of L4Clk cycles
19-14	WE_OFFTIME	R/W	0h	CS deassertion time from start access time Number of L4Clk cycles
13-10	WE_ONTIME	R/W	0h	CS deassertion time from start access time Number of L4Clk cycles
9-4	CS_OFFTIME	R/W	0h	CS deassertion time from start access time Number of L4Clk cycles
3-0	CS_ONTIME	R/W	0h	CS assertion time from start access time Number of L4Clk cycles

13.6.3.14 RFBI_CYCLE_TIME_0 Register (offset = 68h + [i * 18h]) [reset = 0h]

RFBI_CYCLE_TIME_0 is shown in Figure 13-169 and described in Table 13-136.

The control register allows configuration of the RFBI timing.

Figure 13-169. RFBI_CYCLE_TIME_0 Register

31	30	29	28	27	26	25	24
RESERVED				ACCESS_TIME			
R/W-0h				R/W-0h			

23	22	21	20	19	18	17	16
ACCESS_TIME		WR_EN	WW_EN	RR_EN	RW_EN	CS_PULSE_WIDTH	
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

15	14	13	12	11	10	9	8
CS_PULSE_WIDTH				RECYCLE_TIME			
R/W-0h				R/W-0h			

7	6	5	4	3	2	1	0
RECYCLE_TIME		WE_CYCLE_TIME					
R/W-0h		R/W-0h					

Table 13-136. RFBI_CYCLE_TIME_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-22	ACCESS_TIME	R/W	0h	Access Time Number of L4Clk cycles
21	WR_EN	R/W	0h	Write to Read Pulse Width Enable (same CS) 0h (R/W) = CSPulseWidth does not apply 1h (R/W) = CSPulseWidth applies
20	WW_EN	R/W	0h	Read to Read Pulse Width Enable (same CS) 0h (R/W) = CSPulseWidth does not apply 1h (R/W) = CSPulseWidth applies
19	RR_EN	R/W	0h	Read to Read Pulse Width Enable (same CS) 0h (R/W) = CSPulseWidth does not apply 1h (R/W) = CSPulseWidth applies
18	RW_EN	R/W	0h	Read to Write Pulse Width Enable (same CS) 0h (R/W) = CSPulseWidth does not apply 1h (R/W) = CSPulseWidth applies
17-12	CS_PULSE_WIDTH	R/W	0h	CS Pulse Width Number of L4Clk cycles
11-6	RECYCLE_TIME	R/W	0h	RE Cycle Time RW 0x00 Number of L4Clk cycles
5-0	WE_CYCLE_TIME	R/W	0h	WE Cycle Time Number of L4Clk cycles

13.6.3.15 RFBI_DATA_CYCLE1_0 Register (offset = 6Ch + [i * 18h]) [reset = 0h]

RFBI_DATA_CYCLE1_0 is shown in Figure 13-170 and described in Table 13-137.

The control register configures the RFBI data format for 1st cycle.

Figure 13-170. RFBI_DATA_CYCLE1_0 Register

31	30	29	28	27	26	25	24
RESERVED				BIT_ALIGNMENT_PIXEL2			
R/W-0h				R/W-0h			

23	22	21	20	19	18	17	16
RESERVED			NB_BITS_PIXEL2				
R/W-0h			R/W-0h				

15	14	13	12	11	10	9	8
RESERVED				BIT_ALIGNMENT_PIXEL1			
R/W-0h				R/W-0h			

7	6	5	4	3	2	1	0
RESERVED			NB_BITS_PIXEL1				
R/W-0h			R/W-0h				

Table 13-137. RFBI_DATA_CYCLE1_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	BIT_ALIGNMENT_PIXEL2	R/W	0h	Bit alignment Alignment of the bits from pixel#2 on the output interface
23-21	RESERVED	R/W	0h	
20-16	NB_BITS_PIXEL2	R/W	0h	Number of bits Number of bits from the pixel #2 (value from 0 to16 bits). The values from 17 to 31 are invalid.
15-12	RESERVED	R/W	0h	
11-8	BIT_ALIGNMENT_PIXEL1	R/W	0h	Bit alignment Alignment of the bits from pixel#1 on the output interface
7-5	RESERVED	R/W	0h	
4-0	NB_BITS_PIXEL1	R/W	0h	Number of bits Number of bits from the pixel #1 (value from 0 to16 bits). The values from 17 to 31 are invalid.

13.6.3.16 RFBI_DATA_CYCLE2_0 Register (offset = 70h + [i * 18h]) [reset = 0h]

RFBI_DATA_CYCLE2_0 is shown in Figure 13-171 and described in Table 13-138.

The control register configures the RFBI data format for 2nd cycle.

Figure 13-171. RFBI_DATA_CYCLE2_0 Register

31	30	29	28	27	26	25	24
RESERVED				BIT_ALIGNMENT_PIXEL2			
R/W-0h				R/W-0h			

23	22	21	20	19	18	17	16
RESERVED			NB_BITS_PIXEL2				
R/W-0h			R/W-0h				

15	14	13	12	11	10	9	8
RESERVED				BIT_ALIGNMENT_PIXEL1			
R/W-0h				R/W-0h			

7	6	5	4	3	2	1	0
RESERVED			NB_BITS_PIXEL1				
R/W-0h			R/W-0h				

Table 13-138. RFBI_DATA_CYCLE2_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	BIT_ALIGNMENT_PIXEL2	R/W	0h	Bit alignment Alignment of the bits from pixel#2 on the output interface
23-21	RESERVED	R/W	0h	
20-16	NB_BITS_PIXEL2	R/W	0h	Number of bits Number of bits from the pixel #2 (value from 0 to16 bits). The values from 17 to 31 are invalid.
15-12	RESERVED	R/W	0h	
11-8	BIT_ALIGNMENT_PIXEL1	R/W	0h	Bit alignment Alignment of the bits from pixel#1 on the output interface
7-5	RESERVED	R/W	0h	
4-0	NB_BITS_PIXEL1	R/W	0h	Number of bits Number of bits from the pixel #1 (value from 0 to16 bits). The values from 17 to 31 are invalid.

13.6.3.17 RFBI_DATA_CYCLE3_0 Register (offset = 74h + [i * 18h]) [reset = 0h]

RFBI_DATA_CYCLE3_0 is shown in Figure 13-172 and described in Table 13-139.

The control register configures the RFBI data format for 3rd cycle.

Figure 13-172. RFBI_DATA_CYCLE3_0 Register

31	30	29	28	27	26	25	24
RESERVED				BIT_ALIGNMENT_PIXEL2			
R/W-0h				R/W-0h			

23	22	21	20	19	18	17	16
RESERVED			NB_BITS_PIXEL2				
R/W-0h			R/W-0h				

15	14	13	12	11	10	9	8
RESERVED				BIT_ALIGNMENT_PIXEL1			
R/W-0h				R/W-0h			

7	6	5	4	3	2	1	0
RESERVED			NB_BITS_PIXEL1				
R/W-0h			R/W-0h				

Table 13-139. RFBI_DATA_CYCLE3_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	BIT_ALIGNMENT_PIXEL2	R/W	0h	Bit alignment Alignment of the bits from pixel#2 on the output interface
23-21	RESERVED	R/W	0h	
20-16	NB_BITS_PIXEL2	R/W	0h	Number of bits Number of bits from the pixel #2 (value from 0 to16 bits). The values from 17 to 31 are invalid.
15-12	RESERVED	R/W	0h	
11-8	BIT_ALIGNMENT_PIXEL1	R/W	0h	Bit alignment Alignment of the bits from pixel#1 on the output interface
7-5	RESERVED	R/W	0h	
4-0	NB_BITS_PIXEL1	R/W	0h	Number of bits Number of bits from the pixel #1 (value from 0 to16 bits). The values from 17 to 31 are invalid.

13.6.3.18 RFBI_VSYNC_WIDTH Register (offset = 90h) [reset = 0h]

RFBI_VSYNC_WIDTH is shown in Figure 13-173 and described in Table 13-140.

The control register configures the RFBI VSYNC minimum pulse width

Figure 13-173. RFBI_VSYNC_WIDTH Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R/W-0h</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">MIN_VSYNC_PULSE_WIDTH</td></tr>
<tr><td colspan="16">R/W-0h</td></tr>
</table>

Table 13-140. RFBI_VSYNC_WIDTH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	MIN_VSYNC_PULSE_WIDTH	R/W	0h	Programmable min VSYNC pulse width Minimum VSYNC pulse width from 0 to 65535. Number of L4 clock cycles to determine when VSYNC pulse occurs. The values 0 and 1 are invalid.

13.6.3.19 RFBI_HSYNC_WIDTH Register (offset = 94h) [reset = 0h]

RFBI_HSYNC_WIDTH is shown in Figure 13-174 and described in Table 13-141.

The control register configures the RFBI HSYNC minimum pulse width.

Figure 13-174. RFBI_HSYNC_WIDTH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIN_HSYNC_PULSE_WIDTH															
R/W-0h															

Table 13-141. RFBI_HSYNC_WIDTH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	MIN_HSYNC_PULSE_WIDTH	R/W	0h	Programmable min HSYNC pulse width Minimum HSYNC pulse width from 0 to 65535. Number of L4 clock cycles to determine when HSYNC pulse occurs. The values 0 and 1 are invalid.

Camera (VPFE)

This chapter describes the camera (VPFE) of the device.

NOTE: The Camera (VPFE) is not supported for the AMIC120.

Topic **Page**

14.1 Introduction

14.1.1 VPFE Features

The general features of the VPFE module include:

- A buffer memory for interfacing to the DMA at the chip level and preventing the CCDC module from overflowing.
- Support for conventional Bayer pattern and Foveon sensor formats
- Generates HD/VD timing signals and field ID to an external timing generator or can synchronize to the external timing generator
- Support for progressive and interlaced sensors (hardware support for up to 2 fields and firmware support for higher number of fields, typically 3-, 4-, and 5-field sensors)
- Support for up to the lesser of 75 MHz or 1/2 dma_ocp_clk sensor clock in the normal mode of operation
- Support for REC656/CCIR-656 standard (YCbCr 422 format, either 8- or 16-bit)
- Support for YCbCr 422 format, either 8- or 16-bit with discrete H and VSYNC signals.
- Support for up to 16-bit input
- Generates optical black clamping signals
- Support for digital clamping and black level compensation
- Support for 10-bit to 8-bit A-law compression
- Support for a low-pass filter prior to writing to SDRAM. If this filter is enabled, 2 pixels each in the left and right edges of each line are cropped from the output.
- Support for generating output to range from 16-bits to 8-bits wide (8-bits wide allows for 50% saving in storage area)
- Support for downsampling via programmable culling patterns
- Ability to control output to the SDRAM via an external write enable signal
- Support for up to 16K pixels (image size) in both the horizontal and vertical direction.

14.1.2 Unsupported Features

The VPFE module does not support the following features.

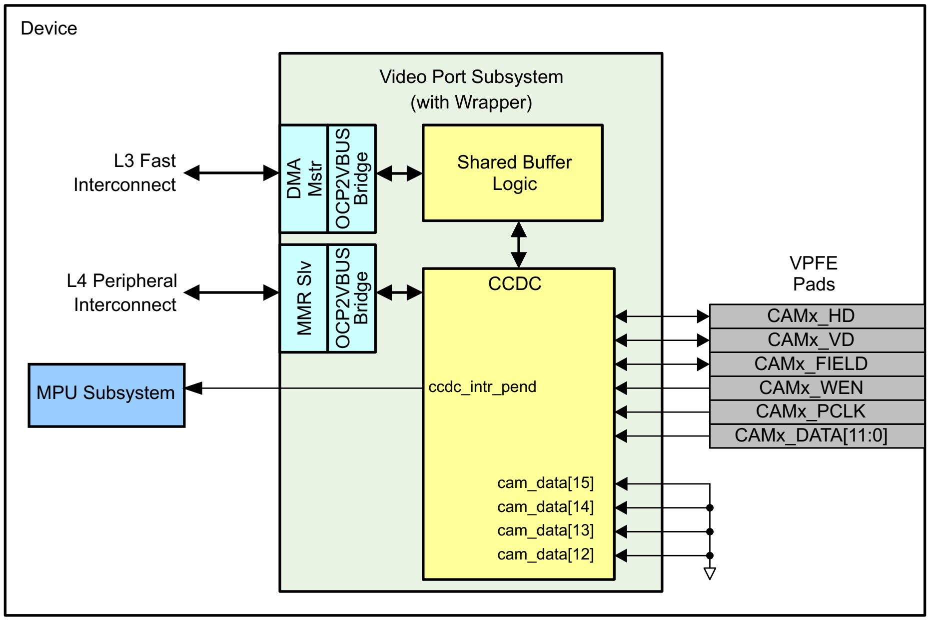
Table 14-1. Unsupported VPFE Features

Feature	Reason
VENC/VPBE Interface	There is no VENC on this device and therefore, Video Encoder direct interface is not connected.
16-bit Input	Only ccdc_data[11:0] pinned out

14.2 Integration

This device includes two instantiations of the Video Port Front End (VPFE) for connection to CCD cameras or BT.656 compliant video encoders.

Figure 14-1. VPFE Integration



14.2.1 VPFE Connectivity Attributes

The general connectivity attributes for the VPFE module are shown in Table 14-2.

Table 14-2. VPFE Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L3_GCLK PD_PER_L3S_GCLK CCDCx_PCLK
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Standby Smart Idle
Interrupt Requests	1 interrupt to MPU Subsystem (CCDCx_INT)
DMA Requests	None
Physical Address	L4 peripheral slave port L3 initiator port

14.2.2 VPFE Clock and Reset Management

The VPFE has separate functional and interface clocks. The pixel clock is supplied from an external pin and is restricted such that ccd_pclk_clk ≤ ½ dma_ocp_clk.

Table 14-3. VPFE Clock Signals

Clock Signal	Maximum Frequency	Reference Source	Comments
mmr_ocp_clk MMR interface clock	100 MHz	CORE_CLKOUTM4	pd_per_l3_gclk from PRCM (divided down using mmr_ocp_clkdiv)
dma_ocp_clk L3 phase clock enable used to create L4 clock	200 MHz	CORE_CLKOUTM4	pd_per_l3_gclk from PRCM
ccd_pclk_clk Functional clock	The lesser of 75 MHz or ½ dma_ocp_clk	CCDCX_PCLK	From external pin

14.2.3 VPFE Pin List

The external pins for the VPFE module are shown in Table 14-4.

Table 14-4. VPFE Pin List

Pin	Type	Description
cam_data[11:0]	I	Camera/VPFE data
cam_hd	I/O	Horizontal sync
cam_vd	I/O	Vertical sync
cam_field	I/O	Field indicator
cam_wen	I	Write enable
cam_pclk	I	Pixel clock

14.3 Functional Description

14.3.1 External IO Interface

The following subsections explain how to connect VPFE signal pins to various input devices.

14.3.1.1 Summary

Table 14-5 summarizes the VPFE signal pins required by common input devices.

Table 14-5. Summary of VPFE Signal Pins and Common Input Devices

Signal Name	I/O	Description	Raw Data Mode 16-bit	Raw Data Mode 8-bit	Raw Data Mode, no Field or WEN 16-bit	Raw Data Mode, no Field or WEN 8-bit	BT.656 Mode 10-bit	BT.656 Mode 8-bit	Digital YCbCr Mode 16-bit	Digital YCbCr Mode 8-bit
cam_hd	I/O	Horizontal sync	+	+	+	+			+	+
cam_vd	I/O	Vertical sync	+	+	+	+			+	+
cam_field	I/O	Field indicator	+	+					+	+
cam_wen	I	Write enable	+	+					+	+
cam_pclk	I	Pixel clock	+	+	+	+	+	+	+	+
cam_data0	I	Camera/ VPFE data (bit 0)	+	+	+	+	+	+	+	+
cam_data1	I	Camera/ VPFE data (bit 1)	+	+	+	+	+	+	+	+
cam_data2	I	Camera/ VPFE data (bit 2)	+	+	+	+	+	+	+	+
cam_data3	I	Camera/ VPFE data (bit 3)	+	+	+	+	+	+	+	+
cam_data4	I	Camera/ VPFE data (bit 4)	+	+	+	+	+	+	+	+
cam_data5	I	Camera/ VPFE data (bit 5)	+	+	+	+	+	+	+	+
cam_data6	I	Camera/ VPFE data (bit 6)	+	+	+	+	+	+	+	+
cam_data7	I	Camera/ VPFE data (bit 7)	+	+	+	+	+	+	+	+
cam_data8	I	Camera/ VPFE data (bit 8)	+		+		+		+	
cam_data9	I	Camera/ VPFE data (bit 9)	+		+		+		+	
cam_data10	I	Camera/ VPFE data (bit 10)	+		+				+	
cam_data11	I	Camera/ VPFE data (bit 11)	+		+				+	

14.3.1.2 Raw Data Mode

Raw data mode is a generic parallel interface that supports up to a 16-bit data path to a CMOS or CCD sensor. The signal interface is described in Table 14-6.

Table 14-6. CCD Interface Signals

Name	I/O	Function
CAM_D[15:0]	I	Image data – A mode set by INPMOD (not R656ON). • Bit width is configurable between 8 and 16 bits (DATSIZ). • The polarity of the input image data is configurable (DATAPOL).
CAM_VD	I	VSYNC - vertical sync signal
CAM_HD	I	HSYNC - horizontal sync signal
CAM_FIELD	I	Field identification signal (optional – FLDMODE) • This signal can be configured to be latched by the VD signal (FIDMD). • The polarity of the field identification signal is configurable (FLDPOL).
CAM_WEN	I	VPFE write enable signal (optional – EXWEN) • The EXWEN signal determines when data is captured, processed, and saved to memory or sent for further processing. • If EXWEN is enabled, image data will be captured, processed, and saved to memory or sent for further processing, depending on the state of WENLOG. • Data can be saved either when CAM_WEN is active and the pixels are within the internal frame (SPH, NPH, SLV, NLV) or when the pixels are within the internal frame (WENLOG).
CAM_PCLK	I	Pixel clock • The PCLK signal is the signal used to latch input image data. • The input image data can be captured on either the rising or on the falling edge of the PCLK signal which is configured by field PCLK_INV. • The maximum pixel clock rate is 75 MHz.

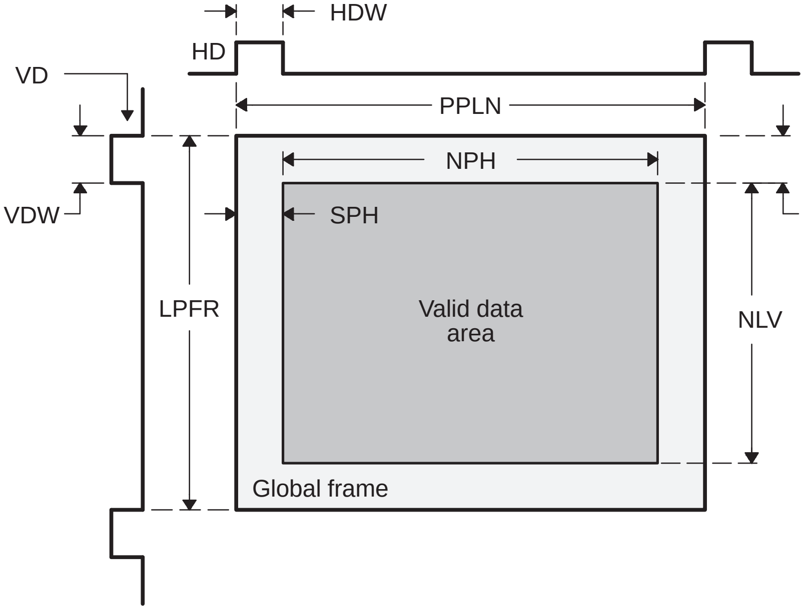
14.3.1.2.1 Mode Information – Always Required

- INPMOD – input mode
- DATSIZ – size (bit width) of input data – always stored in LSBs
- DATAPOL – polarity of input data
- VDPOL – VD polarity
- HDPOL – HD polarity
- FLDMOD – field mode

14.3.1.2.2 Timing Information – Optional, Depending on Control Signals and Sensor Mode

- If FLDMODE is enable
  - FLDPOL – CAM_FIELD polarity
  - FIDMD – CAM_FIELD latch information
- EXWEN – external CAM_WEN signal
  - WENLOG – determines when data is valid along with frame settings

Figure 14-2. CCD Controller Frame and Control Signal Definitions



HDW − Horizontal sync width
HD − Horizontal sync
PPLN − Pixels per line
LPFR − Lines per frame
SPH − Start pixel horizontal
NPH − Size horizontal of valid area
VDW − Veritcal sync width
VD − Vertical sync
OBS − Optical black start
SLVx − Start line vertical field0 or field1
NLV − Size vertical of valid area

14.3.1.3 ITU-R BT.656 Interface

The BT.656 interface supports either 8-bit or 10-bit processing of input video YCbCr data. See Section 14.4 for instructions on how to configure this mode.

Since the data synchronization information is carried along with the data lines, no synchronization signals (i.e., CAM_HD, CAM_VD, and CAM_FIELD) are necessary in this mode.

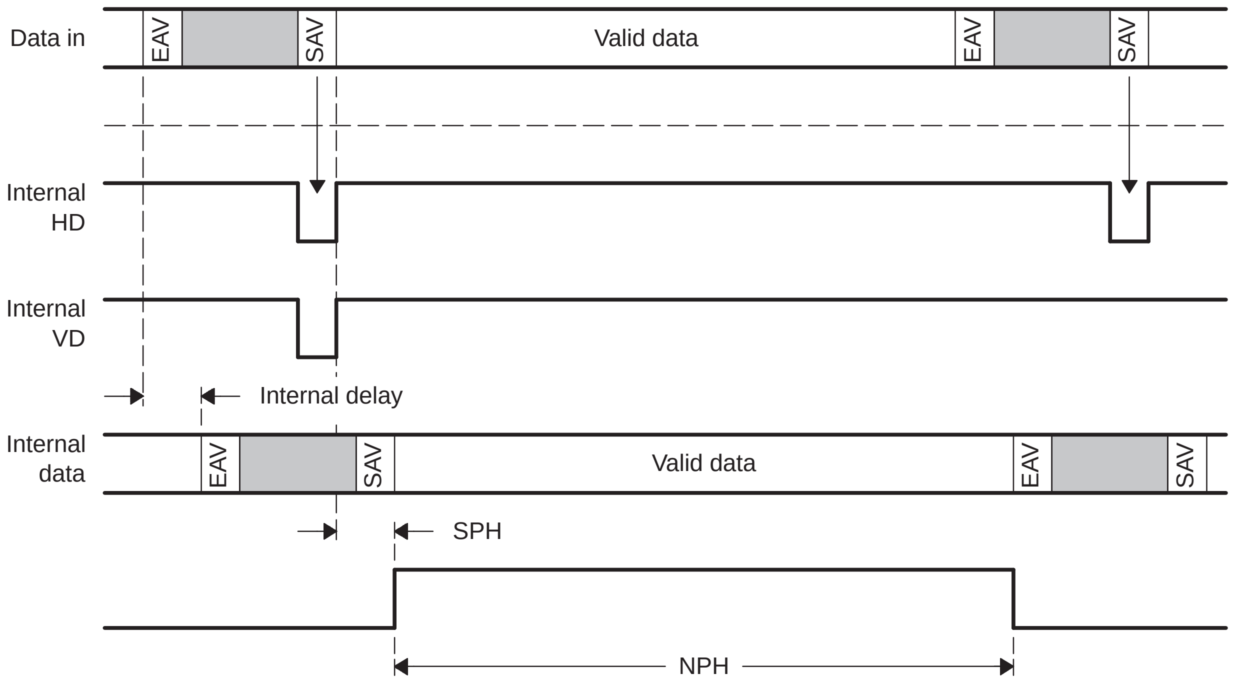
The signal interface is described in Table 14-7.

Table 14-7. ITU-R BT.656 Interface Signals

Name	I/O	Function
CAM_D[9:0]	I	Image data – mode set by R656ON • Bit width is configurable to either eight or ten bits (BW656). • The polarity of the input image data is configurable (DATAPOL).
CAM_PCLK	I	Pixel clock • The PCLK signal is the signal used to latch input image data. • The input image data can be captured on either the rising or on the falling edge of the PCLK signal which is configured by field PCLK_INV. • The maximum pixel clock rate is 75 MHz.

Two timing reference codes are transmitted as the synchronization signal. At the start and end of each video data block, two unique codes are sent, respectively. The start code is called the start of active video signal (SAV), and the end code is called the end of active video signal (EAV). The SAV and EAV codes proceed and follow valid data, as shown in Figure 14-3. The VPFE controller internally bases on SAV and EAV codes to generate the necessary synchronization signals, i.e., horizontal sync, vertical synx, and field ID.

Figure 14-3. BT.656 Signal Interface



Both timing reference signals, SAV and EAV, consist of a four-word sequence in the following format: FF 00 00 XY, where FF 00 00 are a set preamble and the fourth word defines the field identification, the state of vertical field blanking, the state of horizontal line blanking, and protection (error correction) codes. The bit format of the fourth word is shown in Table 14-8 and the definitions for bits, F, V, and H, are given in Table 14-9. F, V, and H are used in place of the usual horizontal sync, vertical sync, and blank timing control signals. Bits P3, P2, P1, and P0 are protection (error correction) bits for F, V, and H. The relationship the between F, V, and H and the protection (error correction) bits is given in Table 14-10. To enable error correction, set the ECCFVH bit in the REC656IF register. The VPFE controller automatically detects and applies error correction when the ECCFVH bit is set.

Table 14-8. Video Timing Reference Codes for SAV and EAV

Data Bit Number	First Word (FF)	Second Word (00)	Third Word (00)	Fourth Word (XY)
9 (MSB)	1	0	0	1
8	1	0	0	F
7	1	0	0	V
6	1	0	0	H
5	1	0	0	P3
4	1	0	0	P2
3	1	0	0	P1
2	1	0	0	P0
1	1	0	0	0
0	1	0	0	0

Table 14-9. F, V, H Signal Descriptions

Signal	Value	Command
F	0	Field 1
	1	Field 2
V	0	0
	1	Vertical blank
H	0	SAV
	1	EAV

Table 14-10. F, H, V Protection (error correction) Bits

F	V	H	P3	P2	P1	P0
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	1	0	0
1	1	1	0	0	0	1

NOTE: The VPFE controller outputs the XY code in the SAV and EAV into the external memory. In order to eliminate this, set the SPH register field to +1. Also set the NPH register field to accurately represent the number of active pixels.

14.3.1.4 Digital YCbCr Interface

The digital YCbCr interface supports either 8-bit or 16-bit devices. The signal interface is described in Table 14-11.

Unlike the BT.656 mode, discrete horizontal sync (CAM_HD) and vertical sync (CAM_VD) signals are required. An NTSC/PAL decoder is an example device that can be connected to the YCbCr interface.

You can use data lines YI[7:0] or CI [7:0] for input in 8-bit mode. Alternately, you can connect two separate devices; however, only one can be active at any given time. Setting the YCINSWP bit in the CCDCFG register determines which set of 8-bit inputs are active.

Use data lines YI [7:0] and CI[7:0] for input in 16-bit mode. Use the YCINSWP bit in the CCDCFG register to swap the Y and Cr/Cb data lines.

Table 14-11. CCD Interface Signals

Name	I/O	Function
CAM_D [15:0] = YI [7:0] / CI [7:0]	I	Image data – mode set by INPMOD (not R656ON) • Bit width is only configurable to either 8 bits or 16 bits (INPMOD). • The polarity of the input image data is reversible (DATAPOL). • When the 16-bit interface is used, you can swap the Y and C inputs (YCINSWP). • When the 8-bit interface is used, you can connect either half of the bus (YCINSWP). • When the 8-bit interface is used, you can set the position of the Y data to either the even or odd pixel (Y8POS).
CAM_VD	I	VSYNC - vertical sync signal
CAM_HD	I	HSYNC - horizontal sync signal
CAM_FIELD	I	Field identification signal (optional – FLDMODE) • This signal can be configured to be latched by the VD signal (FIDMD). • The polarity of the field identification signal is configurable (FLDPOL).
CAM_PCLK	I	Pixel clock • The PCLK signal is the signal used to latch input image data. • The input image data can be captured on either the rising or on the falling edge of the PCLK signal which is configured by field PCLK_INV. • The maximum pixel clock rate is 75 MHz.

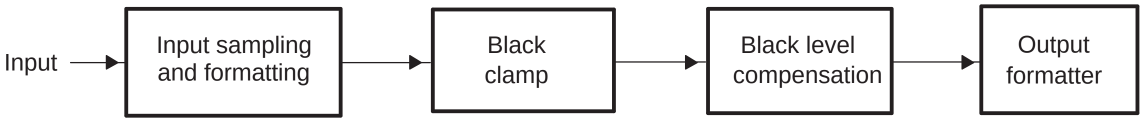
14.3.2 VPFE Data / Image Processing

This section describes the image/data processing of each module in the VPFE in more detail. Section 14.3.2.1 describes the raw data mode while Section 14.3.2.2 explains the YCbCr and BT656 modes.

14.3.2.1 Raw Data Mode

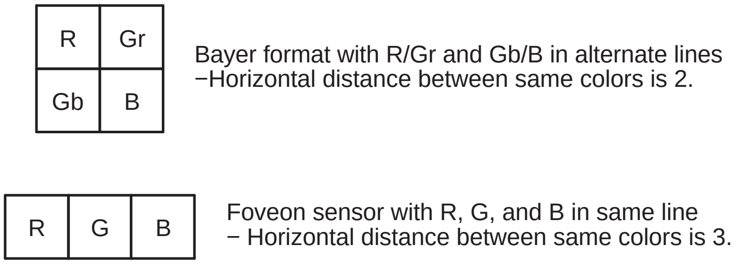
Raw data mode is enabled by setting SYN_MODE.INPMODE to 0 and setting REC656IF.REC656ON to 0. Figure 14-4 shows the corresponding data processing diagram.

Figure 14-4. Data Processing in Raw Data Mode



Typically, raw sensor data is one color per pixel in a color filter array (CFA). The color filter array applied is typically a Bayer pattern, as shown in Figure 14-5 for RGB color space. Alternatively, the special Foveon X3-family sensors capture R, G, and B lights at each pixel location. Both Bayer and Foveon sensors are supported by the VPFE controller.

Figure 14-5. Color Patterns

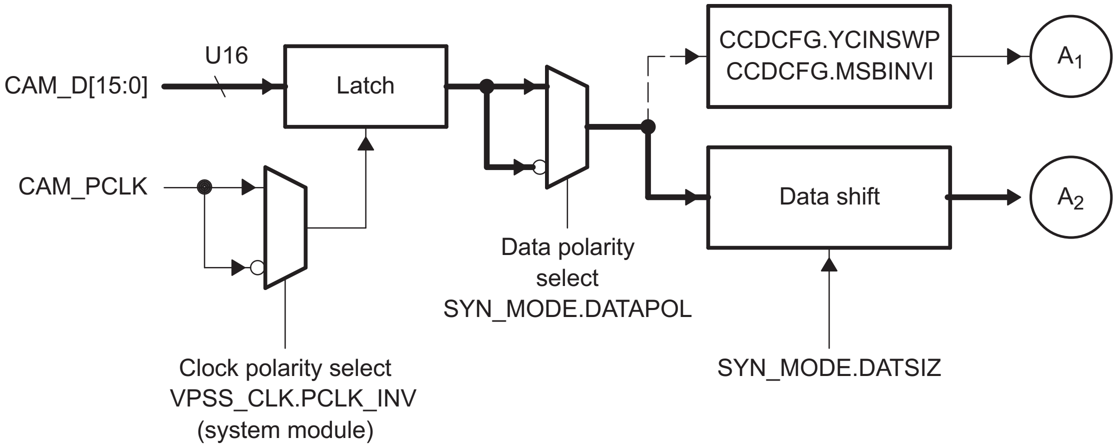


14.3.2.1.1 Input Sampling and Formatting

The data path of raw data mode is shown in the thicker lines in Figure 14-6 (i.e., A2). Data path A1 is applicable to YUV input mode only.

- The pixel clock (CAM_PCLK) latches the data.
- Pixel clock polarity can be either rising or falling edge and is set in VPSS clock control register (VPSS_CLK.PCLK_INV).
- DATAPOL bit in the SYN_MODE register affects the data representation.
- Data is right-shifted to align the data in the least significant bits of the data bus and provide the maximum dynamic range for the remainder of the processing (DATSIZ bit in the SYN_MODE register). This also sets the maximum data size allowed in subsequent clipping/limiting operations and is the output data alignment when it is written to external memory.

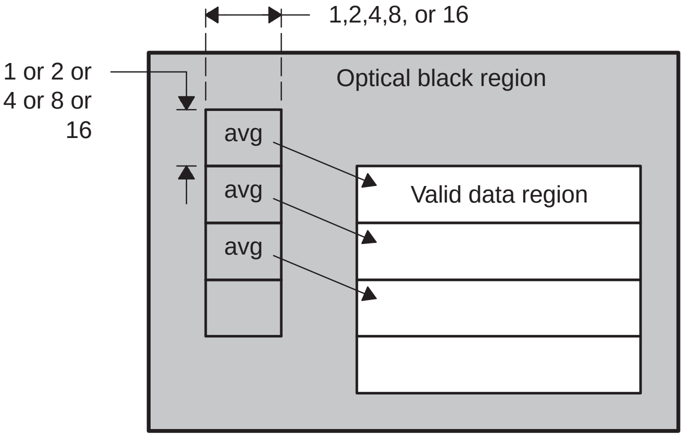
Figure 14-6. Input Formatter



14.3.2.1.2 Optical Black Clamping

Sensor manufacturers typically provide some optically masked pixels at the beginning/end of each line to allow you to determine the noise floor on any given frame of data. The optical black clamping function provides a means to average the optically black pixels and subtract that value from each input pixel as the first step in reducing the noise on the input pixels.

Figure 14-7. Optical Black Averaging & Application

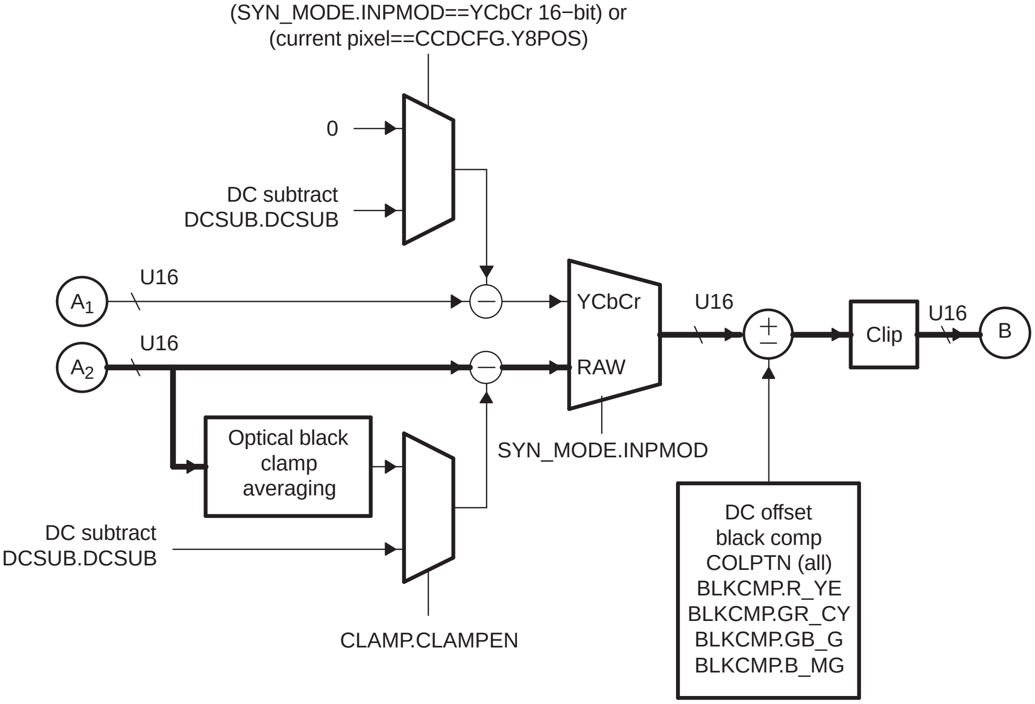


The averaging circuit takes an average of masked (black) pixel values from the image sensor, averaging pixels at the start (OBST bit in the CLAMP register) of each line (OBSLEN bit in the CLAMP register) and for the number of indicated lines (OBSLN bit in the CLAMP register), plus an optional gain adjustment (OBGAIN bit in the CLAMP register). The resultant value is subtracted from the image data at the succeeding line. You can control the position of the black pixels, the number of pixels (8 or 16) in each line that are averaged, and the number of lines (8 or 16) that are averaged.

Alternately, you can disable black clamp averaging (CLAMPEN bit in the CLAMP register) and select a constant black value for subtraction (DCSUB bit in the DCSUB register) instead of using the calculated average value.

The corresponding data path is shown in the thicker lines in Figure 14-8.

Figure 14-8. Black Clamping and Black Level Compensation



14.3.2.1.3 Black Level Compensation

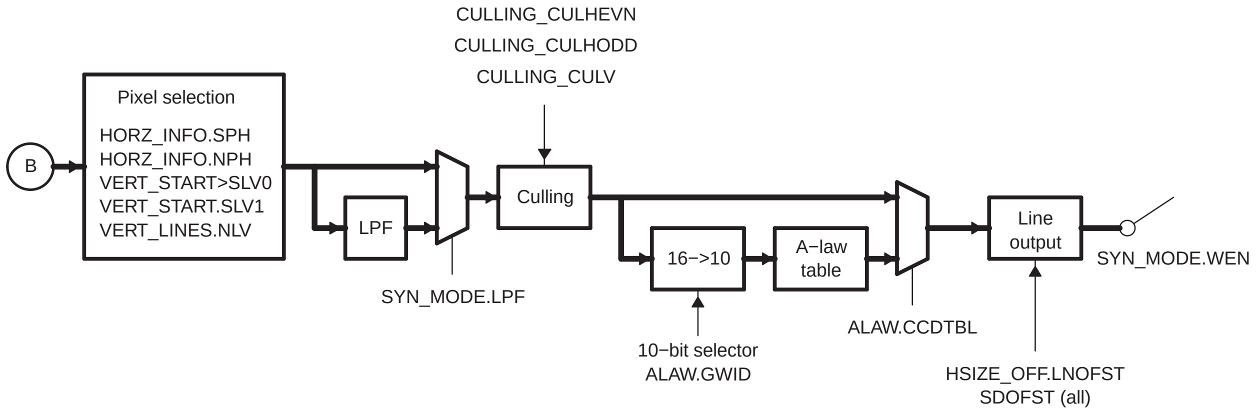
After the digital clamping is applied to the data, black level compensation is applied (Figure 14-8). In this operation, a fixed value is subtracted from the data depending on the color (i.e., R/Ye, Gr/Cy, Gb/G, and B/Mg). The offset (BLKCMP register, fields R_YE, GR_CY, GB_G, B_MG) that is applied to each data sample is selected according to the 0/1/2/3 phase and the color (0/1/2/3) specified for each phase (COLPTN). The color pattern definition is very flexible in order to accommodate many different capture devices, including normal Bayer CFA sampling, Foveon sensors, and VGA movie mode CCDs, whose VGA draft mode output does not follow the typical Bayer pattern.

14.3.2.1.4 Output Formatter

The final stage of VPFE processing is the output formatter, as shown in Figure 14-9. A framing selection is applied to limit the processing area by the settings in the HORZ_INFO, VERT_START and VERT_LINES registers.

NOTE: In addition to the framing applied at the beginning and end of the data formatter operation, you must apply a framing selection to limit the processing area. Please ensure that the settings are relative to that frame.

Figure 14-9. Output Formatter



14.3.2.1.4.1 Low Pass Filter

Use the LPF bit in the SYN_MODE register to apply an optional low-pass filter after the reframing. The low-pass filter consists of a simple 3-tap (¼ , ½ , ¼) filter. Two pixels on the left and two pixels on the right of each line are cropped if the filter is enabled.

14.3.2.1.4.2 Culling

Use the CULEVEN and CULODD bits in the CULLING register (8-bit repeating mask, one per field) to enable an optional culling operation, which culls (deletes) selected pixel data from a line. Use the CULV bit in the CULLING register to select lines from a frame.

Table 14-12 illustrates how the register values apply the decimation pattern to the data. The pixels in white will be saved to external memory and the shaded pixels are discarded.

In this case:

- CULLING = 0x59C40066, with
- CULHEVN = 0x59,
- CULHODD = 0xC4, and
- CULV = 0x66.

Table 14-12. Example for Decimation Pattern

		LSB							
	CULHEVN	0	0	1	0	0	0	1	1	
	CULHODD	1	0	0	1	1	0	1	0	0
0th line		▨	▨	▨	▨	▨	▨	▨	▨	1
1st line			▨	▨			▨		▨	1
2nd line		▨	▨		▨	▨	▨			0
3rd line		▨	▨	▨	▨	▨	▨	▨	▨	0
4th line		▨	▨	▨	▨	▨	▨	▨	▨	1
5th line			▨	▨			▨		▨	1
6th line		▨	▨		▨	▨	▨			0
7th line		▨	▨	▨	▨	▨	▨	▨	▨	CULV

14.3.2.1.4.3 A-Law Transformation

Use the CCDTBL bit in the ALAW register to apply an optional 10-to-8-bit A-Law transformation using a fixed A-Law table as the final processing stage. Using this causes the data width to reduce to 8 bits and allows packing to 8 bits/pixel when saving to external memory. Since the data resolution can be greater than 10 bits at this stage, you must select the 10 bits for input to the A-Law operation. Use the GWID bit in the ALAW register to select the 10 bits for input.

Figure 14-10. A-Law Table

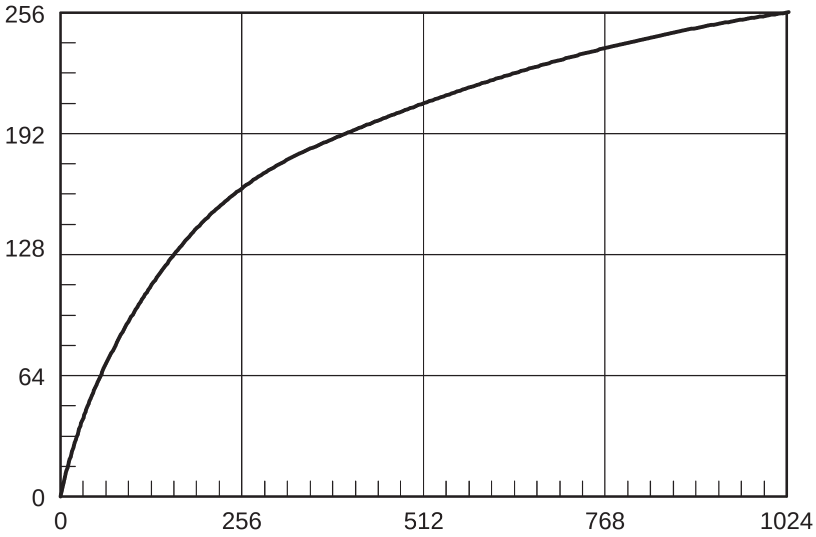


Table 14-13. A-Law Table – Part 1

Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law
0	0	64	64	128	112	192	140	256	161	320	176	384	189	448	200
1	1	65	65	129	113	193	141	257	161	321	176	385	189	449	200
2	2	66	66	130	113	194	141	258	161	322	177	386	189	450	200
3	3	67	67	131	114	195	142	259	161	323	177	387	189	4531	200
4	4	68	68	132	114	196	142	260	162	324	177	388	190	452	200
5	5	69	69	133	115	197	142	261	162	325	177	389	190	453	200
6	6	70	70	134	115	198	143	262	162	326	177	390	190	454	201
7	7	71	71	135	116	199	143	263	162	327	178	391	190	455	201
8	8	72	72	136	116	200	143	264	163	328	178	392	190	456	201
9	9	73	73	137	117	201	144	265	163	329	178	393	190	457	201
10	10	74	74	138	117	202	144	266	163	330	178	394	191	458	201
11	11	75	75	139	118	203	144	267	163	331	178	395	191	459	201
12	12	76	76	140	118	204	145	268	164	332	179	396	191	460	201
13	13	77	77	141	119	205	145	269	164	333	179	397	191	461	202
14	14	78	78	142	119	206	145	270	164	334	179	398	191	462	202
15	15	79	78	143	120	207	146	271	164	335	179	399	191	463	202
16	16	80	79	144	120	208	146	272	165	336	179	400	192	464	202
17	17	81	80	145	121	209	146	273	165	337	180	401	192	465	202
18	18	82	81	146	121	210	147	274	165	338	180	402	192	466	202
19	19	83	82	147	122	211	147	275	166	339	180	403	192	467	202
20	20	84	83	148	122	212	147	276	166	340	180	404	192	468	203
21	21	85	84	149	123	213	148	277	166	341	181	405	193	469	203
22	22	86	84	150	123	214	148	278	166	342	181	406	193	470	203
23	23	87	85	151	124	215	148	279	167	343	181	407	193	471	203

Table 14-13. A-Law Table – Part 1 (continued)

Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law
24	24	88	86	152	124	216	149	280	167	344	181	408	193	472	203
25	25	89	87	153	125	217	149	281	167	345	181	409	193	473	203
26	26	90	88	154	125	218	149	282	167	346	182	410	193	474	204
27	27	91	88	155	125	219	150	283	168	347	182	411	194	475	204
28	28	92	89	156	126	220	150	284	168	348	182	412	194	476	204
29	29	93	90	157	126	221	150	285	168	349	182	413	194	477	204
30	30	94	91	158	127	222	151	286	168	350	182	414	194	478	204
31	31	95	91	159	127	223	151	287	168	351	183	415	194	479	204
32	32	96	92	160	128	224	151	288	169	352	183	416	194	480	204
33	33	97	93	161	128	225	152	289	169	353	183	417	195	481	205
34	34	98	93	162	129	226	152	290	169	354	183	418	195	482	205
35	35	99	94	163	129	227	152	291	169	355	183	419	195	483	205
36	36	100	95	164	129	228	152	292	170	356	184	420	195	484	205
37	37	101	96	165	130	229	153	293	170	357	184	421	195	485	205
38	38	102	96	166	130	230	153	294	170	358	184	422	195	486	205
39	39	103	97	167	131	231	153	295	170	359	184	423	196	487	205
40	40	104	98	168	131	232	154	296	171	360	184	424	196	488	206
41	41	105	98	169	132	233	154	297	171	361	185	425	196	489	206
42	42	106	99	170	132	234	154	298	171	362	185	426	196	490	206
43	43	107	100	171	132	235	155	299	171	363	185	427	196	491	206
44	44	108	100	172	133	236	155	300	172	364	185	428	196	492	206
45	45	109	101	173	133	237	155	301	172	365	185	429	197	493	206
46	46	110	102	174	134	238	155	302	172	366	185	430	197	494	206
47	47	111	102	175	134	239	156	303	172	367	186	431	197	495	207
48	48	112	103	176	134	240	156	304	173	368	186	432	197	496	207
49	49	113	103	177	135	241	156	305	173	369	186	433	197	497	207
50	50	114	104	178.	135	242	157	306	173	370	186	434	197	498	207
51	51	115	105	179	136	243	157	307	173	371	186	435	198	499	207
52	52	116	105	180	136	244	157	308	173	372	187	436	198	500	207
53	53	117	106	181	136	245	157	309	174	373	187	437	198	501	207
54	54	118	106	182	137	246	158	310	174	374	187	438	198	502	208
55	55	119	107	183	137	247	158	311	174	375	187	439	198	503	208
56	56	120	108	184	137	248	158	312	174	376	187	440	198	504	208
57	57	121	108	185	138	249	159	313	175	377	188	441	198	505	208
58	58	122	109	186	138	250	159	314	175	378	188	442	199	506	208
59	59	123	109	187	139	251	159	315	175	379	188	443	199	507	208
60	60	124	110	188	139	252	159	316	175	380	188	444	199	508	208
61	61	125	110	189	139	253	160	317	175	381	188	445	199	509	208
62	62	126	111	190	140	254	160	318	176	382	188	446	199	510	209
63	63	127	112	191	140	255	160	319	176	383	189	447	199	511	209

Table 14-14. A-Law Table – Part 2

Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law
512	209	576	217	640	224	704	231	768	237	832	243	896	248	960	253
513	209	577	217	641	225	705	231	769	237	833	243	897	248	961	253
514	209	578	217	642	225	706	231	770	237	834	243	898	248	962	253
515	209	579	217	643	225	707	231	771	237	835	243	899	248	963	253
516	209	580	218	644	225	708	232	772	238	836	243	900	248	964	253

Table 14-14. A-Law Table – Part 2 (continued)

Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law
517	210	581	218	645	225	709	232	773	238	837	243	901	248	965	253
518	210	582	218	646	225	710	232	774	238	838	243	902	248	966	253
519	210	583	218	647	225	711	232	775	238	839	243	903	249	967	253
520	210	584	218	648	225	712	232	776	238	840	243	904	249	968	253
521	210	585	218	649	225	713	232	777	238	841	244	905	249	969	253
522	210	586	218	650	226	714	232	778	238	842	244	906	249	970	254
523	210	587	218	651	226	715	232	779	238	843	244	907	249	971	254
524	211	588	219	652	226	716	232	780	238	844	244	908	249	972	254
525	211	589	219	653	226	717	232	781	238	845	244	909	249	973	254
526	211	590	219	654	226	718	233	782	238	846	244	910	249	974	254
527	211	591	219	655	226	719	233	783	239	847	244	911	249	975	254
528	211	592	219	656	226	720	233	784	239	848	244	912	249	976	254
529	211	593	219	657	226	721	233	785	239	849	244	913	249	977	254
530	211	594	219	658	226	722	233	786	239	850	244	914	249	97.8	254
531	211	595	219	659	227	723	233	787	239	851	244	915	249	979	254
532	212	596	220	660	227	724	233	788	239	852	244	916	250	980	254
533	212	597	220	661	227	725.	233	789	239	853	245	917	250	981	254
534	212	598	220	662	227	726	233	790	239	854	245	918	250	982	254
535	212	599	220	663	227	727	233	791	239	855	245	919	250	983	254
536	212	600	220	664	227	728	233	792	239	856	245	920	250	984	255
537	212	601	220	665	227	729	234	793	239	857	245	921	250	985	255
538	212	602	220	666	227	730	234	794	240	858	245	922	250	986	255
539	212	603	220	667	227	731	234	795	240	859	245	923	250	987	255
540	213	604	220	668	227	732	234	796	240	860	245	924	250	988	255
541	213	605	221	669	228	733	234	797	240	861	245	925	250	989	255
542	213	606	221	670	228	734	234	798	240	862	245	926	250	990	255
543	213	607	221	671	228	735	234	799	240	863	245	927	250	991	255
544	213	608	221	672	228	736	234	800	240	864	245	928	250	992	255
545	213	609	221	673	228	737	234	801	240	865	246	929.	250	993	255
546	213	610	221	674	228	738	234	802	240	866	246	930	251	994	255
547	214	611	221	675	228	739	235	803	240	867	246	931	251	995	255
548	214	612	221	676	228	740	235	804	240	868	246	932	251	996	255
549	214	613	221	677	228	741	235	805	240	869	246	933	251	997	255
550	214	614	222	678	229	742	235	806	241	870	246	934	251	998	255
551	214	615	222	679	229	743	235	807	241	871	246	935	251	999	255
552	214	616	222	680	229	744	235	808	241	872	246	936	251	1000	255
553	214	617	222	681	229	745	235	809	241	873	246	937	251	1001	255
554	214	618	222	682	229	746	235	810	241	874	246	938	251	1002	255
555	215	619	222	683	229	747	235	811	241	875	246	939	251	1003	255
556	215	620	222	684	229	748	235	812	241	876	246	940	251	1004	255
557	215	621	222	685	229	749	235	813	241	877	246	941	251	1005	255
558	215	622	222	686	229	750	236	814	241	878	247	942	251	1006	255
559	215	623	223	687	229	751	236	815	241	879	247	943	252	1007	255
560	215	624	223	688	230	752	236	816	241	880	247	944	252	1008	255

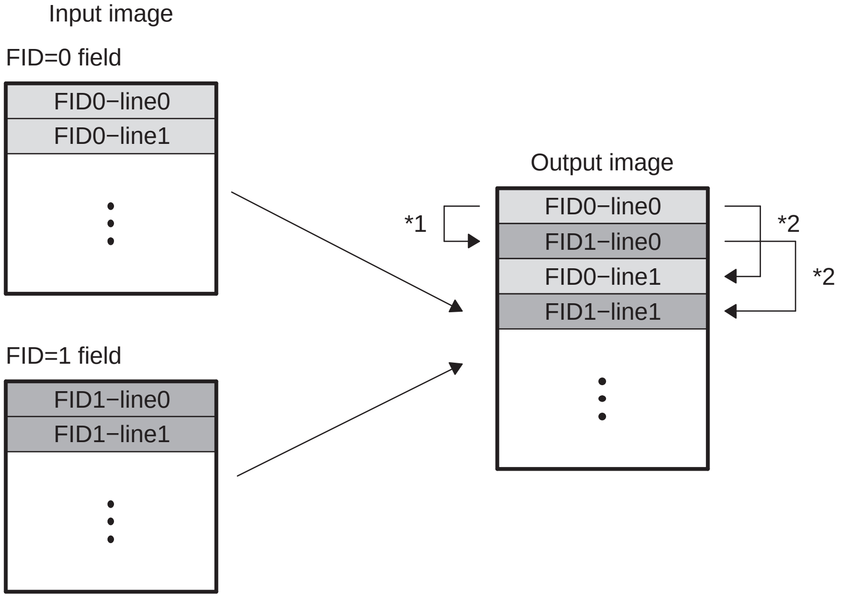
Table 14-14. A-Law Table – Part 2 (continued)

Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law
561	215	625	223	689	230	753	236	817	242	881	247	945	252	1009	255
562	215	626	223	690	230	754	236	818	242	882	247	946	252	1010	255
563	216	627	223	691	230	755	236	819	242	883	247	947	252	1011	255
564	216	628	223	692	230	756	236	820	242	884	247	948	252	1012	255
565	216	629	223	693	230	757	236	821	242	885	247	949	252	1013	255
566	216	630	223	694	230	758	236	822	242	886	247	950	252	1014	255
567	216	631	223	695	230	759	236	823	242	887	247	951	252	1015	255
568	216	632	224	696	230	760	236	824	242	888	247	952	252	1016	255
569	216	633	224	697	230	761	237	825	242	889	247	953	252	1017	255
570	216	634	224	698	231	762	237	826	242	890	247	954	252	1018	255
571	217	635	224	699	231	763	237	827	242	891	248	955	252	1019	255
572	217	636	224	700	231	764	237	828	242	892	248	956	252	1020	255
573	217	637	224	701	231	765	237	829	243	893	248	957	253	1021	255
574	217	638	224	702	231	766	237	830	243	894	248	958	253	1022	255
575	217	639	224	703	231	767	237	831	243	895	248	959	253	1023	255

14.3.2.1.4.4 Line Output Control

The final stage of the raw data mode is the line output control, which controls how the input sensor lines are written to external memory. The value of the ADR bit in the SDR_ADDR register defines the starting address where the frame should be written in external memory. The value of the LNOFST bit in the HSIZE_OFF register defines the distance between the beginning of output lines in bytes. Both the starting address and line offset must be aligned to 32-byte boundaries (i.e., either 16 or 32 pixels, depending on the PACK8 bit in the SYN_MODE register). Use the SDOFST register to define additional offsets, depending on the field ID and the even/odd line numbers. Defining additional offsets provides a means to de-interlace an interlaced, two-field input and also inverts an input image vertically. See Figure 14-11 and Figure 14-12 for example usage.

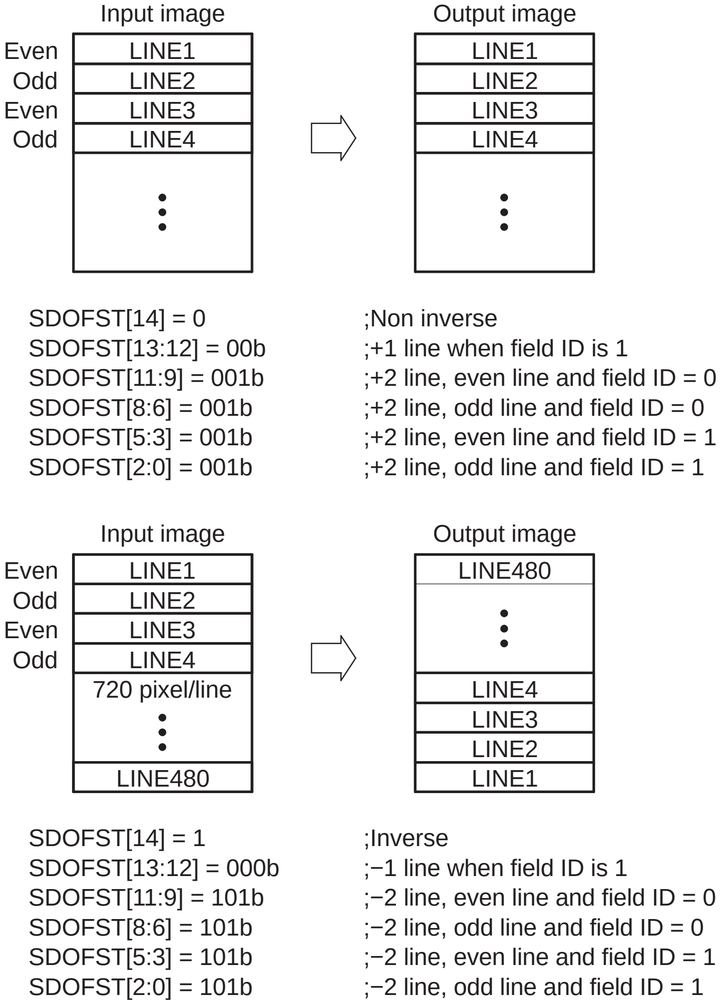
Figure 14-11. Image De-interfacing



*1 − SDOFST[13:12]=00b, +1 line for FID=1 lines

*2 − SDOFST[11:9]=SDOFST[8:6]=SDOFST[5:3]=SDOFST[2:0]=001b, +2 lines

Figure 14-12. Non-inversed vs Inversed Format



```
SDOFST[14] = 0            ;Non inverse
SDOFST[13:12] = 00b       ;+1 line when field ID is 1
SDOFST[11:9] = 001b       ;+2 line, even line and field ID = 0
SDOFST[8:6] = 001b        ;+2 line, odd line and field ID = 0
SDOFST[5:3] = 001b        ;+2 line, even line and field ID = 1
SDOFST[2:0] = 001b        ;+2 line, odd line and field ID = 1
```

```
SDOFST[14] = 1            ;Inverse
SDOFST[13:12] = 000b      ;−1 line when field ID is 1
SDOFST[11:9] = 101b       ;−2 line, even line and field ID = 0
SDOFST[8:6] = 101b        ;−2 line, odd line and field ID = 0
SDOFST[5:3] = 101b        ;−2 line, even line and field ID = 1
SDOFST[2:0] = 101b        ;−2 line, odd line and field ID = 1
```

14.3.2.1.4.5 Output Format

The pixel data format in external memory is shown in Table 14-16.

- If pixel data format is 8-bit, every 16-bit word in external memory stores two pixel data.
- If pixel data format is greater than 8-bit, every 16-bit word in external memory stores one pixel data, and the unused bits are MSB which are filled with 0.

Table 14-15. Storage Format in external memory for Raw Data Mode

	Upper Word		Lower Word	
	MSB (31)	LSB (16)	MSB (15)	LSB (0)
16-bit		Pixel1		Pixel0
15-bit	0	Pixel1	0	Pixel0
14-bit	0	Pixel1	0	Pixel0
13-bit	0	Pixel1	0	Pixel0
12-bit	0	Pixel1	0	Pixel0
11-bit	0	Pixel1	0	Pixel0
10-bit	0	Pixel1	0	Pixel0
9-bit	0	Pixel1	0	Pixel0
8-bit	0	Pixel1	0	Pixel0
8-bit pack	Pixel3	Pixel2	Pixel1	Pixel0

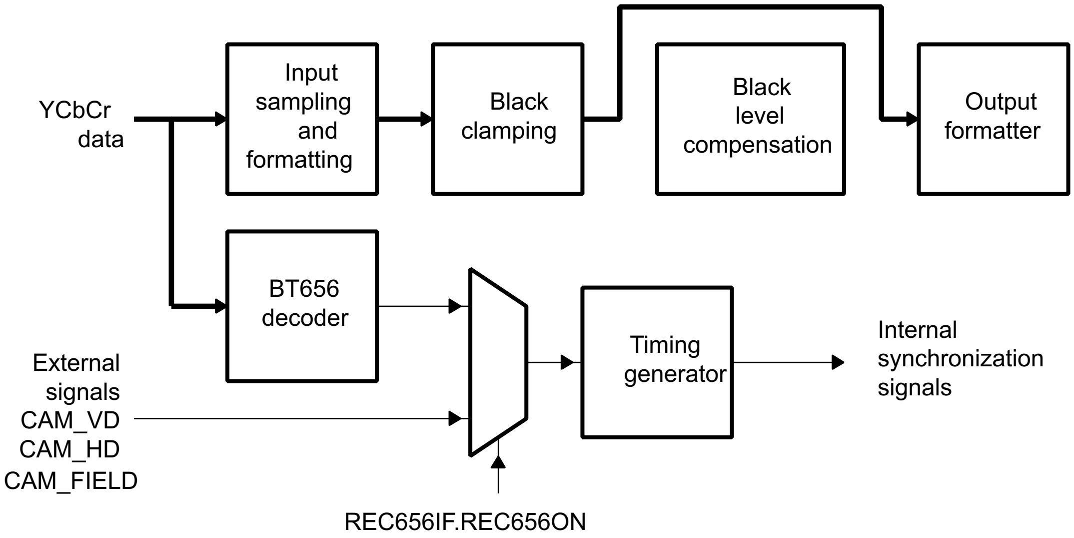
14.3.2.2 YCbCr and BT656 Modes

YCbCr mode and BT656 mode are similar in operation, as shown in Figure 14-13. The additional logic used in BT656 mode is the CCIR656 decoder, which extracts synchronization information from input YCbCr data and regenerates the corresponding timing for internal operation.

- YCbCr mode is enabled by setting field INPMODE in register SYN_MODE to 1 or 2 and clearing field REC656ON in register REC656IF.
- BT656 mode is enabled by setting REC656ON in register REC656IF to 1.

YCbCr mode typically has 8 bits per luma/choma sample while BT656 mode has 8 bits or 10 bits per luma/chroma sample.

Figure 14-13. Data Processing in YUV/BT656 Modes

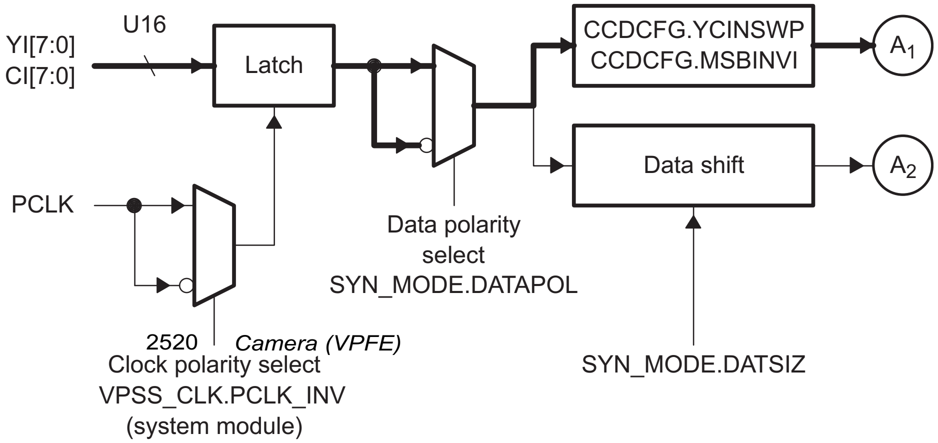


14.3.2.2.1 Input Sampling and Formatting

The data path of YCbCr/BT656 modes is shown in thicker lines in Figure 14-14 (i.e., A1). Data path A2 is applicable to raw data mode only.

- The pixel clock (CAM_PCLK) latches data.
- Pixel clock polarity can be either rising or falling edge and is set in VPSS clock control register (VPSS_CLK.PCLK_INV).
- DATAPOL bit in the SYN_MODE register affects the data representation.
- Bit YCINSWP in register CCDCFG can be used to swap the upper and lower portions of the 16-bit YCbCr data bus.
  - In 16-bit YCbCr mode, this swap bit determines which part of the bus luma and chroma samples occupy respectively.
  - In 8-bit mode, this swap bit determines which part of the bus is the effective 8-bit input. In other words, two external 8-bit YCbCr capture devices can be tied to VPFE module directly.
- Bit MSBINVI in register CCDCFG can be used to invert the MSB of the chroma signal.

Figure 14-14. CCD Controller

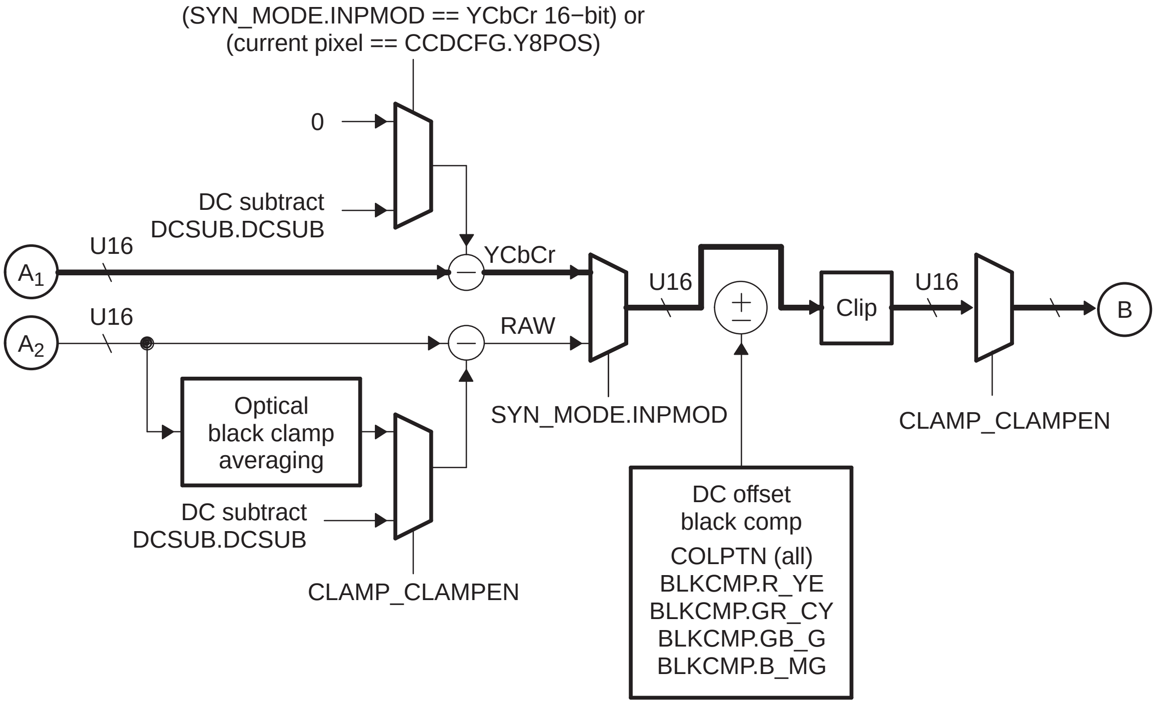


14.3.2.2.2 Black Clamping

The second step in BT.656/YCbCr processing is black clamping

Use the DCSUB bit in the DCSUB register to subtract a fixed value from the luma sample for YUV data. Set the subtraction value to zero to disable the operation. See Figure 14-15 for more details. You may notice that black compensation (BLKCMP) is not used in YCbCr and BT656 modes.

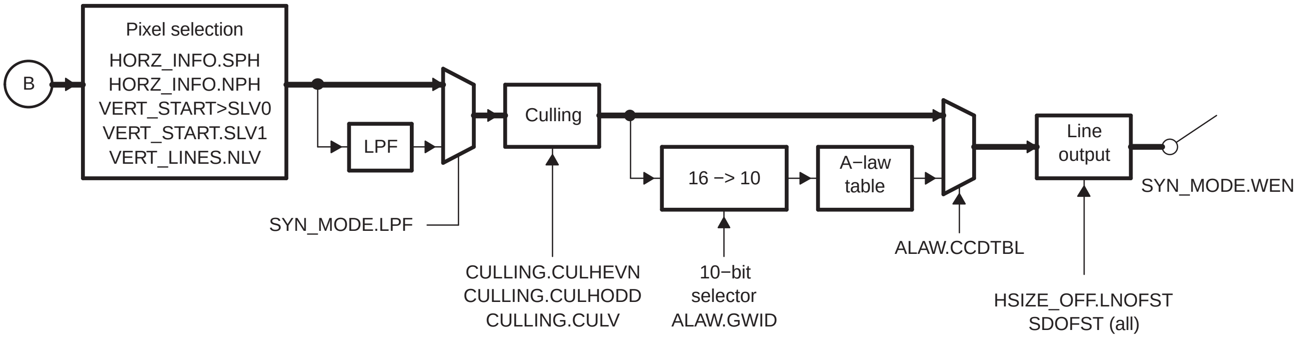
Figure 14-15. Black Clamping and Block Level Compensation



14.3.2.2.3 Output Formatter

The output formatter is the final stage of VPFE processing (as shown as thicker lines in Figure 14-16). Apply a framing selection to limit the processing area by the settings in the HORZ_INFO, VERT_START and VERT_LINES registers.

Figure 14-16. Output Formatter



- Use the LFP bit in the SYN_MODE register to disable the LPF by setting the LFP bit equal to 0.
- Culling can be used in YCbCr and BT656 modes; however, care must be taken to preserve the 422 output format.
- Do not use the A-Law transformation in YCbCr or BT.656 mode (the CCDTBL bit in the ALAW register = 0).

The pixel data in external memory is shown in Table 14-16.

Table 14-16. Storage Format in external memory for BT.656/YCbCr Modes

external memory Address	Upper word		Lower word	
	MSB (31)	LSB (16)	MSB (15)	LSB (0)
N	Y1	Cr0	Y0	Cb0
N + 1	Y3	Cr2	Y2	Cb2
N + 2	Y5	Cr4	Y4	Cb4

14.4 Programming Model

The following programming procedures should be followed to enable the VPFE controller to receive video or image capture data.

1. Power-up the VPFE using the Power, Reset, and Clock Management (PRCM) module.
2. Disable the VPFE controller.
3. Configure the VPFE registers to match the system requirements, including interrupt setup.
4. Enable the VPFE controller.

Once the VPFE controller is enabled, it starts to process input data continuously. No user intervention is necessary to re-start the process. As a result, the VPFE interrupt must be set up correctly in order to operate properly. Users can terminate its operation by disabling the VPFE controller.

The remaining sections describe the programming procedure in more detail.

14.4.1 Enabling and Disabling the VPFE Controller

The following steps describe how to enable and disable the VPFE controller:

- Clearing the ENABLE bit in the VPFE_PCR register disables the VPFE controller.
- Setting the ENABLE bit in the VPFE_PCR register enables the VPFE controller.
  - The VPFE controller should be enabled prior to data transmission from the external device to avoid data loss.

14.4.2 Configuring VPFE Registers

14.4.2.1 General Register Setup

Table 14-17 lists the minimum register fields that must be configured.

Table 14-17. Basic Configuration of VPFE Registers

Function	Register	Fields
External signal configuration (This includes signals CAM_VD, CAM_HD, CAM_FIELD, and CAM_WEN.)	SYN_MODE	VDHDEN VDPOL HDPOL FLDMODE FLDPOL EXWEN DATAPOL
	CCDCFG	VDLC
Input data mode	REC656	R656ON
	SYN_MODE	INPMOD
Color pattern	COLPTN	All
Black compensation	BLKCMP	All
Data path configuration	SYN_MODE	WEN

Table 14-18 describes additional configuration requirements when certain conditions are met.

Table 14-18. Conditional Configuration of VPFE Registers

Category	If...	Then program...
Interlace mode	Field FLDMODE in register SYN_MODE is 1	Field FIDMD in register CCDCFG
External WEN	Field EXWEN in register SYN_MODE is 1	Field WENLOG bit in register CCDCFG
REC656 input	Field R656ON in register REC656 is 1	Field ECCFVH in register REC656 Field BW656 in register CCDCFG
RAW input	Field INPMOD in register SYN_MODE is 0 and Field R656ON in register REC656 is 1	Field DATSIZ in register SYN_MODE Field CLAMPEN in register CLAMP
16-bit YCC input	Field INPMOD in register SYN_MODE is 1 and Field R656ON in register REC656 is 1	Field YCINSWP in register CCDCFG Field MSBINVI in register CCDCFG Register DCSUB
8-bit YCC input	Field INPMOD in register SYN_MODE is 2 and Field R656ON in register REC656 is 1	Field Y8POS in register CCDCFG
Optical black clamp enabled	Field CLAMPEN in register CLAMP is 1 and Field INPMOD in register SYN_MODE is 0	Field OBGAIN in register CLAMP Field OBST in register CLAMP Field OBSLN in register CLAMP Field OBSLEN in register CLAMP
Optical black clamp disabled	Field CLAMPEN in register CLAMP is 0 and Field INPMOD in register SYN_MODE is 0	Register DCSUB
Write to external memory	Field WEN in register SYN_MODE is 1	Field LPF in register SYN_MODE Field PACK8 in register SYN_MODE Field CCDTBL in register ALAW Field BSWD in register CCDCFG Register HORZ_INFO Register VERT_START Register VERT_LINES Register CULLING Register SDR_ADDR Register HSIZE_OFF Register SDOFST
A-law	Field CCDTBL in register ALAW is 1	Field GWID in register ALAW
Interrupt	Interrupts CCDC_VD0_INT and CCDC_VD1_INT are to be used	Register VDINT

14.4.2.2 Interrupts

The VPFE controller can generate three interrupts: CCDC_VD0_INT, CCDC_VD1_INT, and CCDC_VD2_INT.

NOTE: Enable the VDHDEN field in register SYN_MODE to receive any of these VPFE controller interrupts.

The CCDC_VD0_INT and CCDC_VD1_INT interrupts occur relative to the VD pulse, as shown in Figure 14-17 and Figure 14-18. Please note that field VDPOL in register SYN_MODE changes the trigger timing.

- If field VDPOL is 1, the VDINT0 and VDINT1 counters begin counting CAM_HD pulses from the falling edge of signal CAM_VD.
- If field VDPOL is 0, the VDINT0 and VDINT1 counters begin counting CAM_HD pulses from the rising edge of signal CAM_VD.

- Interrupts CCDC_VD0_INT and CCDC_VD1_INT occur after receiving the number of horizontal lines (CAM_HD pulse signals) set in the VDINT0 bit in the VDINT register and the VDINT1 bit in the VDINT register, respectively.

Figure 14-17. VDPOL is 0

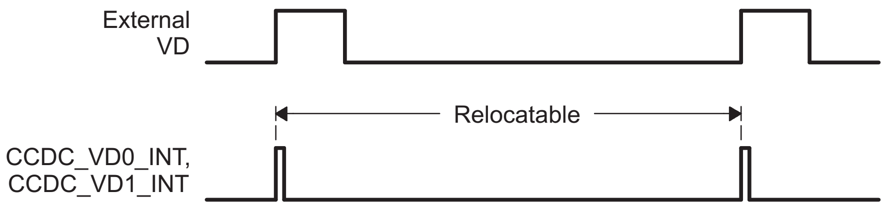
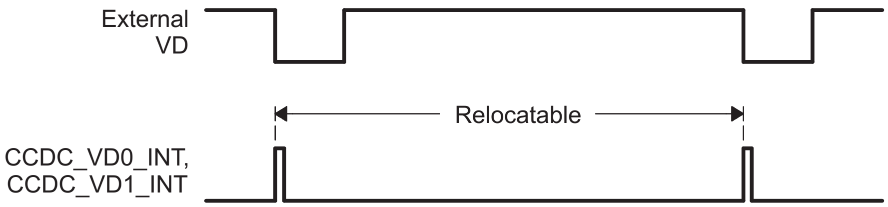
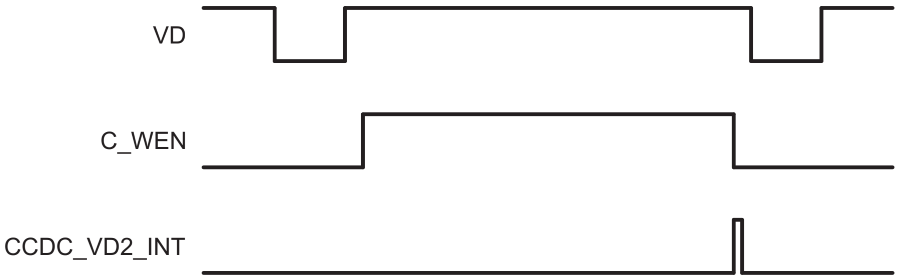


Figure 14-18. VDPOL is 1



The CCDC_VD2_INT interrupt always occurs at the falling edge of the CAM_WEN signal (via an external pin), as shown in Figure 14-19. There are no registers in the VPFE module to configure this interrupt.

Figure 14-19. CCDC_VD2_INT Interrupt



14.4.2.3 Status

The BUSY status bit in the VPFE_PCR register is set when the start of frame occurs (if the ENABLE bit in the VPFE_PCR register is 1 at that time). It automatically resets to 0 at the end of a frame.

14.4.2.4 CAM_VD latched Registers

The VDLC field in register CCDCFG affects the access of the following registers and register fields.

Registers
VPFE_PCR
HORZ_INFO
VERT_START
VERT_LINES
CULLING
HSIZE_OFF
SDOFST
SDR_ADDR
Register Fields
WEN in SYN_MODE
LPF in SYN_MODE

Registers
CLAMPEN in CLAMP
YCINSWP in CCDCFG

NOTE:

1. If the VDLC field is 0, changes to the above registers/register fields will take effect immediately.
2. If the VDLC field is 1, changes to the above registers/register fields will take effect at the start of a new frame (i.e., the changes are latched by the CAM_VD signal). Reads from these registers/register fields will return the most recent write value (even though these values may not take effect).

Be careful to avoid undesired effects when using the first approach.

14.4.2.5 Inter-frame Operations

As described in the previous section, the VPFE_PCR and SDR_ADDR registers will be latched by the CAM_VD signal if the VDLC field in the CCDCFG register is 0. This important feature provides a reliable way for programmers to enable/disable the VPFE module and/or modify the memory pointers in-between frames. For example, the VPFE interrupt service routine (ISR) can program the SDR_ADDR register to a new value before the end of the current frame. By doing so, the current frame is received without any interruption and the new external memory address (SDR_ADDR register) will be used for receiving the next frame.

14.4.3 VPFE Limitations

The major limitations of the VPFE module are summarized as follows:

- The CAM_PCLK (pixel clock) signal cannot be higher than 75 MHz.
- The SDR_ADDR and HSIZE_OFF registers should be programmed to values with 5 LBS bits as 0; namely, the memory space they point to should be on a 32-byte boundary.
- The COLPTN register should be set to 0 in YCbCr and BT.656 modes.
- The BLKCMP register should be set to 0 in YCbCr and BT.656 modes.
- Low-pass filter (LPF field in SYN_MODE register) should be disabled in YCbCr and BT.656 modes.
- A-LAW should be disabled in YCbCr and BT.656 modes.

14.5 Registers

14.5.1 VPFE Registers

Table 14-19 lists the memory-mapped registers for the VPFE. All register offset addresses not listed in Table 14-19 should be considered as reserved locations and the register contents should not be modified.

Table 14-19. VPFE Registers

Offset	Acronym	Register Name	Section
0h	VPFE_REVISION		Section 14.5.1.1
4h	VPFE_PCR		Section 14.5.1.2
8h	VPFE_SYNMODE		Section 14.5.1.3
Ch	VPFE_HD_VD_WID		Section 14.5.1.4
10h	VPFE_PIX_LINES		Section 14.5.1.5
14h	VPFE_HORZ_INFO		Section 14.5.1.6
18h	VPFE_VERT_START		Section 14.5.1.7
1Ch	VPFE_VERT_LINES		Section 14.5.1.8

Table 14-19. VPFE Registers (continued)

Offset	Acronym	Register Name	Section
20h	VPFE_CULLING		Section 14.5.1.9
24h	VPFE_HSIZE_OFF		Section 14.5.1.10
28h	VPFE_SDOFST		Section 14.5.1.11
2Ch	VPFE_SDR_ADDR		Section 14.5.1.12
30h	VPFE_CLAMP		Section 14.5.1.13
34h	VPFE_DCSUB		Section 14.5.1.14
38h	VPFE_COLPTN		Section 14.5.1.15
3Ch	VPFE_BLKCMP		Section 14.5.1.16
48h	VPFE_VDINT		Section 14.5.1.17
4Ch	VPFE_ALAW		Section 14.5.1.18
50h	VPFE_REC656IF		Section 14.5.1.19
54h	VPFE_CCDCFG		Section 14.5.1.20
98h	VPFE_DMA_CNTL		Section 14.5.1.21
104h	VPFE_SYSCONFIG		Section 14.5.1.22
108h	VPFE_CONFIG		Section 14.5.1.23
110h	VPFE_IRQ_EOI		Section 14.5.1.24
114h	VPFE_IRQ_STS_RAW		Section 14.5.1.25
118h	VPFE_IRQ_STS		Section 14.5.1.26
11Ch	VPFE_IRQ_EN_SET		Section 14.5.1.27
120h	VPFE_IRQ_EN_CLR		Section 14.5.1.28

14.5.1.1 VPFE_REVISION Register (Offset = 0h) [reset = 1000h]

VPFE_REVISION is shown in Figure 14-20 and described in Table 14-20.

Return to Summary Table.

IP Revision Identifier (X.Y.R)
Used by software to track features, bugs, and compatibility

Figure 14-20. VPFE_REVISION Register

31	30	29	28	27	26	25	24
SCHEME		RESERVED		FUNC			
R-0h		R-0h		R-0h			

23	22	21	20	19	18	17	16
FUNC							
R-0h							

15	14	13	12	11	10	9	8
R_RTL					X_MAJOR		
R-2h					R-0h		

7	6	5	4	3	2	1	0
CUSTOM		Y_MINOR					
R-0h		R-0h					

Table 14-20. VPFE_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	0h	
29-28	RESERVED	R	0h	
27-16	FUNC	R	0h	
15-11	R_RTL	R	2h	
10-8	X_MAJOR	R	0h	
7-6	CUSTOM	R	0h	
5-0	Y_MINOR	R	0h	

14.5.1.2 VPFE_PCR Register (Offset = 4h) [reset = 0h]

VPFE_PCR is shown in Figure 14-21 and described in Table 14-21.

Return to Summary Table.

Peripheral Control Register

Figure 14-21. VPFE_PCR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td>RESERVED</td><td>RESERVED</td><td>BUSY</td><td>EN</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R-0h</td><td>R-0h</td><td>R-0h</td><td>R/W-0h</td></tr>
</table>

Table 14-21. VPFE_PCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	RESERVED	R	0h	Reserved
2	RESERVED	R	0h	Reserved
1	BUSY	R	0h	VPFE busy bit 0h (R) = Not busy 1h (R) = Busy
0	EN	R/W	0h	This bit is latched by VD (start of frame) 0h (R/W) = Disable 1h (R/W) = Enable

14.5.1.3 VPFE_SYNMODE Register (Offset = 8h) [reset = 0h]

VPFE_SYNMODE is shown in Figure 14-22 and described in Table 14-22.

Return to Summary Table.

SYNC and Mode Set Register

Figure 14-22. VPFE_SYNMODE Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td>WEN</td><td>VDHDEN</td></tr>
<tr><td colspan="6">R-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>FLDSTAT</td><td>LPF</td><td colspan="2">INPMOD</td><td>PACK8</td><td colspan="3">DATSIZ</td></tr>
<tr><td>R/W-0h</td><td>R/W-0h</td><td colspan="2">R/W-0h</td><td>R/W-0h</td><td colspan="3">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>FLDMODE</td><td>DATAPOL</td><td>EXWEN</td><td>FLDPOL</td><td>HDPOL</td><td>VDPOL</td><td>FLDOUT</td><td>VDHDOUT</td></tr>
<tr><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 14-22. VPFE_SYNMODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	
17	WEN	R/W	0h	Data write enable. Controls whether or not input raw data is written to external memory. This bit is latched by VD. 0h (R/W) = Disable 1h (R/W) = Enable
16	VDHDEN	R/W	0h	VD/HD enable. Activates internal timing generator to synchronize with external VD/HD signals. This bit should be set to 1 when HD and VD signals are used at any time. 0h (R/W) = Disable 1h (R/W) = Enable
15	FLDSTAT	R/W	0h	Field status. Indicates the status of the current field when in interlaced mode. 0h (R/W) = Odd field 1h (R/W) = Even field
14	LPF	R/W	0h	3-tap low-pass (anti-aliasing) filter. This bit is latched by VD. 0h (R/W) = Off 1h (R/W) = On
13-12	INPMOD	R/W	0h	Setting data input mode 0h (R/W) = Raw data 1h (R/W) = YCbCr 16-bit 2h (R/W) = YCbCr 8-bit 3h (R/W) = Reserved
11	PACK8	R/W	0h	Pack to 8-bit/pixel (into external memory) 0h (R/W) = Normal (16 bits/pixel) 1h (R/W) = Pack to 8 bits/pixel

Table 14-22. VPFE_SYNMODE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-8	DATSIZ	R/W	0h	CCD data width is only valid when INPMOD is set to 0. 0h (R/W) = 16 bits 1h (R/W) = 15 bits 2h (R/W) = 14 bits 3h (R/W) = 13 bits 4h (R/W) = 12 bits 5h (R/W) = 11 bits 6h (R/W) = 10 bits 7h (R/W) = 8 bits
7	FLDMODE	R/W	0h	Sensor field mode 0h (R/W) = Non-interlaced (progressive) 1h (R/W) = Interlaced
6	DATAPOL	R/W	0h	Input data polarity 0h (R/W) = Normal (no charge) 1h (R/W) = One's complement
5	EXWEN	R/W	0h	External WEN selection. When set to 1 and when VDHDEN is set to 1, the WEN signal is used as the external memory write enable (to external memory). The data is stored to memory only when the external sync (HD and VD) signals are active. 0h (R/W) = Do not use external WEN (write enable) 1h (R/W) = Use external WEN (write enable)
4	FLDPOL	R/W	0h	Field indicator polarity 0h (R/W) = Positive 1h (R/W) = Negative
3	HDPOL	R/W	0h	HD sync polarity 0h (R/W) = Positive 1h (R/W) = Negative
2	VDPOL	R/W	0h	VD sync polarity 0h (R/W) = Positive 1h (R/W) = Negative
1	FLDOUT	R/W	0h	Field ID Direction 0h (R/W) = Input 1h (R/W) = Output
0	VDHDOUT	R/W	0h	VD/HD Sync Direction 0h (R/W) = Input 1h (R/W) = Output

14.5.1.4 VPFE_HD_VD_WID Register (Offset = Ch) [reset = 0h]

VPFE_HD_VD_WID is shown in Figure 14-23 and described in Table 14-23.

Return to Summary Table.

Figure 14-23. VPFE_HD_VD_WID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				HDW											
R-				R/W-											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				VDW											
R-				R/W-											

Table 14-23. VPFE_HD_VD_WID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R		
27-16	HDW	R/W		Width of HD sync pulse if output HDW+1 pixel clocks HDWIDTH is not used when HD is input, i.e when VDHDOUT in SYN_MODE register is cleared to '0' *This bit field is latched by VD
15-12	RESERVED	R		
11-0	VDW	R/W		Width of VD sync pulse if output VDW+1 lines VDWIDTH is not used when VD is input, i.e when VDHDOUT in SYN_MODE register is cleared to '0' *This bit field is latched by VD

14.5.1.5 VPFE_PIX_LINES Register (Offset = 10h) [reset = 0h]

VPFE_PIX_LINES is shown in Figure 14-24 and described in Table 14-24.

Return to Summary Table.

Number of pixels in a horizontal line and number of lines in a frame

Figure 14-24. VPFE_PIX_LINES Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPLN																HLPFR															
R/W-0h																R/W-0h															

Table 14-24. VPFE_PIX_LINES Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PPLN	R/W	0h	Pixels per line - number of pixel clock periods in one line HD period = PPLN+1 pixel clocks PPLN is not used when HD and VD are inputs, i.e when VDHDOUT in SYN_MODE register is cleared to '0' *This bit field is latched by VD
15-0	HLPFR	R/W	0h	Half lines per field or frame - sets number of half lines per frame or field VD period = (HLPFR+1)/2 lines HLPFR is not used when HD and VD are inputs, i.e when VDHDOUT in SYN_MODE register is cleared to '0' This tells the internal timing generator to generate the sufficient number of HD pulses in between two VD pulses. If the sensor is an interlaced sensor, say for example, with a total of 525 (or 526) lines, then this field should be set to 525 (or 526). This means that 525 (or 526) half lines are written for each field. If the sensor is progressive, then this register should be set to be twice the number of lines to be written. For example, if sensor outputs 1024 lines, this field should be set to 2048. Therefore, for interlaced sensors, this field should be set to the total number of lines. For progressive sensors, this field should be set to twice the total number of lines. *This bit field is latched by VD

14.5.1.6 VPFE_HORZ_INFO Register (Offset = 14h) [reset = 0h]

VPFE_HORZ_INFO is shown in Figure 14-25 and described in Table 14-25.

Return to Summary Table.

Horizontal Pixel Information Register

Figure 14-25. VPFE_HORZ_INFO Register

31	30	29	28	27	26	25	24
RESERVED	SPH						
R-0h	R/W-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SPH							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED	NPH						
R-0h	R/W-0h						
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NPH							
R/W-0h							

Table 14-25. VPFE_HORZ_INFO Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-16	SPH	R/W	0h	Start pixel, horizontal. The SPH sets the pixel clock position at which data output to external memory begins, measured from the start of HD. This bit field is latched by VD.
15	RESERVED	R	0h	
14-0	NPH	R/W	0h	Number of pixels, horizontal. NPH sets the number of horizontal pixels that is output to external memory = (NPH + 1) and 0xFFF0 (i.e., the number of horizontal output pixels truncates to multiples of 16). This bit field is latched by VD.

14.5.1.7 VPFE_VERT_START Register (Offset = 18h) [reset = 0h]

VPFE_VERT_START is shown in Figure 14-26 and described in Table 14-26.

Return to Summary Table.

Vertical Line - Settings for the Starting Pixel Register

Figure 14-26. VPFE_VERT_START Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>RESERVED</td><td colspan="7">SLV0</td></tr>
<tr><td>R-0h</td><td colspan="7">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">SLV0</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>RESERVED</td><td colspan="7">SLV1</td></tr>
<tr><td>R-0h</td><td colspan="7">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">SLV1</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 14-26. VPFE_VERT_START Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-16	SLV0	R/W	0h	Start line, vertical (field 0). SLV0 sets line at which data output to external memory will begin, measured from the start of VD. This bit field is latched by VD.
15	RESERVED	R	0h	
14-0	SLV1	R/W	0h	Start line, vertical (field 1). SLV1 sets line at which data output to external memory will begin, measured from the start of VD. For a progressive sensor this field is ignored. This bit field is latched by VD.

14.5.1.8 VPFE_VERT_LINES Register (Offset = 1Ch) [reset = 0h]

VPFE_VERT_LINES is shown in Figure 14-27 and described in Table 14-27.

Return to Summary Table.

Number of Vertical Lines Register

Figure 14-27. VPFE_VERT_LINES Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="17">RESERVED</td><td colspan="15">NLV</td></tr>
<tr><td colspan="17">R-0h</td><td colspan="15">R/W-0h</td></tr>
</table>

Table 14-27. VPFE_VERT_LINES Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14-0	NLV	R/W	0h	Number of lines, vertical. NLV sets the number of vertical lines that will be output to external memory. The number of lines output to external memory = (NLV + 1). This bit field is latched by VD.

14.5.1.9 VPFE_CULLING Register (Offset = 20h) [reset = FFFF00FFh]

VPFE_CULLING is shown in Figure 14-28 and described in Table 14-28.

Return to Summary Table.

Culling Information in Horizontal and Vertical Directions Register

Figure 14-28. VPFE_CULLING Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">CULHEVN</td><td colspan="8">CULHODD</td><td colspan="8">RESERVED</td><td colspan="8">CULV</td></tr>
<tr><td colspan="8">R/W-FFh</td><td colspan="8">R/W-FFh</td><td colspan="8">R-0h</td><td colspan="8">R/W-FFh</td></tr>
</table>

Table 14-28. VPFE_CULLING Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	CULHEVN	R/W	FFh	Horizontal Culling Pattern for Even Line, 8-bit mask. LSB is first pixel, MSB is 8th pixel, then pattern repeats. This bit field is latched by VD. 0h (R/W) = CULLING (deletion) 1h (R/W) = Retain (to be saved to external memory)
23-16	CULHODD	R/W	FFh	Horizontal Culling Pattern for Odd Line, 8-bit mask. LSB is first pixel, MSB is 8th pixel, then pattern repeats. This bit field is latched by VD. 0h (R/W) = CULLING (deletion) 1h (R/W) = Retain (to be saved to external memory)
15-8	RESERVED	R	0h	
7-0	CULV	R/W	FFh	Vertical Culling Pattern, 8-bit mask. LSB is first line, MSB is 8th line, then pattern repeats. This bit field is latched by VD. 0h (R/W) = CULLING (deletion) 1h (R/W) = Retain (to be saved to external memory)

14.5.1.10 VPFE_HSIZE_OFF Register (Offset = 24h) [reset = 0h]

VPFE_HSIZE_OFF is shown in Figure 14-29 and described in Table 14-29.

Return to Summary Table.

Horizontal Size Register

Figure 14-29. VPFE_HSIZE_OFF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LNOFST															
R-0h																R/W-0h															

Table 14-29. VPFE_HSIZE_OFF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	LNOFST	R/W	0h	Address offset for each line. LNOFST Sets offset for each output line in external memory Either 16 or 32 pixels depending on setting of PACK8. The 5 LSB are ignored, and a zero is returned when read the offset will be on a 32-byte boundary. For optimal performance in the system, the address offset should be on a 256-byte boundary. This bit field is latched by VD.

14.5.1.11 VPFE_SDOFST Register (Offset = 28h) [reset = 0h]

VPFE_SDOFST is shown in Figure 14-30 and described in Table 14-30.

Return to Summary Table.

External Memory Line Offset Register

Figure 14-30. VPFE_SDOFST Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>RESERVED</td><td>FIINV</td><td colspan="2">FOFST</td><td colspan="3">LOFTS0</td><td>LOFTS1</td></tr>
<tr><td>R-0h</td><td>R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="3">R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">LOFTS1</td><td colspan="3">LOFTS2</td><td colspan="3">LOFTS3</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="3">R/W-0h</td><td colspan="3">R/W-0h</td></tr>
</table>

Table 14-30. VPFE_SDOFST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14	FIINV	R/W	0h	Field identification signal inverse. This field is latched by VD. 0h (R/W) = Non inverse 1h (R/W) = Inverse
13-12	FOFST	R/W	0h	Line offset value of field ID = 1. This field is latched by VD. 0h (R/W) = +1 line 1h (R/W) = +2 lines 2h (R/W) = +3 lines 3h (R/W) = +4 lines
11-9	LOFTS0	R/W	0h	Line offset values of even line and even field ID = 0. This field is latched by VD. 0h (R/W) = +1 line 1h (R/W) = +2 lines 2h (R/W) = +3 lines 3h (R/W) = +4 lines 4h (R/W) = -1 line 5h (R/W) = -2 lines 6h (R/W) = -3 lines 7h (R/W) = -4 lines
8-6	LOFTS1	R/W	0h	Line offset values of odd line and even field ID = 0. This field is latched by VD. 0h (R/W) = +1 line 1h (R/W) = +2 lines 2h (R/W) = +3 lines 3h (R/W) = +4 lines 4h (R/W) = -1 line 5h (R/W) = -2 lines 6h (R/W) = -3 lines 7h (R/W) = -4 lines

Table 14-30. VPFE_SDOFST Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-3	LOFTS2	R/W	0h	Line offset values of even line and odd field ID = 1. This field is latched by VD. 0h (R/W) = +1 line 1h (R/W) = +2 lines 2h (R/W) = +3 lines 3h (R/W) = +4 lines 4h (R/W) = -1 line 5h (R/W) = -2 lines 6h (R/W) = -3 lines 7h (R/W) = -4 lines
2-0	LOFTS3	R/W	0h	Line offset values of odd line and odd field ID = 1. This field is latched by VD. 0h (R/W) = +1 line 1h (R/W) = +2 lines 2h (R/W) = +3 lines 3h (R/W) = +4 lines 4h (R/W) = -1 line 5h (R/W) = -2 lines 6h (R/W) = -3 lines 7h (R/W) = -4 lines

14.5.1.12 VPFE_SDR_ADDR Register (Offset = 2Ch) [reset = 0h]

VPFE_SDR_ADDR is shown in Figure 14-31 and described in Table 14-31.

Return to Summary Table.

External Memory Address Register

Figure 14-31. VPFE_SDR_ADDR Register

31–0
ADR
R/W-0h

Table 14-31. VPFE_SDR_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADR	R/W	0h	32-bit external memory starting address for VPFE output. This bit field is latched by VD. The address should be aligned on a 32-byte boundary. Therefore, the 5 LSB's are ignored. Furthermore, reading this register will always show the 5 LSB's as 0. For optimal performance in the system, the address should be on a 256-byte boundary.

14.5.1.13 VPFE_CLAMP Register (Offset = 30h) [reset = Fh]

VPFE_CLAMP is shown in Figure 14-32 and described in Table 14-32.

Return to Summary Table.

Optical Black Clamping Setting Register

Figure 14-32. VPFE_CLAMP Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>CLAMPEN</td><td colspan="3">OBSLEN</td><td colspan="3">OBSLN</td><td>OBST</td></tr>
<tr><td>R/W-0h</td><td colspan="3">R/W-0h</td><td colspan="3">R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">OBST</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="6">OBST</td><td colspan="2">RESERVED</td></tr>
<tr><td colspan="6">R/W-0h</td><td colspan="2">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">RESERVED</td><td colspan="5">OBGAIN</td></tr>
<tr><td colspan="3">R-0h</td><td colspan="5">R/W-Fh</td></tr>
</table>

Table 14-32. VPFE_CLAMP Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CLAMPEN	R/W	0h	Clamp enable. Enable or disable clamping of CCD data based on the calculated average of optical black samples. This bit is latched by VD. 0h (R/W) = Disable 1h (R/W) = Enable
30-28	OBSLEN	R/W	0h	Optical black sample length. Number of Optical Black Sample pixels per line to include in the average calculation 0h (R/W) = 1 pixels 1h (R/W) = 2 pixels 2h (R/W) = 4 pixels 3h (R/W) = 8 pixels 4h (R/W) = 16 pixels 5h (R/W) = Reserved 6h (R/W) = Reserved 7h (R/W) = Reserved
27-25	OBSLN	R/W	0h	Optical black sample lines. Number of Optical Black Sample lines to include in the average calculation 0h (R/W) = 1 lines 1h (R/W) = 2 lines 2h (R/W) = 4 lines 3h (R/W) = 8 lines 4h (R/W) = 16 lines 5h (R/W) = Reserved 6h (R/W) = Reserved 7h (R/W) = Reserved
24-10	OBST	R/W	0h	Start pixel of optical black samples. The start pixel position of optical black samples, specified from the start of HD in pixel clocks.
9-5	RESERVED	R	0h	

Table 14-32. VPFE_CLAMP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	OBGAIN	R/W	Fh	Gain to apply to the optical black average. Multiply the optical black average with the specified gain. 1Fh = 1 + 15/16 1Eh = 1 + 14/16 ... = ... 10h = 1 + 0/16 0Fh = 0 + 15/16 0Eh = 0 + 14/16 0Dh = 0 + 13/16 ... = ... 02h = 0 + 2/16 01h = 0 + 1/16 00h = 0 + 0/16

14.5.1.14 VPFE_DCSUB Register (Offset = 34h) [reset = 0h]

VPFE_DCSUB is shown in Figure 14-33 and described in Table 14-33.

Return to Summary Table.

DC Clamp Register

Figure 14-33. VPFE_DCSUB Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="18">RESERVED</td><td colspan="14">DCSUB</td></tr>
<tr><td colspan="18">R-0h</td><td colspan="14">R/W-0h</td></tr>
</table>

Table 14-33. VPFE_DCSUB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	
13-0	DCSUB	R/W	0h	DC level to subtract from CCD data. The DC value set here is subtracted from the CCD data when OBS clamping is disabled - CLAMP.CLAMPEN.

14.5.1.15 VPFE_COLPTN Register (Offset = 38h) [reset = 0h]

VPFE_COLPTN is shown in Figure 14-34 and described in Table 14-34.

Return to Summary Table.

CCD Color Pattern Register

Figure 14-34. VPFE_COLPTN Register

31-30	29-28	27-26	25-24
CP3LPC3	CP3LPC2	CP3LPC1	CP3LPC0
R/W-0h	R/W-0h	R/W-0h	R/W-0h
23-22	**21-20**	**19-18**	**17-16**
CP2LPC3	CP2LPC2	CP2LPC1	CP2LPC0
R/W-0h	R/W-0h	R/W-0h	R/W-0h
15-14	**13-12**	**11-10**	**9-8**
CP1LPC3	CP1LPC2	CP1LPC1	CP1LPC0
R/W-0h	R/W-0h	R/W-0h	R/W-0h
7-6	**5-4**	**3-2**	**1-0**
CP0LPC3	CP0LPC2	CP0LPC1	CP0LPC0
R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 14-34. VPFE_COLPTN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	CP3LPC3	R/W	0h	Color Pattern for 3rd Line, Pixel counter = 3 0h (R/W) = R/Ye 1h (R/W) = Gr/Cy 2h (R/W) = Gb/G 3h (R/W) = B/Mg
29-28	CP3LPC2	R/W	0h	Color Pattern for 3rd Line, Pixel counter = 2 0h (R/W) = R/Ye 1h (R/W) = Gr/Cy 2h (R/W) = Gb/G 3h (R/W) = B/Mg
27-26	CP3LPC1	R/W	0h	Color Pattern for 3rd Line, Pixel counter = 1 0h (R/W) = R/Ye 1h (R/W) = Gr/Cy 2h (R/W) = Gb/G 3h (R/W) = B/Mg
25-24	CP3LPC0	R/W	0h	Color Pattern for 3rd Line, Pixel counter = 0 0h (R/W) = R/Ye 1h (R/W) = Gr/Cy 2h (R/W) = Gb/G 3h (R/W) = B/Mg
23-22	CP2LPC3	R/W	0h	Color Pattern for 2nd Line, Pixel counter = 3 0h (R/W) = R/Ye 1h (R/W) = Gr/Cy 2h (R/W) = Gb/G 3h (R/W) = B/Mg
21-20	CP2LPC2	R/W	0h	Color Pattern for 2nd Line, Pixel counter = 2 0h (R/W) = R/Ye 1h (R/W) = Gr/Cy 2h (R/W) = Gb/G 3h (R/W) = B/Mg

Table 14-34. VPFE_COLPTN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-18	CP2LPC1	R/W	0h	Color Pattern for 2nd Line, Pixel counter = 1 0h (R/W) = R/Ye 1h (R/W) = Gr/Cy 2h (R/W) = Gb/G 3h (R/W) = B/Mg
17-16	CP2LPC0	R/W	0h	Color Pattern for 2nd Line, Pixel counter = 0 0h (R/W) = R/Ye 1h (R/W) = Gr/Cy 2h (R/W) = Gb/G 3h (R/W) = B/Mg
15-14	CP1LPC3	R/W	0h	Color Pattern for 1st Line, Pixel counter = 3 0h (R/W) = R/Ye 1h (R/W) = Gr/Cy 2h (R/W) = Gb/G 3h (R/W) = B/Mg
13-12	CP1LPC2	R/W	0h	Color Pattern for 1st Line, Pixel counter = 2 0h (R/W) = R/Ye 1h (R/W) = Gr/Cy 2h (R/W) = Gb/G 3h (R/W) = B/Mg
11-10	CP1LPC1	R/W	0h	Color Pattern for 1st Line, Pixel counter = 1 0h (R/W) = R/Ye 1h (R/W) = Gr/Cy 2h (R/W) = Gb/G 3h (R/W) = B/Mg
9-8	CP1LPC0	R/W	0h	Color Pattern for 1st Line, Pixel counter = 0 0h (R/W) = R/Ye 1h (R/W) = Gr/Cy 2h (R/W) = Gb/G 3h (R/W) = B/Mg
7-6	CP0LPC3	R/W	0h	Color Pattern for 0th Line, Pixel counter = 3 0h (R/W) = R/Ye 1h (R/W) = Gr/Cy 2h (R/W) = Gb/G 3h (R/W) = B/Mg
5-4	CP0LPC2	R/W	0h	Color Pattern for 0th Line, Pixel counter = 2 0h (R/W) = R/Ye 1h (R/W) = Gr/Cy 2h (R/W) = Gb/G 3h (R/W) = B/Mg
3-2	CP0LPC1	R/W	0h	Color Pattern for 0th Line, Pixel counter = 1 0h (R/W) = R/Ye 1h (R/W) = Gr/Cy 2h (R/W) = Gb/G 3h (R/W) = B/Mg
1-0	CP0LPC0	R/W	0h	Color Pattern for 0th Line, Pixel counter = 0 0h (R/W) = R/Ye 1h (R/W) = Gr/Cy 2h (R/W) = Gb/G 3h (R/W) = B/Mg

14.5.1.16 VPFE_BLKCMP Register (Offset = 3Ch) [reset = 0h]

VPFE_BLKCMP is shown in Figure 14-35 and described in Table 14-35.

Return to Summary Table.

Black Compensation Register

Figure 14-35. VPFE_BLKCMP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RYE								GRCY								GBG								BMG							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 14-35. VPFE_BLKCMP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RYE	R/W	0h	Black level compensation for R/Ye pixels (-128:+127). 2's complement, MSB is sign bit.
23-16	GRCY	R/W	0h	Black level compensation for Gr/Cy pixels (-128:+127). 2's complement, MSB is sign bit.
15-8	GBG	R/W	0h	Black level compensation for Gb/G pixels (-128:+127). 2's complement, MSB is sign bit.
7-0	BMG	R/W	0h	Black level compensation for B/Mg pixels (-128:+127). 2's complement, MSB is sign bit.

14.5.1.17 VPFE_VDINT Register (Offset = 48h) [reset = 0h]

VPFE_VDINT is shown in Figure 14-36 and described in Table 14-36.

Return to Summary Table.

VPFE Interrupt Control Register

Figure 14-36. VPFE_VDINT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>RESERVED</td><td colspan="7">VDINT0</td></tr>
<tr><td>R-0h</td><td colspan="7">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">VDINT0</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>RESERVED</td><td colspan="7">VDINT1</td></tr>
<tr><td>R-0h</td><td colspan="7">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">VDINT1</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 14-36. VPFE_VDINT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-16	VDINT0	R/W	0h	CCDC_VD0_INT interrupt timing. Specify VDINT0 in units of horizontal lines from the start of VD pulse. Resulting value is VDINT0+1. Note that if the rising edge (or falling edge if programmed) of the HD lines up with the rising edge (or falling edge if programmed) of VD, the 1st HD is not counted.
15	RESERVED	R	0h	
14-0	VDINT1	R/W	0h	CCDC_VD1_INT interrupt timing. Specify VDINT1 in units of horizontal lines from the start of VD pulse. Resulting value is VDINT1+1. Note that if the rising edge (or falling edge if programmed) of the HD lines up with the rising edge (or falling edge if programmed) of VD, the 1st HD is not counted.

14.5.1.18 VPFE_ALAW Register (Offset = 4Ch) [reset = 4h]

VPFE_ALAW is shown in Figure 14-37 and described in Table 14-37.

Return to Summary Table.

ALAW Configuration Register

Figure 14-37. VPFE_ALAW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td>CCDTBL</td><td colspan="3">GWDI</td></tr>
<tr><td colspan="4">R-0h</td><td>R/W-0h</td><td colspan="3">R/W-4h</td></tr>
</table>

Table 14-37. VPFE_ALAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	CCDTBL	R/W	0h	Apply Gamma (A-LAW) to VPFE data saved to external memory 0h (R/W) = Disable 1h (R/W) = Enable
2-0	GWDI	R/W	4h	A-law Width Input (A-LAW table) 0h (R/W) = Bits 15-6 1h (R/W) = Bits 14-5 2h (R/W) = Bits 13-4 3h (R/W) = Bits 12-3 4h (R/W) = Bits 11-2 5h (R/W) = Bits 10-1 6h (R/W) = Bits 9-0

14.5.1.19 VPFE_REC656IF Register (Offset = 50h) [reset = 0h]

VPFE_REC656IF is shown in Figure 14-38 and described in Table 14-38.

Return to Summary Table.

REC656IF Configuration Register

Figure 14-38. VPFE_REC656IF Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						ECCFVH	R656ON
R-0h						R/W-0h	R/W-0h

Table 14-38. VPFE_REC656IF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	ECCFVH	R/W	0h	FVH error correction enable 0h (R/W) = Disable 1h (R/W) = Enable
0	R656ON	R/W	0h	REC656 interface enable 0h (R/W) = Disable 1h (R/W) = Enable

14.5.1.20 VPFE_CCDCFG Register (Offset = 54h) [reset = 0h]

VPFE_CCDCFG is shown in Figure 14-39 and described in Table 14-39.

Return to Summary Table.

CCD Configuration Register

Figure 14-39. VPFE_CCDCFG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
VDLC	MSBINVO	MSBINVI	BSWD	Y8POS	RESERVED		WENLOG
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h
7	6	5	4	3	2	1	0
RESERVED		BW656	YCINSWP	RESERVED	YCOUTSWP	RESERVED	
R-0h		R/W-0h	R/W-0h	R-0h	R/W-0h	R-0h	

Table 14-39. VPFE_CCDCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	VDLC	R/W	0h	Enable latching function registers on internal VSYNC. If this bit is set, all the register fields that are VSYNC latched will take on new value immediately. Care should be taken not to alter fields that can cause undesired behavior to the output data. 0h (R/W) = Latched on VSYNC 1h (R/W) = Not latched on VSYNC
14	MSBINVO	R/W	0h	MSB of Chroma signal output inverted 0h (R/W) = Normal 1h (R/W) = MSB inverted
13	MSBINVI	R/W	0h	MSB of Chroma input signal stored to SDRAM inverted 0h (R/W) = normal 1h (R/W) = MSB inverted
12	BSWD	R/W	0h	Byte Swap Data stored to SDRAM 0h (R/W) = normal 1h (R/W) = Swap Bytes
11	Y8POS	R/W	0h	Location of Y signal when YCbCr 8bit data is input 0h (R/W) = even pixel 1h (R/W) = odd pixel
10-9	RESERVED	R	0h	
8	WENLOG	R/W	0h	Specifies CCD valid area 0h (R/W) = Internal valid signal & WEN signal is ANDed logically 1h (R/W) = Internal valid signal & WEN signal is Ored logically
7-6	RESERVED	R	0h	
5	BW656	R/W	0h	The data width in CCIR656 input mode 0h (R/W) = 8bits 1h (R/W) = 10bits
4	YCINSWP	R/W	0h	Y input (YIN [7:0]) and C input (CIN [7:0]) are swapped

Table 14-39. VPFE_CCDCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	RESERVED	R	0h	
2	YCOUTSWP	R/W	0h	Y output (YOUT [7:0]) and C output (COUT [7:0]) are swapped
1-0	RESERVED	R	0h	

14.5.1.21 VPFE_DMA_CNTL Register (Offset = 98h) [reset = 0h]

VPFE_DMA_CNTL is shown in Figure 14-40 and described in Table 14-40.

Return to Summary Table.

DMA Status and Control

Figure 14-40. VPFE_DMA_CNTL Register

31	30	29	28	27	26	25	24
OVERFLOW	RESERVED						
R/W-0h	R-0h						
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					PRIORITY		
R-0h					R/W-0h		

Table 14-40. VPFE_DMA_CNTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	OVERFLOW	R/W	0h	DMA Overflow Flag Flag bit that is set when data is dropped due to a delay in writing data out the DMA interface. This bit remains set until a 1 is written by software. 0h (R/W) = No overflow has occurred 1h (R/W) = Overflow has occurred
30-3	RESERVED	R	0h	
2-0	PRIORITY	R/W	0h	Sets the priority that all command should be sent with on the DMA bus. This register should only be modified when the Module is inactive. A value of 0 is the highest priority while a value of 0x7 would be the lowest priority

14.5.1.22 VPFE_SYSCONFIG Register (Offset = 104h) [reset = 28h]

VPFE_SYSCONFIG is shown in Figure 14-41 and described in Table 14-41.

Return to Summary Table.

Clock management configuration

Figure 14-41. VPFE_SYSCONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		STANDBYMODE		IDLEMODE		RESERVED	
R-0h		R/W-2h		R/W-2h		R-0h	

Table 14-41. VPFE_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5-4	STANDBYMODE	R/W	2h	Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of STANDBY state 0h (R/W) = Force-standby mode: local initiator is unconditionally placed in standby state. Backup mode, for debug only. 1h (R/W) = No-standby mode: local initiator is unconditionally placed out of standby state. Backup mode, for debug only. 2h (R/W) = Smart-standby mode: local initiator standby status depends on local conditions, i.e. the module's functional requirement from the initiator. IP module shall not generate (initiator-related) wakeup events 3h (R/W) = Smart-Standby wakeup-capable mode: local initiator standby status depends on local conditions, i.e. the module's functional requirement from the initiator. IP module may generate (master-related) wakeup events when in standby state. Mode is only relevant if the appropriate IP module "mwakeup" output is implemented

Table 14-41. VPFE_SYSCONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	IDLEMODE	R/W	2h	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state 0h (R/W) = Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements. Backup mode, for debug only. 1h (R/W) = No-idle mode: local target never enters idle state. Backup mode, for debug only. 2h (R/W) = Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wakeup events. 3h (R/W) = Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state. Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented.
1-0	RESERVED	R	0h	

14.5.1.23 VPFE_CONFIG Register (Offset = 108h) [reset = 0h]

VPFE_CONFIG is shown in Figure 14-42 and described in Table 14-42.

Return to Summary Table.

Module configuration register

Figure 14-42. VPFE_CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					VPFE_ST	VPFE_EN	PCLK_INV
R-0h					R/W-0h	R/W-0h	R/W-0h

Table 14-42. VPFE_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	VPFE_ST	R/W	0h	VPFE Master OCP interface Status 0h (R/W) = OCP Master Interface is active 1h (R/W) = OCP Master Interface is in Standby mode
1	VPFE_EN	R/W	0h	VPFE Master OCP interface enable. Software can has to use this bit to enable/disable the VPFE master OCP interface. When the master OCP interface is disabled, it is placed in Standby mode. Standby mode can also be entered by using the right setting on the STANDBYMODE field in the SYSCONFIG register 0h (R/W) = Disable VPFE Master OCP interface 1h (R/W) = Enable VPFE Master OCP Interface
0	PCLK_INV	R/W	0h	Pixel clock inversion enable 0h (R/W) = pixel clock is not inverted, data is sampled on the rising edge of the clock 1h (R/W) = pixel clock is inverted, data is sampled on the falling edge of the clock

14.5.1.24 VPFE_IRQ_EOI Register (Offset = 110h) [reset = 0h]

VPFE_IRQ_EOI is shown in Figure 14-43 and described in Table 14-43.

Return to Summary Table.

Module EOI register

Figure 14-43. VPFE_IRQ_EOI Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="15">RESERVED</td><td>EOI</td></tr>
<tr><td colspan="15">R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 14-43. VPFE_IRQ_EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	EOI	R/W	0h	EOI for VPFE. This register allows software to acknowledge the completion of an interrupt. When this register is written, an eoi_write signal is generated internal to the module and another interrupt will be triggered if the interrupt sources are still present. The register will clear itself one cycle after it is written

14.5.1.25 VPFE_IRQ_STS_RAW Register (Offset = 114h) [reset = 0h]

VPFE_IRQ_STS_RAW is shown in Figure 14-44 and described in Table 14-44.

Return to Summary Table.

Interrupt raw status register

Figure 14-44. VPFE_IRQ_STS_RAW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">RESERVED</td><td>VD2_INT_RAW</td><td>VD1_INT_RAW</td><td>VD0_INT_RAW</td></tr>
<tr><td colspan="5">R-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 14-44. VPFE_IRQ_STS_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	VD2_INT_RAW	R/W	0h	CCDC VD2 interrupt status raw value - A read value of ‘1’ from this register indicates that the VD2 interrupt status is 1. - When the read value is ‘0’, software can write the value to ‘1’ to set the interrupt - Writing a ‘0’ to this bit has no effect.
1	VD1_INT_RAW	R/W	0h	CCDC VD1 interrupt status raw value - A read value of ‘1’ from this register indicates that the VD1 interrupt status is 1. - When the read value is ‘0’, software can write the value to ‘1’ to set the interrupt - Writing a ‘0’ to this bit has no effect
0	VD0_INT_RAW	R/W	0h	CCDC VD0 interrupt status raw value - A read value of ‘1’ from this register indicates that the VD0 interrupt status is 1. - When the read value is ‘0’, software can write the value to ‘1’ to set the interrupt - Writing a ‘0’ to this bit has no effect.

14.5.1.26 VPFE_IRQ_STS Register (Offset = 118h) [reset = 0h]

VPFE_IRQ_STS is shown in Figure 14-45 and described in Table 14-45.

Return to Summary Table.

Interrupt status register

Figure 14-45. VPFE_IRQ_STS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					VD2_INT	VD1_INT	VD0_INT
R-0h					R/W-0h	R/W-0h	R/W-0h

Table 14-45. VPFE_IRQ_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	VD2_INT	R/W	0h	CCDC VD2 interrupt status value - A read value of ‘1’ from this register indicates that the VD2 interrupt status is 1., if it is enabled - When the read value is ‘1’, software can write a ‘1’ to clear the interrupt. - Writing a ‘0’ to this bit has no effect.
1	VD1_INT	R/W	0h	CCDC VD1 interrupt status value - A read value of ‘1’ from this register indicates that the VD2 interrupt status is 1., if it is enabled - When the read value is ‘1’, software can write a ‘1’ to clear the interrupt. - Writing a ‘0’ to this bit has no effect
0	VD0_INT	R/W	0h	CCDC VD0 interrupt status value - A read value of ‘1’ from this register indicates that the VD2 interrupt status is 1., if it is enabled - When the read value is ‘1’, software can write a ‘1’ to clear the interrupt. - Writing a ‘0’ to this bit has no effect

14.5.1.27 VPFE_IRQ_EN_SET Register (Offset = 11Ch) [reset = 0h]

VPFE_IRQ_EN_SET is shown in Figure 14-46 and described in Table 14-46.

Return to Summary Table.

Interrupt enable set

Figure 14-46. VPFE_IRQ_EN_SET Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">RESERVED</td><td>VD2_INT_EN</td><td>VD1_INT_EN</td><td>VD0_INT_EN</td></tr>
<tr><td colspan="5">R-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 14-46. VPFE_IRQ_EN_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	VD2_INT_EN	R/W	0h	CCDC VD2 interrupt enable - Write ‘1’ to enable this interrupt - Write ‘0’ has no effect - Read ‘1’ indicates interrupt is enabled - Read ‘0’ indicates interrupt is not enabled.
1	VD1_INT_EN	R/W	0h	CCDC VD1 interrupt enable - Write ‘1’ to enable this interrupt - Write ‘0’ has no effect - Read ‘1’ indicates interrupt is enabled - Read ‘0’ indicates interrupt is not enabled
0	VD0_INT_EN	R/W	0h	CCDC VD0 interrupt enable - Write ‘1’ to enable this interrupt - Write ‘0’ has no effect - Read ‘1’ indicates interrupt is enabled - Read ‘0’ indicates interrupt is not enabled

14.5.1.28 VPFE_IRQ_EN_CLR Register (Offset = 120h) [reset = 0h]

VPFE_IRQ_EN_CLR is shown in Figure 14-47 and described in Table 14-47.

Return to Summary Table.

Interrupt enable clear

Figure 14-47. VPFE_IRQ_EN_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					VD2_INT_DIS	VD1_INT_DIS	VD0_INT_DIS
R-0h					R/W-0h	R/W-0h	R/W-0h

Table 14-47. VPFE_IRQ_EN_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	VD2_INT_DIS	R/W	0h	CCDC VD2 interrupt disable - Write ‘1’ to disable this interrupt - Write ‘0’ has no effect - Read ‘1’ indicates interrupt is enabled - Read ‘0’ indicates interrupt is not enabled
1	VD1_INT_DIS	R/W	0h	CCDC VD1 interrupt disable - Write ‘1’ to disable this interrupt - Write ‘0’ has no effect - Read ‘1’ indicates interrupt is enabled - Read ‘0’ indicates interrupt is not enabled
0	VD0_INT_DIS	R/W	0h	CCDC VD0 interrupt disable - Write ‘1’ to disable this interrupt - Write ‘0’ has no effect - Read ‘1’ indicates interrupt is enabled - Read ‘0’ indicates interrupt is not enabled

Chapter 15

Ethernet Subsystem

This chapter describes the ethernet subsystem of the device.

NOTE: The CPSW is not supported for the AMIC120.

Topic **Page**

15.1 Introduction

Described in the following sections is the Layer 2 3-port switch (3PSW) Ethernet subsystem. The 3-port switch gigabit ethernet subsystem provides ethernet packet communication and can be configured as an ethernet switch. It provides the gigabit media independent interface (GMII),reduced gigabit media independent interface (RGMII), reduced media independent interface (RMII), the management data input output (MDIO) for physical layer device (PHY) management.

As a Layer 2 switch, this device is capable of supporting all higher layers and various protocols, such as IPV4, IPV6, and 802.3x, in software.

15.1.1 Features

The general features of the ethernet switch subsystem are:

- Two 10/100/1000 Ethernet ports with GMII, RMII and RGMII interfaces
- Wire rate switching (802.1d)
- Non Blocking switch fabric
- Flexible logical FIFO based packet buffer structure
- Four priority level QOS support (802.1p)
- CPPI 3.1 compliant DMA controllers
- Support for IEEE 1588v2 Clock Synchronization (2008 Annex D, E, and F)
  - Timing FIFO and time stamping logic inside the SS
- Device Level Ring (DLR) Support
- Address Lookup Engine
  - 1024 addresses plus VLANs
  - Wire rate lookup
  - VLAN support
  - Host controlled time-based aging
  - Spanning tree support
  - L2 address lock and L2 filtering support
  - MAC authentication (802.1x)
  - Receive or destination based Multicast and Broadcast limits
  - MAC address blocking
  - Source port locking
  - OUI host accept/deny feature
- Flow Control Support (802.3x)
- EtherStats and 802.3Stats RMON statistics gathering (shared)
- Support for external packet dropping engine
- CPGMAC_SL transmit to CPGMAC_SL receive Loopback mode (digital loopback) supported
- CPGMAC_SL receive to CPGMAC_SL transmit Loobback mode (FIFO loopback) supported
- Maximum frame size 2016 bytes (2020 with VLAN)
- 8k (2048 x 32) internal CPPI buffer descriptor memory
- MDIO module for PHY Management
- Programmable interrupt control with selected interrupt pacing
- Emulation Support.
- Programmable transmit Inter-Packet Gap (IPG)
- Reset isolation

15.1.2 Unsupported Features

There are 18 level interrupts from the CPGMAC module and 2 (used) level interrupts from the MDIO module. The CPSW_3GSS includes an interrupt combiner/pacer to combine these interrupts together to produce 4 interrupt outputs (per processing core). This device does not split processing among multiple cores but allows servicing of the Core0 interrupts by the Cortex-A9 or the PRU-ICSS.

The unsupported CPGMAC features in the device are shown in the following table.

Table 15-1. Unsupported CPGMAC Features

Feature	Reason
GMII	Only 4 Rx/Tx data pins are pinned out for each port. The device supports MII (on GMII interface), RGMII, and RMII interfaces only
Phy link status	The MLINK inputs are not pinned out. Phy link status outputs can be connected to device GPIOs.
Internal Delay mode for RGMII	RGMII Internal Delay mode is not supported.

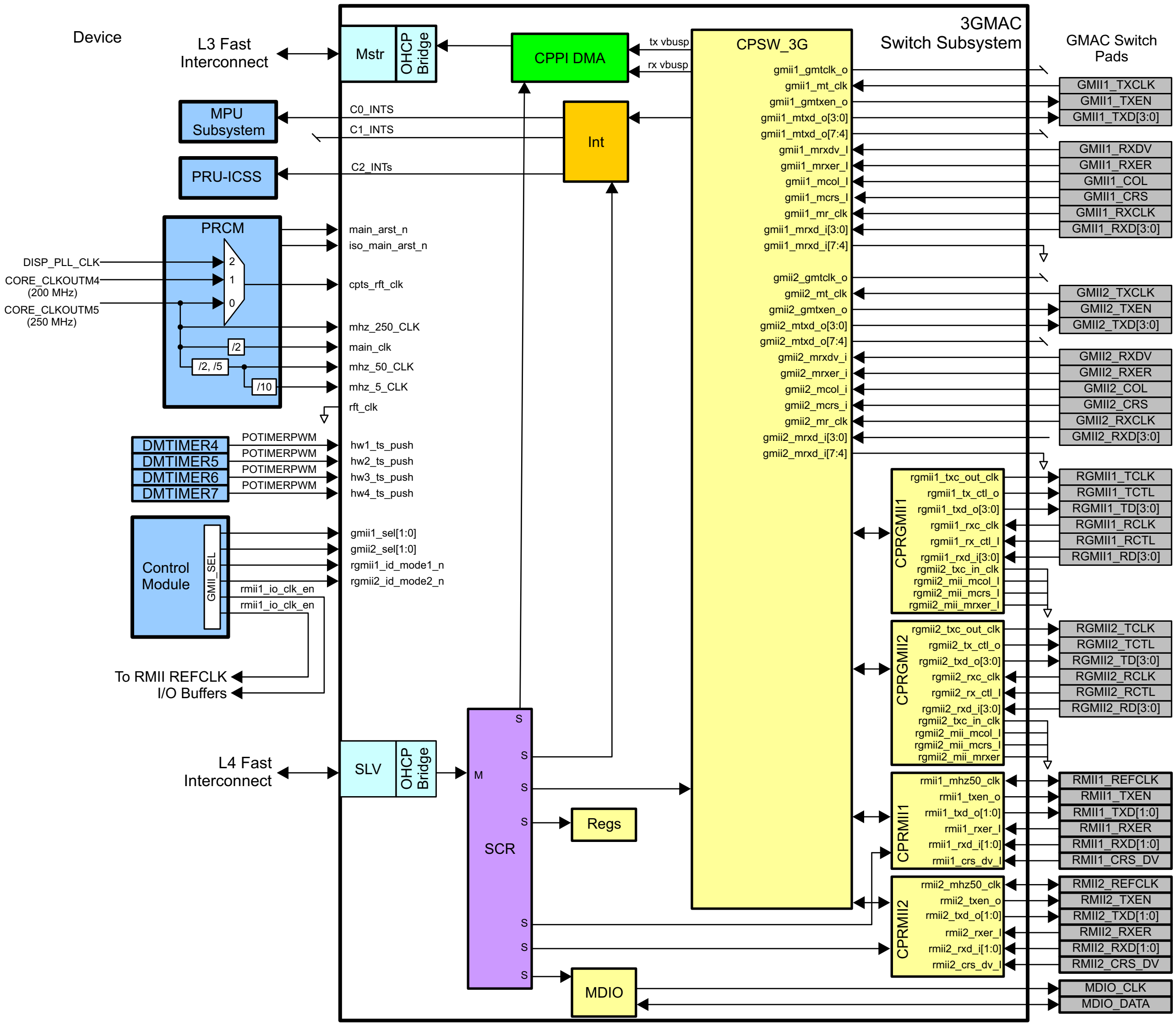
15.2 Integration

This device includes a single instantiation of the three-port Gigabit Ethernet Switch Subsystem (CPSW_3GSS_RG). The switch provides 2 external ethernet ports (ports 1 and 2) and an internal CPPI interface port (port 0) with IEEE 1588v2 and 802.1ae support. The subsystem consists of:

- One instance of the 3-port Gigabit switch CPSW-3G, which contains:
  - 2 CPGMAC_SL 10/100/1000 ethernet port modules with GMII interface
- Two RGMII interface modules
- Two RMII interface modules
- One MDIO interface module
- One Interrupt Controller module
- Local CPPI memory of size 8K Bytes

The integration of the Ethernet Switch is shown in Figure 15-1

Figure 15-1. Ethernet Switch Integration



15.2.1 Ethernet Switch Connectivity Attributes

The general connectivity attributes for the Ethernet Switch module are shown in Table 15-2.

Table 15-2. Ethernet Switch Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_CPSW_125MHZ_GCLK (Main) PD_PER_CPSW_250MHZ_GCLK (MHZ_250_CLK) PD_PER_CPSW_50MHZ_GCLK (MHZ_50_CLK) PD_PER_CPSW_5MHZ_GCLK (MHZ_5_CLK) PD_PER_CPSW_CPTS_RFT_CLK (CPTS_RFT_CLK)
Reset Signals	CPSW_MAIN_ARST_N CPSW_ISO_MAIN_ARST_N
Idle/Wakeup Signals	Smart Idle Standby
Interrupt Requests	Three sets of 4 Interrupts RX_THRESH (3PGSWRXTHR0) – Receive Threshold interrupt (nonpaced) RX (3PGSWRXINT0) – Receive interrupt (paced) TX (3PGSWTXINT0) – Transmit interrupt (paced) Misc (3PGSWMISC0) – Other interrupts The Subsystem contains 3 sets of interrupts—C0, C1, and C2—to allow for split core processing of packets. On this device, the C0 version of the interrupts is used for the MPU Subsystem, the C1 version is unused, and the C2 version is used for PRU-ICSS.
DMA Requests	None
Physical Address	L4 Fast slave port L3 Fast initiator port

15.2.2 Ethernet Switch Clock and Reset Management

The ethernet switch controller operates in its own clock domain and its initiator and target interfaces are connected to the L3/L4 through asynchronous bridges. The OCP interfaces are driven by the MAIN clock input. Additional reference clock inputs are provided for operating the various ethernet ports at different rates.

Table 15-3. Ethernet Switch Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
rft_clk Gigabit GMII Tx Reference clock	125 MHz	Tied low	not supported
main_clk Logic/Interface clock	125 MHz	CORE_CLKOUTM5 / 2	pd_per_cpsw_125mhz_gclk from PRCM
mhz250_clk Gigabit RGMII Reference clock	250 MHz	CORE_CLKOUTM5	pd_per_cpsw_250mhz_gclk from PRCM
mhz50_clk RMII and 100mbps RGMII Reference clock	50 MHz	CORE_CLKOUTM5 / 5	pd_per_cpsw_50mhz_gclk from PRCM
mhz5_clk 10 mbpsRGMII Reference clock	5 MHz	CORE_CLKOUTM5 / 50	pd_per_cpsw_5mhz_gclk from PRCM
cpts_rft_clk IEEE 1588v2 clock	250 MHz	CORE_CLKOUTM4 CORE_CLKOUTM5 Display PLL CLKOUT	pd_per_cpsw_cpts_rft_clk from PRCM
gmii1_mr_clk GMII Port 1 Receive clock	25 MHz	External Pin	gmii1_rxclk_in from GMII1_RCLK pad
gmii2_mr_clk GMII Port 2 Receive clock	25 MHz	External Pin	gmii2_rxclk_in from GMII2_RCLK pad
gmii1_mt_clk GMII Port 1 Transmit clock	25 MHz	External Pin	gmii1_txclk_in from GMII1_TCLK pad
gmii2_mt_clk GMII Port 2 Transmit clock	25 MHz	External Pin	gmii2_txclk_in from GMII2_TCLK pad
rgmii1_rxc_clk RGMII Port 1 Receive clock	250 MHz	External Pin	rgmii1_rclk_in from RGMII1_RCLK pad
rgmii2_rxc_clk RGMII Port 2 Receive clock	250 MHz	External Pin	rgmii2_rclk_in from RGMII2_RCLK pad
rmii1_mhz_50_clk RMII Port 1 Reference clock	50 MHz	External Pin	rmii1_refclk_in from RMII1_REFCLK pad
rmii2_mhz_50_clk RMII Port 2 Reference clock	50 MHz	External Pin	rmii2_refclk_in from RMII2_REFCLK pad

15.2.3 Ethernet Switch Pin List

The external signals for the Ethernet Switch module are shown in the following table.

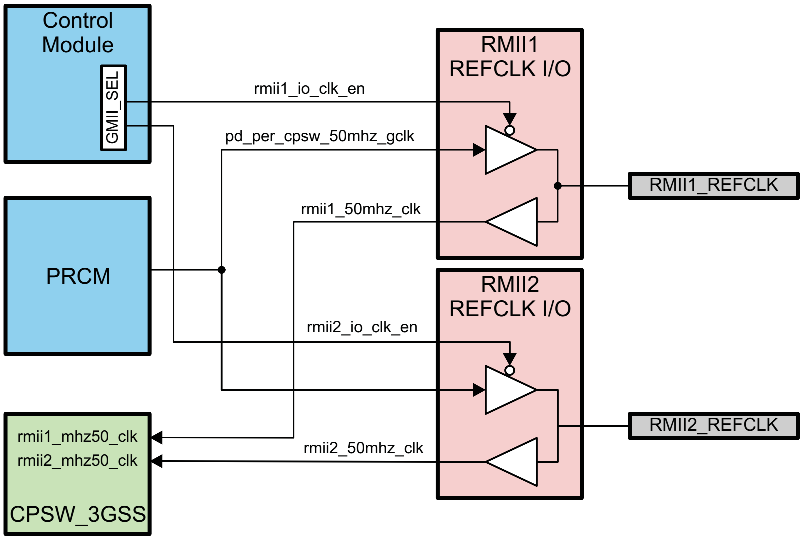
Table 15-4. Ethernet Switch Pin List

Pin	Type*	Description
GMIIx_RXCLK	I	GMII/MII Receive clock
GMIIx_RXD[3:0]	I	GMII/MII Receive data
GMIIx_RXDV	I	GMII/MII Receive data valid
GMIIx_RXER	I	GMII/MII Receive error
GMIIx_COL	I	GMII/MII Collision detect
GMIIx_CRS	I	GMII/MII Carrier sense
GMIIx_TXCLK	I	GMII/MII Transmit clock
GMII_TXD[3:0]	O	GMII/MII Transmit data
GMIIx_TXEN	O	GMII/MII Transmit enable
RGMIIx_RCLK	I	RGMII Receive clock
RGMIIx_RCTL	I	RGMII Receive control
RGMIIx_RD[3:0]	I	RGMII Receive data
RGMIIx_TCLK	O	RGMII Transmit clock
RGMIIx_TCTL	O	RGMII Transmit control
RGMIIx_TD[3:0]	O	RGMII Transmit data
RMIIx_RXD[1:0]	I	RMII Receiver data
RMIIx_RXER	I	RMII Receiver error
RMIIx_CRS_DV	I	RMII Carrier sense / Data valid
RMIIx_TXEN	O	RMII Transmit enable
RMIIx_REFCLK	I/O	RMII Reference clock
RMIIx_TXD[1:0]	O	RMII Transmit data
MDIO_CLK	O	MDIO Serial clock
MDIO_DATA	I/O	MDIO Serial data

15.2.4 Ethernet Switch RMII Clocking Details

The RMII interface reference clock pin operates as an input. When used as an input, the clock is driven by the I/O pad. The operation is controlled by the GMII_SEL[RMIIx_IO_CLK_EN] fields in the Control Module, as shown in Figure 15-2, and defaults to input mode.

Figure 15-2. Ethernet Switch RMII Clock Detail



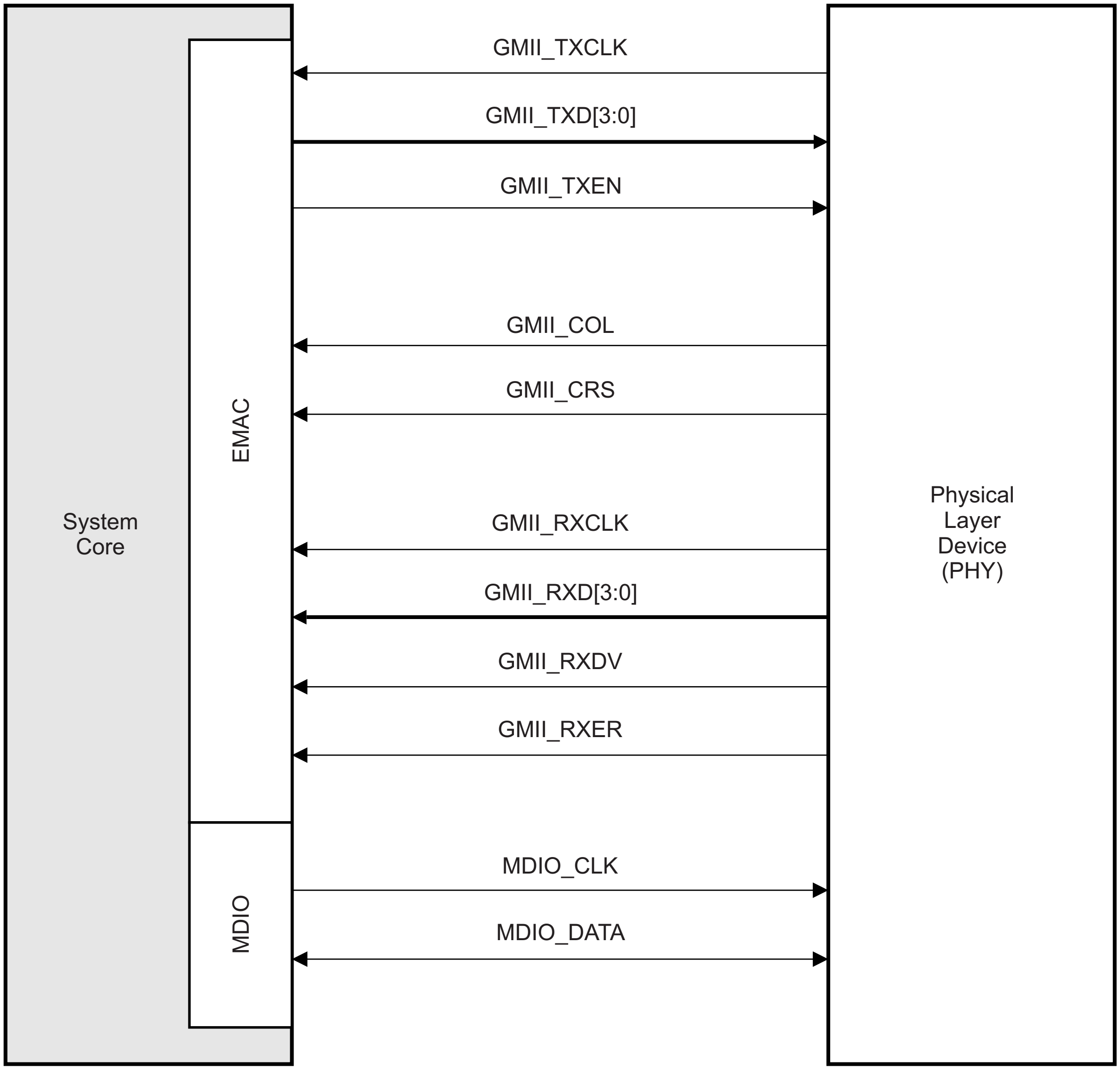
15.2.5 GMII Interface Signal Connections and Descriptions

GMII Interface can operate in MII Mode.

In MII Mode(100/10 Mbps) 3PSW operates in Full duplex and Half Duplex.

The pin connections of the GMII Interface are shown in Figure 15-3.

Figure 15-3. MII Interface Connections



The detailed description of the signals in MII Mode are explained in the following sections.

Table 15-5. GMII Interface Signal Descriptions in MII (100/10Mbps) Mode

Signal	Type	Description
GMII_TXCLK	I	The transmit clock is a continuous clock that provides the timing reference for transmit operations. The GMII_TXD and GMII_TXEN signals are tied to this clock. The clock is generated by the PHY and is 2.5 MHz at 10 Mbps operation and 25 MHz at 100 Mbps operation.
GMII_TXD[3:0]	O	The transmit data pins are a collection of 4 data signals GMII_TXD[3:0] comprising 4 bits of data. GMII_TXD[0] is the least-significant bit (LSB). The signals are synchronized by GMII_TXCLK and valid only when GMII_TXEN is asserted.
GMII_TXEN	O	The transmit enable signal indicates that the GMII_TXD[3:0] pins are generating 4-bit data for use by the PHY. It is driven synchronously by GMII_TXCLK
GMII_COL	I	In half-duplex operation, the GMII_COL pin is asserted by the PHY when it detects a collision on the network. It remains asserted while the collision condition persists. This signal is not necessarily synchronous to GMII_TXCLK nor GMII_RXCLK In full-duplex operation, the GMII_COL pin is used for hardware transmit flow control. Asserting the GMII_COL pin stops packet transmissions; packets transmitting when MCOL is asserted will complete transmission. The GMII_COL pin should be held low if hardware transmit flow control is not used
GMII_CRS	I	In half-duplex operation, the GMII_CRS pin is asserted by the PHY when the network is not idle in either transmit or receive. The pin is deasserted when both transmit and receive are idle. This signal is not necessarily synchronous to GMII_TXCLK nor GMII_RXCLK. In full-duplex operation, the GMII_CRS pin should be held low.
GMII_RXCLK	I	The receive clock is a continuous clock that provides the timing reference for receive operations. The GMII_RXD, GMII_RXDV, and MRXER signals are tied to this clock. The clock is generated by the PHY and is 2.5 MHz at 10 Mbps operation and 25 MHz at 100 Mbps operation.
GMII_RXD[3:0]	I	The receive data pins are a collection of 4 data signals comprising 4 bits of data. GMII_RXD[0] is the least-significant bit (LSB).The signals are synchronized by GMII_RXCLK and valid only when GMII_RXDV is asserted.
GMII_RXDV	I	The receive data valid signal indicates that the GMII_RXD pins are generating nibble data for use by the 3PSW. It is driven synchronously to GMII_RXCLK
MDIO_CLK	O	Management data clock (MDIO_CLK). The MDIO data clock is sourced by the MDIO module on the system. It is used to synchronize MDIO data access operations done on the MDIO_DATA pin.
MDIO_DATA	I/O	The MDIO_DATA pin drives PHY management data into and out of the PHY by way of an access frame consisting of start of frame, read/write indication, PHY address, register address, and data bit cycles. The MDIO_DATA pin acts as an output for all but the data bit cycles at which time it is an input for read operations.

15.2.6 RMII Signal Connections and Descriptions

Figure 15-4 shows a device with integrated 3PSW and MDIO interfaced via a RMII connection in a typical system.

The individual CPSW and MDIO signals for the RMII interface are summarized in Table 15-6.

For more information, see either the IEEE 802.3 standard or ISO/IEC 8802-3:2000(E).

Figure 15-4. RMII Interface Connections

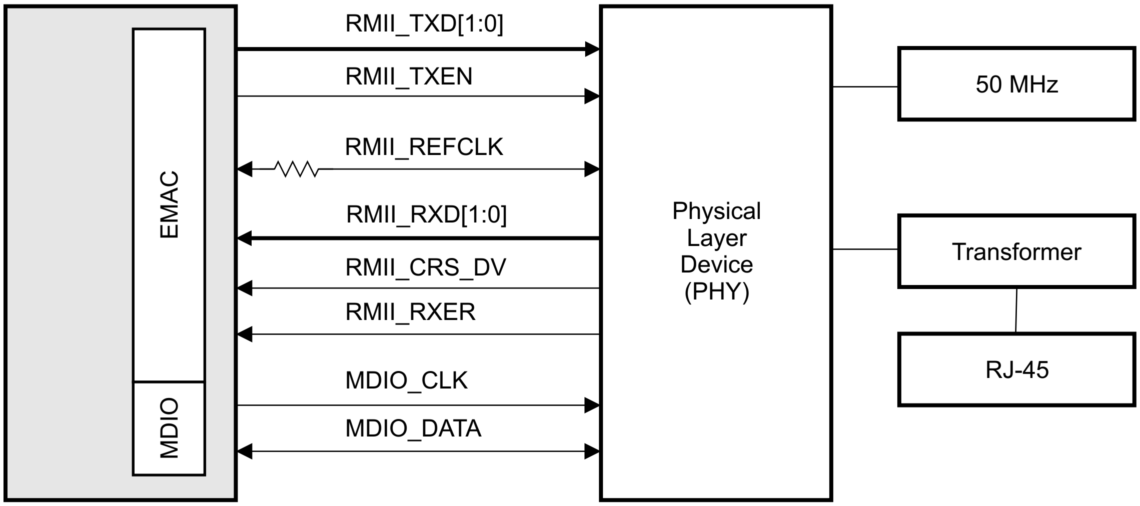


Table 15-6. RMII Interface Signal Descriptions

Signal	Type	Description
RMII_TXD[1-0]	O	Transmit data. The transmit data pins are a collection of 2 bits of data. RMII_TXD0 is the least-significant bit (LSB). The signals are synchronized by RMII_REFCLK and valid only when RMII_TXEN is asserted.
RMII_TXEN	O	Transmit enable. The transmit enable signal indicates that the RMII_TXD pins are generating data for use by the PHY. RMII_TXEN is synchronous to RMII_MHZ_50_CLK.
RMII_REFCLK	I	RMII reference clock. The reference clock is used to synchronize all RMII signals. RMII_REFCLK must be continuous and fixed at 50 MHz.
RMII_RXD[1-0]	I	Receive data. The receive data pins are a collection of 2 bits of data. RMII_RXD0 is the least-significant bit (LSB). The signals are synchronized by RMII_REFCLK and valid only when RM_CRS_DV is asserted and RMII_RXER is deasserted.
RMII_CRS_DV	I	Carrier sense/receive data valid. Multiplexed signal between carrier sense and receive data valid.
RMII_RXER	I	Receive error. The receive error signal is asserted to indicate that an error was detected in the received frame.
MDIO_CLK	O	Management data clock. The MDIO data clock is sourced by the MDIO module on the system. It is used to synchronize MDIO data access operations done on the MDIO pin.
MDIO_DATA	I/O	MDIO DATA. MDIO data pin drives PHY management data into and out of the PHY by way of an access frame consisting of start of frame, read/write indication,PHY address, register address, and data bit cycles. The MDIO_DATA signal acts as an output for all but the data bit cycles at which time it is an input for read operations.

15.2.7 RGMII Signal Connections and Descriptions

Figure 15-5 shows a device with integrated CPSW and MDIO interfaced via a RGMII connection in a typical system.

Figure 15-5. RGMII Interface Connections

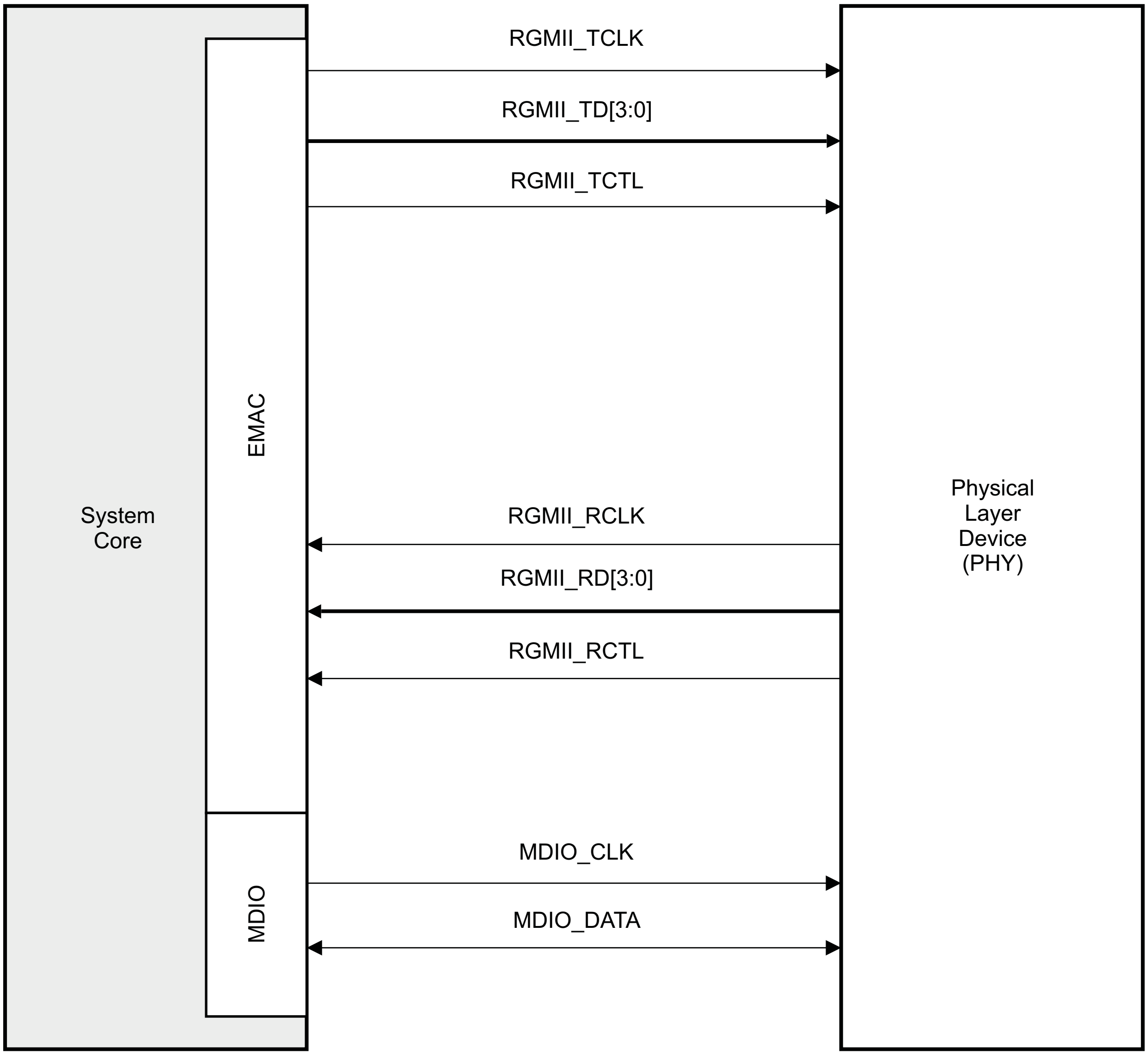


Table 15-7. RGMII Interface Signal Descriptions

Signal	Type	Description
RGMII_TD[3-0]	O	The transmit data pins are a collection of 4 bits of data. RGMII_RD0 is the least-significant bit (LSB). The signals are valid only when RGMII_TCTL is asserted.
RGMII_TCTL	O	Transmit Control/enable .The transmit enable signal indicates that the RGMII_TD pins are generating data for use by the PHY.
RGMII_TCLK	O	The transmit reference clock will be 125Mhz, 25Mhz, or 2.5Mhz depending on speed of operation.
RGMII_RD[3-0]	I	The receive data pins are a collection of 4 bits of data. RGMII_RD is the least-significant bit (LSB). The signals are valid only when RGMII_RCTL is asserted.

Table 15-7. RGMII Interface Signal Descriptions (continued)

Signal	Type	Description
RGMII_RCTL	I	The receive data valid/control signal indicates that the RGMII_RD pins are nibble data for use by the 3PSW.
RGMII_RCLK	I	The receive clock is a continuous clock that provides the timing reference for receive operations.The clock is generated by the PHY and is 2.5 MHz at 10 Mbps operation and 25 MHz at 100 Mbps operation,125 MHz at 1000Mbps of operation.
MDIO_CLK	O	Management data clock. The MDIO data clock is sourced by the MDIO module on the system. It is used to synchronize MDIO data access operations done on the MDIO pin.
MDIO_DATA	I/O	MDIO DATA. MDIO data pin drives PHY management data into and out of the PHY by way of an access frame consisting of start of frame, read/write indication,PHY address, register address, and data bit cycles. The MDIO_DATA pin acts as an output for all but the data bit cycles at which time it is an input for read operations.

15.3 Functional Description

The 3 port switch (3PSW) Ethernet Subsystem peripheral are compliant to the IEEE Std 802.3 Specification. The 3PSW Ethernet Subsystem contains two RGMII/RMII interfaces, one CPPI 3.0 interface, Interrupt Controller, MDIO and CPSW_3G which contains two GMII interfaces as shown in Figure 15-6.

The subsystem modules are discussed in detail in the following sections.

15.3.1 CPSW_3G Subsystem

15.3.1.1 Interrupt Pacing

The receive and transmit pulse interrupts can be paced. The receive threshold and miscellaneous interrupts are not paced. The Interrupt pacing feature limits the number of interrupts that occur during a given period of time. For heavily loaded systems in which interrupts can occur at a very high rate (e.g. 148,800 packets per second for Ethernet), the performance benefit is significant due to minimizing the overhead associated with servicing each interrupt. Interrupt pacing increases the CPU cache hit ratio by minimizing the number of times that large interrupt service routines are moved to and from the CPU instruction cache.

Each CPU receive and transmit pulse interrupt contains an interrupt pacing sub-block (six total). Each sub-block is disabled by default allowing the selected interrupt inputs to pass through unaffected. The interrupt pacing module counts the number of interrupts that occur over a 1ms interval of time. At the end of each 1ms interval, the current number of interrupts is compared with a target number of interrupts (specified by the associated maximum number of interrupts register).

Based on the results of the comparison, the length of time during which interrupts are blocked is dynamically adjusted. The 1ms interval is derived from a 4us pulse that is created from a prescale counter whose value is set in the int_prescale value in the Int_Ctrl register. The int_prescale value should be written with the number of VBUSP_CLK periods in 4us. The pacing timer determines the interval during which interrupts are blocked and decrements every 4us. It is reloaded each time a zero count is reached. The value loaded into the pacing timer is calculated by hardware every 1ms according to the following algorithm:

```
if (intr_count > 2*intr_max)
pace_timer = 255;
else if (intr_count > 1.5*intr_max)
pace_timer = last_pace_timer*2 + 1;
else if (intr_count > 1.0*intr_max)
pace_timer = last_pace_timer + 1;
else if (intr_count > 0.5*intr_max)
pace_timer = last_pace_timer - 1;
else if (intr_count != 0)
pace_timer = last_pace_timer/2;
else
pace_timer = 0;
```

If the rate of interrupt inputs is much less than the target interrupt rate specified in the associated maximum interrupts register, then the interrupt is not blocked. If the interrupt rate is greater than the target rate, the interrupt will be “paced” at the rate specified in the interrupt maximum register. The interrupt maximum register should be written with a value between 2 and 63 inclusive indicating the target number of interrupts per milli-second.

15.3.1.2 Reset Isolation

Reset isolation for the Ethernet switch on Device is that the switch function of the ethernet IP remains active even in case of all device resets except for POR pin reset and ICEPICK COLD reset. Packet traffic to/from the 3PSW host will be flushed/dropped, but the ethernet switch will remain operational for all traffic between external devices on the switch even though the device is under-going a device reset.Pin mux configuration for ethernet related IO and reference clocks needed by the Ethernet switch IP to be active is controlled by a protected control module bit. If reset isolation is enabled, then only a POR pin or ICEPICK COLD reset event should fully reset the Ethernet switch IP including the actual switch and also the reference clocks and pin mux control specifically associated with the Ethernet IP.

15.3.1.2.1 Modes of Operation

The device has two modes of operation concerning the reset of the 3PSW Ethernet switch.

The mode is controlled by the CPSW_ISO_CTRL bit in **RESET_ISO** register of the **device control module**.This bit should default to '0'. Writes to the CPSW_ISO_CTRL bit must be supervisor mode writes.

Mode 1: CPSW_ISO_CTRL=0 (reset isolation disabled)

- This mode is selected when CPSW_ISO_CTRL bit of control module is = 0. This should be the default state of the bit after control module reset.
- Upon any device level resets, the entire CPSW_3GSS_R IP, L3/L4, control module (including all pin mux control and the CPSW_ISO_CTRL bit) is immediately reset.

Mode 2: CPSW_ISO_CTRL=1 (reset isolation enabled)

- This mode is selected when CPSW_ISO_CTRL bit of control module is = 1.
- Upon any device reset source other than POR pin or ICEPICK cold (so this includes SW global cold, any watchdog reset, warm RESETn pin, ICEPICK warm, SW global warm), the following should be true:
  - The CPSW_3GSS_R is put into ‘isolate’ mode and non-switch related portions of the IP are reset.
  - The 50-MHz and 125-MHz reference clocks to the 3PSW Ethernet Subsystem remains active throughout the entire reset condition.
  - The control for pin multiplexing for all of the signals should maintain their current configuration throughout the entire reset condition.
  - The reset isolated logic inside 3PSW Ethernet Subsystem IP which maintains the switch functionality
  -
- Upon any cold reset sources, the entire 3PSW Ethernet Subsystem, control module (including all pin mux control and the CPSW_ISO_CTRL bit itself) is reset.

15.3.1.3 Interrupts

The 3 Port Switch Ethernet Subsystem generates four interrupt events.

15.3.1.3.1 Receive Packet Completion Pulse Interrupt (RX_PULSE)

The RX_PULSE interrupt is a paced pulse interrupt selected from the 3PSW RX_PEND [7:0] interrupts. The receive DMA controller has eight channels with each channel having a corresponding (RX_PEND[7:0]).

The following steps will enable the receive packet completion interrupt.

- Enable the required channel interrupts of the DMA engine by setting 1 to the appropriate bit in the RX_INTMASK_SET register.
- The receive completion interrupt(s) to be routed to RX_PULSE is selected by setting one or more bits in the receive interrupt enable register C*n*_RX_EN. The masked interrupt status can be read in the Receive Interrupt Masked Interrupt Status (Cn_RX_STAT) register.

When the 3PSW completes a packet reception, the subsystem issues an interrupt to the CPU by writing the packet's last buffer descriptor address to the appropriate channel queue's receive completion pointer located in the state RAM block. The interrupt is generated by the write when enabled by the interrupt mask, regardless of the value written.

Upon interrupt reception, the CPU processes one or more packets from the buffer chain and then

acknowledges one or more interrupt(s) by writing the address of the last buffer descriptor processed to the queue's associated receive completion pointer (RX*n*_CP) in the receive DMA state RAM.

Upon reception of an interrupt, software should perform the following:

- Read the RX_STAT register to determine which channel(s) caused the interrupt.
- Process received packets for the interrupting channel(s).
- Write the 3PSW completion pointer(s) (RX*n*_CP). The data written by the host (buffer descriptor address of the last processed buffer) is compared to the data in the register written by the subsystem (address of last buffer descriptor used by the subsystem). If the two values are not equal (which means that the 3PSW has received more packets than the CPU has processed), the receive packet completion interrupt signal remains asserted. If the two values are equal (which means that the host has processed all packets that the system has received), the pending interrupt is de-asserted. The value that the 3PSW is expecting is found by reading the receive channel*n* completion pointer register (RX*n*_CP).
- Write the value 1h to the CPDMA_EOI_VECTOR register.

To disable the interrupt:

- The eight channel interrupts may be individually disabled by writing to 1 the appropriate bit in the RX_INTMASK_CLR
- The receive completion pulse interrupt could be disabled by clearing to 0 all the bits of the RX_EN.

The software could still poll for the RX_INTSTAT_RAW and RX_INTSTAT_MASKED registers if the corresponding interrupts are enabled.

15.3.1.3.2 Transmit Packet Completion Pulse Interrupt (TX_PULSE)

The TX_PULSE interrupt is a paced pulse interrupt selected from the 3PSW TX_PEND [7:0] interrupts. The transmit DMA controller has eight channels with each channel having a corresponding (TX_PEND[7:0]).

To enable the transmit packet completion interrupt:

- Enable the required channel interrupts of the DMA engine by setting 1 to the appropriate bit in the TX_INTMASK_SET register.
- The transmit completion interrupt(s) to be routed to TX_PULSE is selected by setting one or more bits in the transmit interrupt enable register C*n*_TX_EN .The masked interrupt status can be read in the Transmit Interrupt Masked Interrupt Status (C*n*_TX_STAT) register.

When the 3PSW completes the transmission of a packet, the 3PSW subsystem issues an interrupt to the CPU by writing the packet's last buffer descriptor address to the appropriate channel queue's transmit completion pointer located in the state RAM block. The interrupt is generated by the write when enabled by the interrupt mask, regardless of the value written.

Upon receiving an interrupt, software should perform the following:

- Read the TX_STAT register to determine which channel(s) caused the interrupt
- Process received packets for the interrupting channel(s).
- Write the 3PSW completion pointer(s) (TX*n*_CP). The data written by the host (buffer descriptor address of the last processed buffer) is compared to the data in the register written by the 3PSW (address of last buffer descriptor used by the 3PSW). If the two values are not equal (which means that the 3PSW has transmitted more packets than the CPU has processed), the transmit packet completion interrupt signal remains asserted. If the two values are equal (which means that the host has processed all packets that the subsystem has transferred), the pending interrupt is cleared. The value that the 3PSW is expecting is found by reading the transmit channel*n* completion pointer register (TX*n*_CP).
- Write the 2h to the CPDMA_EOI_VECTOR register.

To disable the interrupt:

- The eight channel interrupts may be individually disabled by writing to 1 the appropriate bit in the TX_INTMASK_CLR.
- The receive completion pulse interrupt could be disabled by clearing to 0 all the bits of the TX_EN. The software could still poll for the TX_INTSTAT_RAW and TX_INTSTAT_MASKED registers if the corresponding interrupts are enabled.

15.3.1.3.3 Receive Threshold Pulse Interrupt (RX_THRESH_PULSE)

The RX_THRESH_PULSE interrupt is an immediate (non-paced) pulse interrupt selected from the CPSW_3G RX_THRESH_PEND[7:0] interrupts. The receive DMA controller has eight channels with each channel having a corresponding threshold pulse interrupt (RX_THRESH_PEND [7:0]).

To enable the receive threshold pulse Interrupt:

- Enable the required channel interrupts of the DMA engine by setting 1 to the appropriate bit in the RX_INTMASK_SET register.
- The receive threshold interrupt(s) to be routed to RX_THRESH_PULSE is selected by setting one or more bits in the receive threshold interrupt enable register RX_THRESH_EN. The masked interrupt status can be read in the Receive Threshold Masked Interrupt Status (C*n*_RX_THRESH_STAT) register.

The RX_THRESH_PULSE is asserted when enabled when the channel's associated free buffer count RX_FREEBUFFER*n* is less than or equal to the corresponding RX_PENDTHRESH*n* register.

Upon receiving an interrupt, software should perform the following:

- Read the C*n*_RX_THRESH_STAT bit address location to determine which channel(s) caused the interrupt.
- Process the received packets in order to add more buffers to any channel that is below the threshold value.
- Write the CPSW_3G completion pointer(s).
- Write the value 0h to the CPDMA_EOI_VECTOR register.

The threshold pulse interrupt is an immediate interrupt intended to indicate that software should immediately process packets to preclude an overrun condition from occurring for the particular channels.

To disable the interrupt:

- The eight channel receive threshold interrupts may be individually disabled by writing to 1 the appropriate bit in the RX_INTMASK_CLR register.
- The receive threshold pulse interrupt could be disabled by clearing to Zero the corresponding bits of the RX_THRESH_EN. The software could still poll for the RX_INTSTAT_RAW and INTSTAT_MASKED registers if the corresponding interrupts are enabled.

15.3.1.3.4 Miscellaneous Pulse Interrupt (MISC_PULSE)

The MISC_PULSE interrupt is an immediate, non-paced, pulse interrupt selected from the miscellaneous interrupts (EVNT_PEND, STAT_PEND, HOST_PEND, MDIO_LINKINT, MDIO_USERINT). The miscellaneous interrupt(s) is selected by setting one or more bits in the C*n*_MISC_EN[4:0] register. The masked interrupt status can be read in the C*n*_MISC_STAT[4:0] register. Upon reception of an interrupt, software should perform the following:

- Read the Misc_Stat[4:0] register to determine which caused the interrupt.
- Process the interrupt.
- Write the appropriate value (0x3) to the CPDMA_EOI_VECTOR register.
- Write a 1 to the appropriate bit in the MDIOLINKINTRAW register.

This device does not support multiple link interrupts. Only MDIO_LINKINT[0] and MDIO_USERINT[0] are used. MDIO_LINKINT[1] and MDIO_USERINT[1] are not used.

15.3.1.3.4.1 EVNT_PEND (CPTS_PEND) Interrupt

See Section 15.3.7, *Common Platform Time Sync (CPTS)*, for information on the time sync event interrupt.

15.3.1.3.4.2 Statistics Interrupt

The statistics level interrupt (STAT_PEND) will be asserted if enabled when any statistics value is greater than or equal to 0x80000000. The statistics interrupt is cleared by writing to decrement all statistics values greater than 0x80000000 (such that their new values are less than 0x80000000). The statistics interrupt is enabled by setting to one the appropriate bit in the INTMASK_SET register in the CPDMA submodule.

The statistics interrupt is disabled by writing one to the appropriate bit in the INTMASK_CLR register. The raw and masked statistics interrupt status may be read by reading the TX_IntStat_Raw and TX_IntStat_Masked registers, respectively

15.3.1.3.4.3 Host Error Interrupt

The host error interrupt (HOST_PEND) will be asserted if enabled when a host error is detected during transmit or receive CPDMA transactions. The host error interrupt is intended for software debug, and is cleared by a warm reset or a system reset. The raw and masked statistics interrupt status can be read by reading the TX_INTSTAT_RAW and TXINTSTAT_MASKED registers, respectively.

The following list shows the transmit host error conditions:

- SOP error
- OWNERSHIP bit not set in SOP buffer
- next buffer descriptor pointer without EOP set to 0
- buffer pointer set to 0
- buffer length set to 0
- packet length error

The receive host error conditions are show in the following list:

- Ownership bit not set in input buffer.
- Zero buffer pointer.
- Zero buffer Length on non-SOP descriptor.
- SOP buffer length not greater than offset.

The host error interrupt is disabled by clearing to 0 the appropriate bit in the CPDMA_INTMASK_CLR register.

15.3.1.3.4.4 MDIO Interrupts

MDIO_LINKINT is set if there is a change in the link state of the PHY corresponding to the address in the PHYADDRMON field of the MDIOUSERPHYSEL*n* register and the corresponding LINKINTENB bit is set.The MDIO_LINKINT event is also captured in the MDIOLINKINTMASKED register.When the GO bit in the MDIOUSERACCESS*n* registers transitions from 1 to 0, indicating the completion of a user access, and the corresponding USERINTMASKSET bit in the MDIOUSERINTMASKSET register is set, the MDIO_USERINT signal is asserted 1. The MDIO_USERINT event is also captured in the MDIOUSERINTMASKED register.

To enable the miscellaneous pulse interrupt:

The miscellaneous interrupt(s) is selected by setting one or more bits in the miscellaneous interrupt enable register (MISC_EN).

- The Statistics interrupt is enabled by setting to 1 the STAT_INT_MASK bit in the DMA_INTMASK_SET register.
- The HOST_PEND is enabled by setting to 1 the HOST_ERR_INT_MASK in the DMA_INTMASK_SET register.

Upon receiving of an interrupt, software should perform the following:

- Read the C*n*_MISC_STAT register to determine the source of the interrupt.
- Process the interrupt.
- Write the value 3h to the CPDMA_EOI_VECTOR register.

15.3.1.4 Embedded Memories

Memory Type Description	Number of Instances
Single port 2560 by 64 RAM	3 (Packet FIFOs)
Single port 64-word by 1152-bit RAM	1 (ALE)
Single port 2048-word by 32-bit RAM	1 (CPPI)

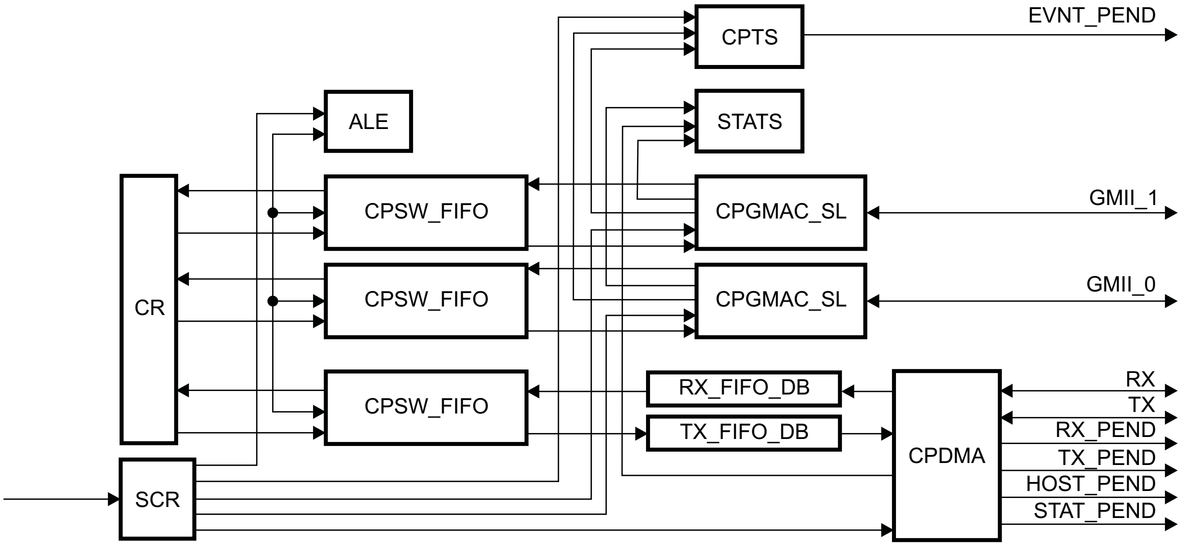
15.3.2 CPSW_3G

The CPSW_3G GMII interfaces are compliant to the IEEE Std 802.3 Specification.

The CPSW_3G contains two CPGMAC_SL interfaces (ports 1 and 2), one CPPI 3.0 interface Host Port (port 0), Common Platform Time Sync (CPTS), ALE Engine and CPDMA.

The top level block diagram of CPSW_3G is shown below:

Figure 15-6. CPSW_3G Block Diagram



15.3.2.1 Media Independent Interface (GMII)

The CPSW_3G contains two CPGMAC_SL submodules. Each CPGMAC_SL has a single GMII interface. The CPGMAC_SL submodules are ports 1 and 2. For more details, see Section 15.3.3, *Ethernet Mac Sliver (CPGMAC_SL)*.

15.3.2.2 IEEE 1588v2 Clock Synchronization Support

The CPSW_3G supports IEEE 1588v2 clock synchronization. Ethernet GMII Transmit (egress) and receive (ingress) time sync operation are also supported.

15.3.2.2.1 IEEE 1588v2 Receive Packet Operation

There are two CPSW_3G receive time sync interfaces for each ethernet port. The first is the TS_RX_MII interface and the second is the TS_RX_DEC interface. Both interfaces are generated in the switch and are input to the CPTS module. There are register bits in the CPSW_3G that control time sync operations in addition to the registers in the CPTS module. The pX_ts_rx_en bit in the switch Px_Ctrl register must be set for receive time sync operation to be enabled (TS_RX_MII).

The TS_RX_MII interface issues a record signal (pX_ts_rx_mii_rec) along with a handle (pX_ts_rx_mii_hndl[3:0]) to the CPTS controller for each packet that is received. The record signal is a single clock pulse indicating that a receive packet has been detected at the associated port MII interface.

The handle value is incremented with each packet and rolls over to zero after 15. There are 16 possible handle values so there can be a maximum of 16 packets “in flight” from the TS_RX_MII to the TS_RX_DEC block (through the CPGMAC_SL) at any given time. A handle value is reused (not incremented) for any received packet that is shorter than about 31 octets (including preamble). Handle reuse on short packets prevents any possible overrun condition if multiple fragments are consecutively received.

The TS_RX_MII logic is in the receive wireside clock domain. There is no decode logic in the TS_RX_MII to determine if the packet is a time sync event packet or not. Each received packet generates a record signal and new handle. The handle is sent to the CPTS controller with the record pulse and the handle is also sent to the TS_RX_DEC block along with the packet. The packet decode is performed in the TS_RX_DEC block. The decode function is separated from the record function because in some systems the incoming packet can be encrypted. The decode function would be after packet decryption in those systems.

The TS_RX_DEC function decodes each received packet and determines if the packet meets the time sync event packet criteria. If the packet is determined to be a time sync event packet, then the time sync event packet is signaled to the CPTS controller via the TS_RX_DEC interface (pX_ts_rx_dec_evnt, pX_ts_rx_dec_hndl[3:0], pX_ts_rx_dec_msg_type, pX_ts_rx_dec_seq_id). The event signal is a single clock pulse indicating that the packet matched the time sync event packet criteria and that the associated packet handle, message type, and sequence ID are valid. No indication is given for received packets that do not meet the time sync event criteria. The 16-bit sequence ID is found in the time sync event packet at the sequence ID offset into the PTP message header (pX_ts_seq_id_offset). A packet is determined to be a receive event packet under the following conditions:

15.3.2.2.1.1 Annex F

1. Receive time sync is enabled (pX_ts_rx_en is set in the switch Px_Ctrl register).
2. One of the following sequences is true:

   Where the first packet ltype matches:

   - ts_ltype1 and pX_ts_ltype1_en is set
   - ts_ltype2 and pX_ts_ltype2_en is set
   - vlan_ltype1 and pX_vlan_ltype1_en is set and the second packet ltype matches ts_ltype1 and pX_ts_ltype1_en is set
   - vlan_ltype1 and pX_vlan_ltype1_en is set and the second packet ltype matches ts_ltype2 and pX_ts_ltype2_en is set
   - vlan_ltype2 and pX_vlan_ltype2_en is set and the second packet ltype matches ts_ltype1 and pX_ts_ltype1_en is set
   - vlan_ltype2 and pX_vlan_ltype2_en is set and the second packet ltype matches ts_ltype2 and pX_ts_ltype2_en is set
   - vlan_ltype1 and pX_vlan_ltype1_en is set and the second packet ltype matches vlan_ltype2 and pX_vlan_ltype2_en is set and the third packet ltype matches ts_ltype1 and pX_ts_ltype1_en is set
   - vlan_ltype1 and pX_vlan_ltype1_en is set and the second packet ltype matches vlan_ltype2 and pX_vlan_ltype2_en is set and the third packet ltype matches ts_ltype2 and pX_ts_ltype2_en is set
3. The PTP message begins in the byte after the LTYPE.
4. The packet message type is enabled in the pX_ts_msg_type_en field in the Px_TS_SEQ_MTYPE register.
5. The packet was received without error (not long/short/mac_ctl/crc/code/align).
6. The ALE determined that the packet is to be sent only to the host (port 0).

15.3.2.2.1.2 Annex D

1. Receive time sync is enabled (pX_ts_rx_en is set in the switch Px_Ctrl register).
2. One of the following sequences is true:

   Where the first packet ltype matches:

   - 0x0800 and pX_ts_annex_d_en is set
   - vlan_ltype1 and pX_vlan_ltype1_en is set and the second packet ltype matches 0x0800 and pX_ts_annex_d_en is set
   - vlan_ltype2 and pX_vlan_ltype2_en is set and the second packet ltype matches 0x0800 and pX_ts_annex_d_en is set
   - vlan_ltype1 and pX_vlan_ltype1_en is set and the second packet ltype matches vlan_ltype2 and pX_vlan_ltype2_en is set and the third packet ltype matches 0x0800 and pX_ts_annex_d_en is set
3. Byte 14 (the byte after the LTYPE) contains 0x45 (IP_VERSION).

   Note: The byte numbering assumes that there are no vlans. The byte number is intended to show the relative order of the bytes.
4. Byte 22 contains 0x00 if the pX_ts_ttl_nonzero bit in the switch Px_Ctrl register is zero or byte 22 contains any value if pX_ts_ttl_nonzero is set. Byte 22 is the time to live field.

5. Byte 23 contains 0x11 (UDP Fixed).
6. Byte 30 contains decimal 224 (0xe0).
7. Byte 31 contains 0x00.
8. Byte 32 contains 0x01.
9. Byte 33 contains one of the following:
   - Decimal 129 and the pX_ts_129 bit in the switch Px_Ctrl register is set
   - Decimal 130 and the pX_ts_130 bit in the switch Px_Ctrl register is set
   - Decimal 131 and the pX_ts_131 bit in the switch Px_Ctrl register is set
   - Decimal 129 and the pX_ts_132 bit in the switch Px_Ctrl register is set
10. Bytes 36 and 37 contain one of the following:
    - Decimal 0x01 and 0x3f respectively and and the pX_ts_319 bit in the switch Px_Ctrl register is set.
    - Decimal 0x01 and 0x40 respectively and and the pX_ts_320 bit in the switch Px_Ctrl register is set.
11. The PTP message begins in byte 42.
12. The packet message type is enabled in the pX_ts_msg_type_en field in Px_Ctrl.
13. The packet was received without error (not long/short/mac_ctl/crc/code/align).
14. The ALE determined that the packet is to be sent only to the host (port 0).

15.3.2.2.2 IEEE 1588v2 Transmit Packet Operation

There are two CPSW_3G transmit time sync interfaces for each ethernet port. The first is the TS_TX_DEC interface and the second is the TS_TX_MII interface. Both interfaces are generated in the switch and are input to the CPTS module. The pX_ts_tx_en bit in the Px_Ctrl register must be set for transmit time sync operation to be enabled.

The TS_TX_DEC function decodes each packet to be transmitted and determines if the packet meets the time sync event packet criteria. If the packet is determined to be a time sync event packet, then the time sync event is signaled to the CPTS controller via the TS_TX_DEC interface (pX_ts_tx_dec_evnt, pX_ts_tx_dec_hndl[3:0], pX_ts_tx_dec_msg_type, pX_ts_tx_dec_seq_id). The event signal is a single clock pulse indicating that the packet matched the time sync event packet criteria and that the associated packet handle, message type, and sequence ID are valid.

The 16-bit sequence ID is found in the time sync event packet at the sequence ID offset into the message header (pX_ts_seq_id_offset). No indication is given for transmit packets that do not meet the time sync event criteria. The time sync event packet handle is also passed along with the packet to the TS_TX_MII with an indication that the packet is a time sync event packet. Unlike receive, only transmit event packets increment the handle value. The decode function is separated from the record function because some systems may encript the packet. The encription is after the decode function on transmit (egress). A packet is determined to be a transmit event packet if the following is met:

15.3.2.2.2.1 Annex F

1. Transmit time sync is enabled (pX_ts_tx_en is set in the switch Px_Ctrl register).
2. One of the following sequences is true:
   - The first packet ltype matches ts_ltype1 and pX_ts_ltype1_en is set
   - The first packet ltype matches ts_ltype2 and pX_ts_ltype2_en is set
   - The first packet ltype matches vlan_ltype1 and pX_vlan_ltype1_en is set and the second packet matches ts_ltype1 and pX_ts_ltype1_en is set
   - The first packet ltype matches vlan_ltype1 and pX_vlan_ltype1_en is set and the second packet ltype matches ts_ltype2 and pX_ts_ltype2_en is set
   - The first packet ltype matches vlan_ltype2 and pX_vlan_ltype2_en is set and the second packet ltype matches ts_ltype1 and pX_ts_ltype1_en is set
   - The first packet ltype matches vlan_ltype2 and pX_vlan_ltype2_en is set and the second packet ltype matches ts_ltype2 and pX_ts_ltype2_en is set

  - The first packet ltype matches vlan_ltype1 and pX_vlan_ltype1_en is set and the second packet ltype matches vlan_ltype2 and pX_vlan_ltype2_en is set and the third packet ltype matches ts_ltype1 and pX_ts_ltype1_en is set
  - The first packet ltype matches vlan_ltype1 and pX_vlan_ltype1_en is set and the second packet ltype matches vlan_ltype2 and pX_vlan_ltype2_en is set and the third packet ltype matches ts_ltype2 and pX_ts_ltype2_en is set

3. The packet message type is enabled in pX_ts_msg_type_en.
4. The packet was received by the host (port 0).

15.3.2.2.2.2 Annex D

1. Transmit time sync is enabled (pX_ts_tx_en is set in the switch Px_Ctrl register).
2. One of the following sequences is true:
   - The first packet ltype matches 0x0800 and pX_ts_annex_d_en is set
   - The first packet ltype matches vlan_ltype1 and pX_vlan_ltype1_en is set and the second packet ltype matches 0x0800 and pX_ts_annex_d_en is set
   - The first packet ltype matches vlan_ltype2 and pX_vlan_ltype2_en is set and the second packet ltype matches 0x0800 and pX_ts_annex_d_en is set
   - The first packet ltype matches vlan_ltype1 and pX_vlan_ltype1_en is set and the second packet ltype matches vlan_ltype2 and pX_vlan_ltype2_en is set and the third packet ltype matches 0x0800 and pX_ts_annex_d_en is set
3. Byte 14 (the byte after the LTYPE) contains 0x45 (IP_VERSION).

   Note: The byte numbering assumes that there are no vlans. The byte number is intended to show the relative order of the bytes. If VLAN(s) are present, then the byte numbers push down.
4. Byte 22 contains 0x00 if the pX_ts_ttl_nonzero bit in the switch Px_Ctrl register is zero or byte 22 contains any value if pX_ts_ttl_nonzero is set. Byte 22 is the time to live field.
5. Byte 23 contains 0x11 (UDP Fixed).
6. Byte 30 contains decimal 224 (0xe0)
7. Byte 31 contains 0x00
8. Byte 32 contains 0x01
9. Byte 33 contains one of the following:
   - Decimal 129 and the pX_ts_129 bit in the switch Px_Ctrl register is set
   - Decimal 130 and the pX_ts_130 bit in the switch Px_Ctrl register is set
   - Decimal 131 and the pX_ts_131 bit in the switch Px_Ctrl register is set
   - Decimal 132 and the pX_ts_132 bit in the switch Px_Ctrl register is set
10. Bytes 36 and 37 contain either of the following:
    - Decimal 1 (hex 0x01) and decimal 63 (hex 0x3f) respectively and and the pX_ts_319 bit in the switch Px_Ctrl register is set
    - Decimal 1 (hex 0x01) and decimal 64 (hex 0x40) respectively and and the pX_ts_320 bit in the switch Px_Ctrl register is set
11. The PTP message begins in byte 42 (this is offset 0).
12. The packet message type is enabled in pX_ts_msg_type_en.
13. The packet was received by the host (port 0).

The TS_TX_MII interface issues a single clock record signal (pX_ts_tx_mii_rec) at the beginning of each transmitted packet. If the packet is a time sync event packet then a single clock event signal (pX_ts_tx_mii_evnt) along with a handle (pX_ts_rx_mii_hndl[2:0]) will be issued before the next record signal for the next packet. The event signal will not be issued for packets that did not meet the time sync event criteria in the TS_TX_DEC function. If consecutive record indications occur without an interleaving event indication, then the packet associated with the first record was not a time sync event packet.

The record signal is a single clock pulse indicating that a transmit packet egress has been detected at the associated port MII interface. The handle value is incremented with each time sync event packet and rolls over to zero after 7. There are 8 possible handle values so there can be a maximum of eight time sync event packets “in flight” from the TS_TX_DEC to the TS_TX_MII block at any given time. The handle value increments only on time sync event packets. The TS_TX_MII logic is in the transmit wireside clock domain.

15.3.2.3 Device Level Ring (DLR) Support

Device Level Ring (DLR) support is enabled by setting the dlr_en bit in the CPSW_Ctrl register. When enabled, incoming received DLR packets are detected and sent to queue 3 (highest priority) of the egress port(s). If the host port is the egress port for a DLR packet then the packet is sent on the CPDMA Rx channel selected by the p0_dlr_cpdma_ch field in the P0_Ctrl register. The supervisor node MAC address feature is supported with the dlr_unicast bit in the unicast address table entry.

When set, the dlr_unicast bit causes a packet with the matching destination address to be flooded to the vlan_member_list minus the receive port and minus the host port (the port_number field in the unicast address table entry is a don’t care). Matching dlr_unicast packets are flooded regardless of whether the packet is a DLR packet or not. The en_p0_uni_flood bit in the ALE_Ctrl register has no effect on DLR unicast packets. Packets are determined to be DLR packets, as shown:

1. DLR is enabled (dlr_en is set in the switch CPSW_Ctrl register).
2. One of the following sequences is true:
   - The first packet ltype matches vlan_ltype1 and pX_vlan_ltype1_en is set and the second packet ltype matches dlr_ltype.
   - The first packet ltype matches vlan_ltype2 and pX_vlan_ltype2_en is set and the second packet ltype matches dlr_ltype.
   - The first packet ltype matches vlan_ltype1 and pX_vlan_ltype1_en is set and the second packet ltype matches vlan_ltype2 and pX_vlan_ltype2_en is set and the third packet ltype matches dlr_ltype.

15.3.2.4 CPDMA RX and TX Interfaces

The CPDMA submodule is a CPPI 3.0 compliant packet DMA transfer controller. The CPPI 3.0 interface is port 0.

After reset, initialization, and configuration the host may initiate transmit operations. Transmit operations are initiated by host writes to the appropriate transmit channel head descriptor pointer contained in the STATERAM block. The transmit DMA controller then fetches the first packet in the packet chain from memory in accordance with CPPI 3.0 protocol. The DMA controller writes the packet into the external transmit FIFO in 64-byte bursts (maximum).

Receive operations are initiated by host writes to the appropriate receive channel head descriptor pointer after host initialization and configuration. The receive DMA controller writes the receive packet data to external memory in accordance with CPPI 3.0 protocol. See the CPPI Buffer Descriptors section for detailed description of Buffer Descriptors

15.3.2.4.1 CPPI Buffer Descriptors

The buffer descriptor is a central part of the 3PSW Ethernet Subsytem and is how the application software describes Ethernet packets to be sent and empty buffers to be filled with incoming packet data.

Host Software sends and receives network frames via the CPPI 3.0 compliant host interface. The host interface includes module registers and host memory data structures. The host memory data structures are buffer descriptors and data buffers. Buffer descriptors are data structures that contain information about a single data buffer. Buffer descriptors may be linked together to describe frames or queues of frames for transmission of data and free buffer queues available for received data.

Note: The 8k bytes of Ethernet Subsystem CPPI RAM begin at address 0x4a102000 and end at 0x4a103FFF from the 3PSW perspective. The buffer descriptors programmed to access the CPPI RAM memory should use the address range from 0x4a102000.

15.3.2.4.1.1 TX Buffer Descriptors

A TX buffer descriptor is a contiguous block of four 32-bit data words aligned on a 32-bit word boundary.

Figure 15-7. Tx Buffer Descriptor Format

Word 0

31 – 0
Next Descriptor Pointer

Word 1

31 – 0
Buffer Pointer

Word 2

31 – 16	15 – 0
Buffer Offset	Buffer Length

Word 3

31	30	29	28	27	26	25 – 21	20	19 – 18	17 – 16
SOP	EOP	Ownership	EOQ	Teardown_Complete	Pass CRC	Reserved	To_Port_En	Reserved	To_Port

15 – 11	10 – 0
Reserved	packet_length

15.3.2.4.1.1.1 CPPI Tx Data Word – 0

Next Descriptor Pointer

The next descriptor pointer points to the 32-bit word aligned memory address of the next buffer descriptor in the transmit queue. This pointer is used to create a linked list of buffer descriptors. If the value of this pointer is zero, then the current buffer is the last buffer in the queue. The software application must set this value prior to adding the descriptor to the active transmit list. This pointer is not altered by the EMAC.The value of pNext should never be altered once the descriptor is in an active transmit queue, unless its current value is NULL.

If the pNext pointer is initially NULL, and more packets need to be queued for transmit, the software application may alter this pointer to point to a newly appended descriptor. The EMAC will use the new pointer value and proceed to the next descriptor unless the pNext value has already been read. In this latter case, the transmitter will halt on the transmit channel in question, and the software application may restart it at that time. The software can detect this case by checking for an end of queue (EOQ) condition flag on the updated packet descriptor when it is returned by the EMAC.

15.3.2.4.1.1.2 CPPI Tx Data Word – 1

Buffer Pointer

The byte aligned memory address of the buffer associated with the buffer descriptor. The host sets the **buffer_pointer**. The software application must set this value prior to adding the descriptor to the active transmit list. This pointer is not altered by the EMAC.

15.3.2.4.1.1.3 CPPI Tx Data Word – 2

Buffer _Offset

Buffer Offset – Indicates how many unused bytes are at the start of the buffer. A value of 0x0000 indicates that no unused bytes are at the start of the buffer and that valid data begins on the first byte of the buffer. A value of 0x000F (decimal 15) indicates that the first 15 bytes of the buffer are to be ignored by the port and that valid buffer data starts on byte 16 of the buffer. The host sets the buffer_offset value (which may be zero to the buffer length minus 1). Valid only on sop.

Buffer _Length

Buffer Length – Indicates how many valid data bytes are in the buffer. Unused or protocol specific bytes at the beginning of the buffer are not counted in the Buffer Length field. The host sets the buffer_length. The buffer_length must be greater than zero.

15.3.2.4.1.1.4 CPPI Tx Data Word – 3

Start of Packet (SOP) Flag

When set, this flag indicates that the descriptor points to a packet buffer that is the start of a new packet.In the case of a single fragment packet, both the SOP and end of packet (EOP) flags are set. Otherwise,the descriptor pointing to the last packet buffer for the packet sets the EOP flag. This bit is set by the software application and is not altered by the EMAC.

0 - Not start of packet buffer

1 - Start of packet buffer

End of Packet (EOP) Flag

When set, this flag indicates that the descriptor points to a packet buffer that is last for a given packet. In the case of a single fragment packet, both the start of packet (SOP) and EOP flags are set. Otherwise, the descriptor pointing to the last packet buffer for the packet sets the EOP flag. This bit is set by the software application and is not altered by the EMAC.

0 - Not end of packet buffer.

1 - End of packet buffer.

Ownership

When set this flag indicates that all the descriptors for the given packet (from SOP to EOP) are currently owned by the EMAC. This flag is set by the software application on the SOP packet descriptor before adding the descriptor to the transmit descriptor queue. For a single fragment packet, the SOP, EOP, and OWNER flags are all set. The OWNER flag is cleared by the EMAC once it is finished with all the descriptors for the given packet. Note that this flag is valid on SOP descriptors only.

0 - The packet is owned by the host

1 - The packet is owned by the port

EOQ

When set, this flag indicates that the descriptor in question was the last descriptor in the transmit queue for a given transmit channel, and that the transmitter has halted. This flag is initially cleared by the software application prior to adding the descriptor to the transmit queue. This bit is set by the EMAC when the EMAC identifies that a descriptor is the last for a given packet (the EOP flag is set), and there are no more descriptors in the transmit list (next descriptor pointer is NULL).

The software application can use this bit to detect when the EMAC transmitter for the corresponding channel has halted. This is useful when the application appends additional packet descriptors to a transmit queue list that is already owned by the EMAC. Note that this flag is valid on EOP descriptors only.

0 - The Tx queue has more packets to transfer.

1 - The Descriptor buffer is the last buffer in the last packet in the queue.

teardown_Complete

This flag is used when a transmit queue is being torn down, or aborted, instead of allowing it to be

transmitted. This would happen under device driver reset or shutdown conditions. The EMAC sets this bit in the SOP descriptor of each packet as it is aborted from transmission. Note that this flag is valid on SOP descriptors only. Also note that only the first packet in an unsent list has the TDOWNCMPLT flag set. Subsequent descriptors are not processed by the EMAC.

0 - The port has not completed the teardown process.

1 - The port has completed the commanded teardown process.

pass_crc

This flag is set by the software application in the SOP packet descriptor before it adds the descriptor to the transmit queue. Setting this bit indicates to the EMAC that the 4 byte Ethernet CRC is already present in the packet data, and that the EMAC should not generate its own version of the CRC.When the CRC flag is cleared, the EMAC generates and appends the 4-byte CRC. The buffer length and packet length fields do not include the CRC bytes. When the CRC flag is set, the 4-byte CRC is supplied by the software application and is already appended to the end of the packet data. The buffer length and packet length fields include the CRC bytes, as they are part of the valid packet data. Note that this flag is valid on SOP descriptors only.

0 – The CRC is not included with the packet data and packet length.

1 – The CRC is included with the packet data and packet length.

to_port

To Port – Port number to send the directed packet to. This field is set by the host. This field is valid on SOP. Directed packets go to the directed port, but an ALE lookup is performed to determine untagged egress in VLAN_AWARE mode.

1 – Send the packet to port 1 if to_port_en is asserted.

2 – Send the packet to port 2 if to_port_en is asserted.

To_port_enable

To Port Enable – Indicates when set that the packet is a directed packet to be sent to the to_port field port number. This field is set by the host. The packet is sent to one port only (index not mask). This bit is valid on SOP.

0 – not a directed packet

1 – directed packet

Packet Length

Specifies the number of bytes in the entire packet. Offset bytes are not included. The sum of the buffer_length fields should equal the packet_length. Valid only on SOP. The packet length must be greater than zero. The packet data will be truncated to the packet length if the packet length is shorter than the sum of the packet buffer descriptor buffer lengths. A host error occurs if the packet length is greater than the sum of the packet buffer descriptor buffer lengths

15.3.2.4.1.2 RX Buffer Descriptors

An RX buffer descriptor is a contiguous block of four 32-bit data words aligned on a 32-bit word boundary.

Figure 15-8. Rx Buffer Descriptor Format

Word 0

31 – 0
Next Descriptor Pointer

Word 1

31 – 0
Buffer Pointer

Word 2

31 – 27	26 – 16	15 – 11	10 – 0
Reserved	Buffer Offset	Reserved	Buffer Length

Word 3

31	30	29	28	27	26	25	24	23	22	21 – 20	19	18 – 16
SOP	EOP	Ownership	EOQ	Teardown_Complete	Passed_CRC	Long	Short	MAC_Ctl	Overrun	PKT_Err	Rx_Vlan_Encap	From_Port

15 – 11	10 – 0
Reserved	packet_length

15.3.2.4.1.2.1 CPPI Rx Data Word – 0

next_descriptor_pointer

The 32-bit word aligned memory address of the next buffer descriptor in the RX queue. This is the mechanism used to reference the next buffer descriptor from the current buffer descriptor. If the value of this pointer is zero then the current buffer is the last buffer in the queue. The host sets the **next_descriptor_pointer**.

15.3.2.4.1.2.2 CPPI Rx Data Word – 1

buffer_pointer

The byte aligned memory address of the buffer associated with the buffer descriptor. The host sets the **buffer_pointer**.

15.3.2.4.1.2.3 CPPI Rx Data Word – 2

Buffer _Offset

Buffer Offset – Indicates how many unused bytes are at the start of the buffer. A value of 0x0000 indicates that there are no unused bytes at the start of the buffer and that valid data begins on the first byte of the buffer. A value of 0x000F (decimal 15) indicates that the first 15 bytes of the buffer are to be ignored by the port and that valid buffer data starts on byte 16 of the buffer. The port writes the **buffer_offset** with the value from the **rx_buffer_offset** register value. The host initializes the **buffer_offset** to zero for free buffers. The **buffer_length** must be greater than the **rx_buffer_offset** value. The buffer offset is valid only on **sop**.

Buffer _Length

Buffer Length – Indicates how many valid data bytes are in the buffer. Unused or protocol specific bytes at the beginning of the buffer are not counted in the Buffer Length field. The host initializes the **buffer_length**, but the port may overwrite the host initiated value with the actual buffer length value on SOP and/or EOP buffer descriptors. SOP buffer length values will be overwritten if the packet size is less than the size of the buffer or if the offset is nonzero. EOP buffer length values will be overwritten if the entire buffer is not filled up with data. The **buffer_length** must be greater than zero.

15.3.2.4.1.2.4 CPPI Rx Data Word – 3

Start of Packet (SOP) Flag

When set, this flag indicates that the descriptor points to a packet buffer that is the start of a new packet.In the case of a single fragment packet, both the SOP and end of packet (EOP) flags are set. Otherwise, the descriptor pointing to the last packet buffer for the packet has the EOP flag set. This flag is initially cleared by the software application before adding the descriptor to the receive queue. This bit is set by the EMAC on SOP descriptors.

End of Packet (EOP) Flag

When set, this flag indicates that the descriptor points to a packet buffer that is last for a given packet. In the case of a single fragment packet, both the start of packet (SOP) and EOP flags are set. Otherwise, the descriptor pointing to the last packet buffer for the packet has the EOP flag set. This flag is initially cleared by the software application before adding the descriptor to the receive queue. This bit is set by the EMAC on EOP descriptors.

Ownership (OWNER) Flag

When set, this flag indicates that the descriptor is currently owned by the EMAC. This flag is set by the software application before adding the descriptor to the receive descriptor queue. This flag is cleared by the EMAC once it is finished with a given set of descriptors, associated with a received packet. The flag is updated by the EMAC on SOP descriptor only. So when the application identifies that the OWNER flag is cleared on an SOP descriptor, it may assume that all descriptors up to and including the first with the EOP flag set have been released by the EMAC. (Note that in the case of single buffer packets, the same descriptor will have both the SOP and EOP flags set.)

End of Queue (EOQ) Flag

When set, this flag indicates that the descriptor in question was the last descriptor in the receive queue for a given receive channel, and that the corresponding receiver channel has halted. This flag is initially cleared by the software application prior to adding the descriptor to the receive queue. This bit is set by the EMAC when the EMAC identifies that a descriptor is the last for a given packet received (also sets the EOP flag), and there are no more descriptors in the receive list (next descriptor pointer is NULL).The software application can use this bit to detect when the EMAC receiver for the corresponding channel has halted. This is useful when the application appends additional free buffer descriptors to an active receive queue. Note that this flag is valid on EOP descriptors only.

Teardown Complete (TDOWNCMPLT) Flag

This flag is used when a receive queue is being torn down, or aborted, instead of being filled with received data. This would happen under device driver reset or shutdown conditions. The EMAC sets this bit in the descriptor of the first free buffer when the tear down occurs. No additional queue processing is performed.

Pass CRC (PASSCRC) Flag

This flag is set by the EMAC in the SOP buffer descriptor if the received packet includes the 4-byte CRC.This flag should be cleared by the software application before submitting the descriptor to the receive queue.

Long (Jabber) Flag

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet is a jabber frame and was not discarded because the RX_CEF_EN bit was set in the MACCTRL. Jabber frames are frames that exceed the RXMAXLEN in length, and have CRC, code, or alignment errors.

Short (Fragment) Flag

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet is only a packet fragment and was not discarded because the RX_CSF_EN bit was set in the MACCTRL.

Control Flag

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet is an EMAC control frame and was not discarded because the RX_CMF_EN bit was set in the MACCTRL.

Overrun Flag

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet was aborted due to a receive overrun.

Pkt_error Flag

Packet Contained Error on Ingress –

00 – no error

01 – CRC error on ingress

10 – Code error on ingress

11 – Align error on ingress

rx_vlan_encap

VLAN Encapsulated Packet – Indicates when set that the packet data contains a 32-bit VLAN header word that is included in the packet byte count. This field is set by the port to be the value of the CPSW control register rx_vlan_encap bit

from_port

From Port – Indicates the port number that the packet was received on (ingress to the switch).

Packet Length

Specifies the number of bytes in the entire packet. The packet length is reduced to 12-bits. Offset bytes are not included. The sum of the buffer_length fields should equal the packet_length. Valid only on SOP.

15.3.2.4.2 Receive DMA Interface

The Receive DMA is an eight channel CPPI 3.0 compliant interface. Each channel has a single queue for frame reception.

15.3.2.4.2.1 Receive DMA Host Configuration

To configure the Rx DMA for operation the host must perform the following:

- Initialize the receive addresses.
- Initialize the Rx_HDP Registers to zero.
- Enable the desired receive interrupts in the IntMask register.
- Write the rx_buffer_offset register value.
- Setup the receive channel(s) buffer descriptors in host memory as required by CPPI 3.0.
- Enable the RX DMA controller by setting the rx_en bit in the Rx_Ctrl register.

15.3.2.4.2.2 Receive Channel Teardown

The host commands a receive channel teardown by writing the channel number to the Rx_Teardown register. When a teardown command is issued to an enabled receive channel the following will occur:

- Any current frame in reception will complete normally.
- The teardown_complete bit will be set in the next buffer descriptor in the chain

if there is one.

- The channel head descriptor pointer will be cleared to zero
- A receive interrupt for the channel will be issued to the host.
- The host should acknowledge a teardown interrupt with a 0xfffffffc acknowledge value.

Channel teardown may be commanded on any channel at any time. The host is informed of the teardown completion by the set teardown complete buffer descriptor bit. The port does not clear any channel enables due to a teardown command. A teardown command to an inactive channel issues an interrupt that software should acknowledge with a 0xfffffffc acknowledge value (note that there is no buffer descriptor in this case). Software may read the interrupt acknowledge location to determine if the interrupt was due to a commanded teardown. The read value will be 0xfffffffc if the interrupt was due to a teardown command.

15.3.2.4.3 Transmit DMA Interface

The Transmit DMA is an eight channel CPPI 3.0 compliant interface. Priority between the eight queues may be either fixed or round robin as selected by tx_ptype in the DMA_Ctrl register. If the priority type is fixed, then channel 7 has the highest priority and channel 0 has the lowest priority. Round robin priority proceeds from channel 0 to channel 7. Packet Data transfers occur on the TX_VBUSP interface in 64-byte maximum burst transfers

15.3.2.4.3.1 Transmit DMA Host Configuration

To configure the TX DMA for operation the host must do the following:

- Initialize the Tx_HDP registers to a zero value.
- Enable the desired transmit interrupts in the IntMask register.
- Setup the transmit channel(s) buffer descriptors in host memory as defined in CPPI 3.0.
- Configure and enable the transmit operation as desired in the Tx_Ctrl register.
- Write the appropriate Tx_HDP registers with the appropriate values to start transmit operations.

15.3.2.4.3.2 Transmit Channel Teardown

The host commands a transmit channel teardown by writing the channel number to the Tx_Teardown register. When a teardown command is issued to an enabled transmit channel the following will occur:

- Any frame currently in transmission will complete normally
- The teardown complete bit will be set in the next sop buffer descriptor (if there is one).
- The channel head descriptor pointer will be set to zero.
- An interrupt will be issued to inform the host of the channel teardown.
- The host should acknowledge a teardown interrupt with a 0xfffffffc acknowledge value

Channel teardown may be commanded on any channel at any time. The host is informed of the teardown completion by the set teardown complete buffer descriptor bit. The port does not clear any channel enables due to a teardown command. A teardown command to an inactive channel issues an interrupt that software should acknowledge with a 0xfffffffc acknowledge value (note that there is no buffer descriptor in this case). Software may read the interrupt acknowledge location to determine if the interrupt was due to a commanded teardown. The read value will be 0xfffffffc if the interrupt was due to a teardown command.

15.3.2.4.4 Transmit Rate Limiting

Transmit operations can be configured to rate limit the transmit data for each transmit priority. Rate limiting is enabled for a channel when the tx_rlim[7:0] bit associated with that channel is set in the DMA_Ctrl register. Rate limited channels must be the highest priority channels. For example, if two rate limited channels are required then tx_rlim[7:0] should be set to 11000000 with the msb corresponding to channel 7.

When any channels are configured to be rate-limiting, the priority type must be fixed for transmit. Round-robin priority type is not allowed when rate-limiting. Each of the eight transmit priorities has an associated register to control the rate at which the priority is allowed to send data (Tx_Pri(7..0)_Rate) when the channel is rate-limiting. Each priority has a send count (pri(7..0)_send_cnt[13:0]) and an idle count (pri(7..0)_idle_cnt[13:0]). The transfer rate includes the inter-packet gap (12 bytes) and the preamble (8 bytes). The rate in Mbits/second that each priority is allowed to send is controlled by the below equation.

Priority Transfer rate in Mbit/s = ((priN_idle_cnt/(priN_idle_cnt + priN_send_cnt)) * frequency * 32

Where the frequency is the CLK frequency.

15.3.2.4.5 Command IDLE

The cmd_idle bit in the DMA_Ctrl register allows CPDMA operation to be suspended. When the idle state is commanded, the CPDMA will stop processing receive and transmit frames at the next frame boundary. Any frame currently in reception or transmission will be completed normally without suspension. For transmission, any complete or partial frame in the tx cell fifo will be transmitted. For receive, frames that are detected by the CPDMA after the suspend state is entered are ignored. No statistics will be kept for ignored frames. Commanded idle is similar in operation to emulation control and clock stop.

15.3.2.5 VLAN Aware Mode

The CPSW_3G is in VLAN aware mode when the CPSW Control register vlan_aware bit is set. In VLAN aware mode ports 0 receive packets (out of the CPSW_3G) may or may not be VLAN encapsulated depending on the CPSW Control register rx_vlan_encap bit. The header packet VLAN is generated as described in Section 15.3.3, *Ethernet Mac Sliver (CPGMAC_SL)*. Port 0 receive packet data is never modified. VLAN is not removed regardless of the force untagged egress bit for Port 0. VLAN encapsulated receive packets have a 32-bit VLAN header encapsulation word added to the packet data.VLAN encapsulated packets are specified by a set rx_vlan_encap bit in the packet buffer descriptor.

Port 0 transmit packets are never VLAN encapsulated (encapsulation is not allowed).

In VLAN aware mode, transmitted packet data is changed depending on the packet type (pkt_type), packet priority (pkt_pri), and VLAN information as shown in the below tables:

Figure 15-9. VLAN Header Encapsulation Word

31-29	28	27-16
HDR_PKT_Priority	HDR_PKT_CFI	HDR_PKT_Vid

15-10	9-8	7-0
Reserved	PKT_Type	Reserved

Table 15-8. VLAN Header Encapsulation Word Field Descriptions

Field	Description
HDR_PKT_Priority	Header Packet VLAN priority (Highest priority: 7)
HDR_PKT_CFI	Header Packet VLAN CFI bit.
HDR_PKT_Vid	Header Packet VLAN ID
PKT_Type	Packet Type. Indicates whether the packet is VLAN-tagged, priority-tagged, or non-tagged. 00: VLAN-tagged packet 01: Reserved 10: Priority-tagged packet 11: Non-tagged packet

15.3.2.6 VLAN Unaware Mode

The CPSW_3G is in VLAN unaware mode when the CPSW Control register vlan_aware bit is cleared. Port 0 receive packets (out of the CPSW_3G) may or may not be VLAN encapsulated depending on the CPSW Control register rx_vlan_encap bit. Port 0 transmit packets are never VLAN encapsulated.

15.3.2.7 Address Lookup Engine (ALE)

The address lookup engine (ALE) processes all received packets to determine which port(s) if any that the packet should the forwarded to. The ALE uses the incoming packet received port number, destination address, source address, length/type, and VLAN information to determine how the packet should be forwarded. The ALE outputs the port mask to the switch fabric that indicates the port(s) the packet should be forwarded to. The ALE is enabled when the ale_enable bit in the ALE_Ctrl register is set. All packets are dropped when the ale_enable bit is cleared to zero.

In normal operation, the CPGMAC_SL modules are configured to issue an abort, instead of an end of packet, at the end of a packet that contains an error (runt, frag, oversize, jabber, crc, alignment, code etc.) or at the end of a mac control packet. However, when the CPGMAC_SL configuration bit(s) cef, csf, or cmf are set, error frames, short frames or mac control frames have a normal end of packet instead of an abort at the end of the packet. When the ALE receives a packet that contains errors (due to a set header error bit), or a mac control frame and does not receive an abort, the packet will be forwarded only to the host port (port 0). No ALE learning occurs on packets with errors or mac control frames. Learning is based on source address and lookup is based on destination address.

The ALE may be configured to operate in bypass mode by setting the ale_bypass bit in the ALE_Ctrl register. When in bypass mode, all CPGMAC_SL received packets are forwarded only to the host port (port 0). Packets from the two ports can be on separate Rx DMA channels by configuring the CPDMA_Rx_Ch_Map register. In bypass mode, the ALE processes host port transmit packets the same as in normal mode. In general, packets would be directed by the host in bypass mode.

The ALE may be configured to operate in OUI deny mode by setting the enable_oui_deny bit in the ALE_Ctrl register. When in OUI deny mode, a packet with a non-matching OUI source address will be dropped unless the destination address matches a multicast table entry with the super bit set. Broadcast packets will be dropped unless the broadcast address is entered into the table with the super bit set. Unicast packets will be dropped unless the unicast address is in the table with block and secure both set (supervisory unicast packet).

Multicast supervisory packets are designated by the super bit in the table entry. Unicast supervisory packets are indicated when block and secure are both set. Supervisory packets are not dropped due to rate limiting, OUI, or VLAN processing.

15.3.2.7.1 Address Table Entry

The ALE table contains 1024 entries. Each table entry represents a free entry, an address, a VLAN, an address/VLAN pair, or an OUI address. Software should ensure that there are not double address entries in the table. The double entry used would be indeterminate. Reserved table bits must be written with zeroes.

Source Address learning occurs for packets with a unicast, multicast or broadcast destination address and a unicast or multicast (including broadcast) source address. Multicast source addresses have the group bit (bit 40) cleared before ALE processing begins, changing the multicast source address to a unicast source address. A multicast address of all ones is the broadcast address which may be added to the table. A learned unicast source address is added to the table with the following control bits:

Table 15-9. Learned Address Control Bits

unicast_type	11
Block	0
Secure	0

If a received packet has a source address that is equal to the destination address then the following occurs:

- The address is learned if the address is not found in the table.
- The address is updated if the address is found.
- The packet is dropped.

15.3.2.7.1.1 Free Table Entry

Table 15-10. Free (Unused) Address Table Entry Bit Values

71:62	61:60	59:0
Reserved	Entry Type (00)	Reserved

15.3.2.7.1.2 Multicast Address Table Entry

Table 15-11. Multicast Address Table Entry Bit Values

71:70	68:66	65	64	63:62	61:60	59:48	47:0
Reserved	Port Mask	Super	Reserved	Mcast Fwd State	Entry Type (01)	Reserved	Multicast Address

Table Entry Type

00: Free Entry

01: Address Entry : unicast or multicast determined by dest **address bit 40 .**

10: VLAN entry

11: VLAN Address Entry : unicast or multicast determined by **address bit 40.**

Supervisory Packet (SUPER)

When set, this field indicates that the packet with a matching multicast destination address is a supervisory packet.

0: Non-supervisory packet

1: Supervisory packet

Port Mask(2:0) (PORT_MASK)

This three bit field is the port bit mask that is returned with a found multicast destination address. There may be multiple bits set indicating that the multicast packet may be forwarded to multiple ports (but not the receiving port).

Multicast Forward State (MCAST_FWD_STATE)

Multicast Forward State – Indicates the port state(s) required for the received port on a destination address lookup in order for the multicast packet to be forwarded to the transmit port(s). A transmit port must be in the Forwarding state in order to forward the packet. If the transmit port_mask has multiple set bits then each forward decision is independent of the other transmit port(s) forward decision.

00: Forwarding

01: Blocking/Forwarding/Learning

10: Forwarding/Learning

11: Forwarding

The forward state test returns a true value if both the Rx and Tx ports are in the required state.

Table Entry Type (ENTRY_TYPE)

Address entry type. Unicast or multicast determined by address bit 40.

01: Address entry. Unicast or multicast determined by address bit 40.

Packet Address (MULTICAST_ADDRESS)

This is the 48-bit packet MAC address. For an OUI address, only the upper 24-bits of the address are used in the source or destination address lookup. Otherwise, all 48-bits are used in the lookup.

15.3.2.7.1.3 VLAN/Multicast Address Table Entry

Table 15-12. VLAN/Multicast Address Table Entry Bit Values

71:69	68:66	65	64	63:62	61:60	59:48	47:0
Reserved	Port Mask	Super	Reserved	Mcast Fwd State	Entry Type (11)	vlan_id	Multicast Address

Supervisory Packet (SUPER)

When set, this field indicates that the packet with a matching multicast destination address is a supervisory packet.

0: Non-supervisory packet

1: Supervisory packet

Port Mask(2:0) (PORT_MASK)

This three bit field is the port bit mask that is returned with a found multicast destination address. There may be multiple bits set indicating that the multicast packet may be forwarded to multiple ports (but not the receiving port).

Multicast Forward State (MCAST_FWD_STATE)

Multicast Forward State – Indicates the port state(s) required for the received port on a destination address lookup in order for the multicast packet to be forwarded to the transmit port(s). A transmit port must be in the Forwarding state in order to forward the packet. If the transmit port_mask has multiple set bits then each forward decision is independent of the other transmit port(s) forward decision.

00 – Forwarding

01 – Blocking/Forwarding/Learning

10 – Forwarding/Learning

11 – Forwarding

The forward state test returns a true value if both the Rx and Tx ports are in the required state.

Table Entry Type (ENTRY_TYPE)

Address entry type. Unicast or multicast determined by address bit 40.

11: VLAN address entry. Unicast or multicast determined by address bit 40.

VLAN ID (VLAN_ID)

The unique identifier for VLAN identification. This is the 12-bit VLAN ID.

Packet Address (MULTICAST_ADDRESS)

This is the 48-bit packet MAC address. For an OUI address, only the upper 24-bits of the address are used in the source or destination address lookup. Otherwise, all 48-bits are used in the lookup.

15.3.2.7.1.4 Unicast Address Table Entry

Table 15-13. Unicast Address Table Entry Bit Values

71:70	69	68	67:66	65	64	63:62	61:60	59:48	47:0
Reserved	DLR Unicast	Reserved	Port Number	Block	Secure	Unicast Type (00) or (X1)	Entry Type (01)	Reserved	Unicast Address

DLR Unicast

DLR Unicast – When set, this bit indicates that the address is a Device Level Ring (DLR) unicast address. Received packets with a matching destination address will be flooded to the vlan_member_list (minus the receive port and the host port). The port_number field is a don’t care when this bit is set. Matching packets received on port 1 egress on port 2. Matching packets received on port 2 egress on port 1. Matching packets received on port 0 egress on ports 1 and 2.

Port Number (PORT_NUMBER)

Port Number – This field indicates the port number (not port mask) that the packet with a unicast destination address may be forwarded to. Packets with unicast destination addresses are forwarded only to a single port (but not the receiving port).

Block (BLOCK)

Block – The block bit indicates that a packet with a matching source or destination address should be dropped (block the address).

0 – Address is not blocked.

1 – Drop a packet with a matching source or destination address (secure

must be zero)

If block and secure are both set, then they no longer mean block and secure. When both are set, the block and secure bits indicate that the packet is a unicast supervisory (super) packet and they determine the unicast forward state test criteria. If both bits are set then the packet is forwarded if the receive port is in the Forwarding/Blocking/Learning state. If both bits are not set then the packet is forwarded if the receive port is in the Forwarding state.

Secure (SECURE)

Secure – This bit indicates that a packet with a matching source address should be dropped if the received port number is not equal to the table entry port_number.

0 – Received port number is a don't care.

1 – Drop the packet if the received port is not the secure port for the source

address and do not update the address (block must be zero)

Unicast Type (UNICAST_TYPE)

Unicast Type – This field indicates the type of unicast address the table entry contains.

00 – Unicast address that is not ageable.

01 – Ageable unicast address that has not been touched.

10 – OUI address - lower 24-bits are don't cares (not ageable).

11 – Ageable unicast address that has been touched.

Table Entry Type (ENTRY_TYPE)

Address entry. Unicast or multicast determined by address bit 40.

01: Address entry. Unicast or multicast determined by address bit 40.

Packet Address (UNICAST_ADDRESS)

This is the 48-bit packet MAC address. All 48-bits are used in the lookup.

15.3.2.7.1.5 OUI Unicast Address Table Entry

Table 15-14. OUI Unicast Address Table Entry Bit Values

71:64	63:62	61:60	59:48	47:24	23:0
Reserved	Unicast Type (10)	Entry Type (01)	Reserved	Unicast OUI	Reserved

Unicast Type (UNICAST_TYPE)

Unicast Type – This field indicates the type of unicast address the table entry contains.

00 – Unicast address that is not ageable.

01 – Ageable unicast address that has not been touched.

10 – OUI address - lower 24-bits are don't cares (not ageable).

11 – Ageable unicast address that has been touched.

Table Entry Type (ENTRY_TYPE)

Address entry. Unicast or multicast determined by address bit 40.

01: Address entry. Unicast or multicast determined by address bit 40.

Packet Address (UNICAST_OUI)

For an OUI address, only the upper 24-bits of the address are used in the source or destination address lookup.

15.3.2.7.1.6 VLAN/Unicast Address Table Entry

Table 15-15. Unicast Address Table Entry Bit Values

71:68	67:66	65	64	63:62	61:60	59:48	47:0
Reserved	Port Number	Block	Secure	Unicast Type (00) or (X1)	Entry Type (11)	vlan_id	Unicast Address

Port Number (PORT_NUMBER)

Port Number – This field indicates the port number (not port mask) that the packet with a unicast destination address may be forwarded to. Packets with unicast destination addresses are forwarded only to a single port (but not the receiving port).]

Block (BLOCK)

Block – The block bit indicates that a packet with a matching source or destination address should be dropped (block the address).

0 – Address is not blocked.

1 – Drop a packet with a matching source or destination address (secure

must be zero)

If block and secure are both set, then they no longer mean block and secure. When both are set, the block and secure bits indicate that the packet is a unicast supervisory (super) packet and they determine the unicast forward state test criteria. If both bits are set then the packet is forwarded if the receive port is in the Forwarding/Blocking/Learning state. If both bits are not set then the packet is forwarded if the receive port is in the Forwarding state.

Secure (SECURE)

Secure – This bit indicates that a packet with a matching source address should be dropped if the received port number is not equal to the table entry port_number.

0 – Received port number is a don’t care.

1 – Drop the packet if the received port is not the secure port for the source

address and do not update the address (block must be zero)

Unicast Type (UNICAST_TYPE)

Unicast Type – This field indicates the type of unicast address the table entry contains.

00 – Unicast address that is not ageable.

01 – Ageable unicast address that has not been touched.

10 – OUI address - lower 24-bits are don’t cares (not ageable).

11 – Ageable unicast address that has been touched.

Table Entry Type (ENTRY_TYPE)

Address entry. Unicast or multicast determined by address bit 40.

11 – VLAN address entry. Unicast or multicast determined by address bit 40.

VLAN ID (VLAN_ID)

The unique identifier for VLAN identification. This is the 12-bit VLAN ID.

Packet Address (UNICAST_ADDRESS)

This is the 48-bit packet MAC address. All 48-bits are used in the lookup.

15.3.2.7.1.7 VLAN Table Entry

Table 15-16. VLAN Table Entry

71:62	61:60	59:48	47:27	26:24	23:19	18:16	15:11	10:8	7:3	2:0
Reserved	Entry Type (10)	vlan_id	Reserved	Force Untagged Egress	Reserved	Reg Mcast Flood Mask	Reserved	Unreg Mcast Flood Mask	Reserved	Vlan Member List

Table Entry Type (ENTRY_TYPE)

10: VLAN entry

VLAN ID (VLAN_ID)

The unique identifier for VLAN identification. This is the 12-bit VLAN ID.

Force Untagged Packet Egress (FORCE_UNTAGGED_EGRESS)

This field causes the packet VLAN tag to be removed on egress (except on port 0).

Registered Multicast Flood Mask (REG_MCAST_FLOOD_MASK)

Mask used for multicast when the multicast address is found

Unregistered Multicast Flood Mask (UNREG_MCAST_FLOOD_MASK)

Mask used for multicast when the multicast address is not found

VLAN Member List (VLAN_MEMBER_LIST)

This three bit field indicates which port(s) are members of the associated VLAN.

15.3.2.7.2 Packet Forwarding Processes

There are four processes that an incoming received packet may go through to determine packet forwarding. The processes are Ingress Filtering, VLAN_Aware Lookup, VLAN_Unaware Lookup, and Egress.

Packet processing begins in the Ingress Filtering process. Each port has an associated packet forwarding state that can be one of four values (Disabled, Blocked, Learning, or Forwarding). The default state for all ports is disabled. The host sets the packet forwarding state for each port. The receive packet processes are described in the following sections.

In the packet ingress process (receive packet process), there is a forward state test for unicast destination addresses and a forward state test for multicast addresses. The multicast forward state test indicates the port states required for the receiving port in order for the multicast packet to be forwarded to the transmit port(s). A transmit port must be in the Forwarding state for the packet to be forwarded for transmission. The mcast_fwd_state indicates the required port state for the receiving port as indicated in Table 15-12.

The unicast forward state test indicates the port state required for the receiving port in order to forward the unicast packet. The transmit port must be in the Forwarding state in order to forward the packet. The block and secure bits determine the unicast forward state test criteria. If both bits are set then the packet is forwarded if the receive port is in the Forwarding/Blocking/Learning state. If both bits are not set then the packet is forwarded if the receive port is in the Forwarding state. The transmit port must be in the Forwarding state regardless. The forward state test used in the ingress process is determined by the destination address packet type (multicast/unicast).

In general, packets received with errors are dropped by the address lookup engine without learning, updating, or touching the address. The error condition and the abort are indicated by the CPGMAC_SL to the ALE. Packets with errors may be passed to the host (not aborted) by a CPGMAC_SL port if the port has a set rx_cmf_en, rx_cef_en, or rx_csf_en bit(s).

Error packets that are passed to the host by the CPGMAC_SL are considered to be bypass packets by the ALE and are sent only to the host. Error packets do not learn, update, or touch addresses regardless of whether they are aborted or sent to the host. Packets with errors received by the host are forwarded as normal.

The following control bits are in the CPGMAC_SL1/2_MACCTRL register.

rx_cef_en: This CPGMAC_SL control bit enables frames that are fragments, long, jabber, CRC, code, and alignment errors to be forwarded.

rx_csf_en: This CPGMAC_SL bit enables short frames to be forwarded.

rx_cmf_en: This CPGMAC_SL control bit enables mac control frames to be forwarded.

15.3.2.7.2.1 Ingress Filtering Process

```
If (Rx port_state is Disabled)
then discard the packet

if (directed packet)
then use directed port number and go to Egress process

if ((ale_bypass or error packet) and (host port is not the receive port))
then use host portmask and go to Egress process

if (((block) and (unicast source address found)) or ((block) and (unicast destination address found)))
then discard the packet

if ((enable_rate_limit) and (rate limit exceeded) and (not rate_limit_tx)
then if (((mcast/bcast destination address found) and (not super)) or (mcast/bcast destination address not found))
  then discard the packet

if ((not forward state test valid) and (destination address found))
then discard the packet to any port not meeting the requirements
  • Unicast destination addresses use the unicast forward state test and multicast destination addresses use the multicast forward
    state test.

if ((destination address not found) and ((not transmit port forwarding) or (not receive port forwarding)))
then discard the packet to any ports not meeting the above requirements

if (source address found) and (secure) and (receive port number != port_number))
then discard the packet

if ((not super) and (drop_untagged) and ((non-tagged packet) or ((priority tagged) and not(en_vid0_mode))
then discard the packet

If (VLAN_Unaware)
  force_untagged_egress = “000000”
  reg_mcast_flood_mask = “111111”
  unreg_mcast_flood_mask = “111111”
  vlan_member_list = “111111”
else if (VLAN not found)
  force_untagged_egress = unknown_force_untagged_egress
  reg_mcast_flood_mask = unknown_reg_mcast_flood_mask
  unreg_mcast_flood_mask = unknown unreg_mcast_flood_mask
  vlan_member_list = unknown_vlan_member_list
else
  force_untagged_egress = found force_untagged_egress
  reg_mcast_flood_mask = found reg_mcast_flood_mask
  unreg_mcast_flood_mask = found unreg_mcast_flood_mask
  vlan_member_list = found vlan_member_list

if ((not super) and (vid_ingress_check) and (Rx port is not VLAN member))
then discard the packet

if ((enable_auth_mode) and (source address not found) and not(destination address found and (super)))
then discard the packet
```

```
if (destination address equals source address)
then discard the packet

if (vlan_aware) goto VLAN_Aware_Lookup process
else goto VLAN_Unaware_Lookup process
```

15.3.2.7.2.2 VLAN_Aware Lookup Process

```
if ((unicast packet) and (destination address found with or without VLAN) and dlr_unicast)
  then portmask is the vlan_member_list less the host port
  and goto Egress process

if ((unicast packet) and (destination address found with or without VLAN) and (not super))
  then portmask is the logical “AND” of the port_number and the vlan_member_list and goto Egress process

if ((unicast packet) and (destination address found with or without VLAN) and (super))
  then portmask is the port_number and goto Egress process

if (Unicast packet) # destination address not found
  then portmask is vlan_member_list less host port and goto Egress process

if ((Multicast packet) and (destination address found with or without VLAN) and (not super))
  then portmask is the logical “AND” of reg_mcast_flood_mask and found destination address/VLAN portmask (port_mask) and
  vlan_member_list and goto Egress process

if ((Multicast packet) and (destination address found with or without VLAN) and (super))
  then portmask is the port_mask and goto Egress process

if (Multicast packet) # destination address not found
  then portmask is the logical “AND” of unreg_mcast_flood_mask and vlan_member_list
  then goto Egress process

if (Broadcast packet)
  then use found vlan_member_list and goto Egress process
```

15.3.2.7.2.3 VLAN_Unaware Lookup Process

```
if ((unicast packet) and (destination address found with or without VLAN) and dlr_unicast)
  then portmask is the vlan_member_list less the host port
  and goto Egress process

if ((unicast packet) and (destination address found with or without VLAN) and (not super))
  then portmask is the logical “AND” of the port_number and the vlan_member_list and goto Egress process

if ((unicast packet) and (destination address found with or without VLAN) and (super))
  then portmask is the port_number and goto Egress process

if (Unicast packet) # destination address not found
  then portmask is vlan_member_list less host port and goto Egress process

if ((Multicast packet) and (destination address found with or without VLAN) and (not super))
  then portmask is the logical “AND” of reg_mcast_flood_mask and found destination address/VLAN portmask (port_mask) and
  vlan_member_list and goto Egress process

if ((Multicast packet) and (destination address found with or without VLAN) and (super))
  then portmask is the port_mask and goto Egress process

if (Multicast packet) # destination address not found
  then portmask is the logical “AND” of unreg_mcast_flood_mask and vlan_member_list
  then goto Egress process

if (Broadcast packet)
  then use found vlan_member_list and goto Egress process
```

15.3.2.7.2.4 Egress Process

Clear Rx port from portmask (don’t send packet to Rx port).
Clear disabled ports from portmask.
if ((enable_oui_deny) and (OUI source address not found) and (not ale_bypass) and (not error packet) and not ((mcast destination address) and (super))) then Clear host port from portmask
if ((enable_rate_limit) and (rate_limit_tx)) then if (not super) and (rate limit exceeded on any tx port) then clear rate limited tx port from portmask If address not found then super cannot be set.
If portmask is zero then discard packet
Send packet to portmask ports.

15.3.2.7.3 Learning/Updating/Touching Processes

The learning, updating, and touching processes are applied to each receive packet that is not aborted. The processes are concurrent with the packet forwarding process. In addition to the following, a packet must be received without error in order to learn/update/touch an address.

15.3.2.7.3.1 Learning Process

If (not(Learning or Forwarding) or (enable_auth_mode) or (packet error) or (no_learn)) then do not learn address
if ((Non-tagged packet) and (drop_untagged)) then do not learn address
if ((vlan_aware) and (VLAN not found) and (unknown_vlan_member_list = “000”)) then do not learn address
if ((vid_ingress_check) and (Rx port is not VLAN member) and (VLAN found)) then do not learn address
if ((source address not found) and (vlan_aware) and not(learn_no_vid)) then learn address with VLAN
if ((source address not found) and ((not vlan_aware) or (vlan_aware and learn_no_vid))) then learn address without VLAN

15.3.2.7.3.2 Updating Process

if (dlr_unicast) then do not update address
If (not(Learning or Forwarding) or (enable_auth_mode) or (packet error) or (no_sa_update)) then do not update address
if ((Non-tagged packet) and (drop_untagged)) then do not update address
if ((vlan_aware) and (VLAN not found) and (unknown_vlan_member_list = “000”)) then do not update address
if ((vid_ingress_check) and (Rx port is not VLAN member) and (VLAN found)) then do not update address
if ((source address found) and (receive port number != port_number) and (secure or block)) then do not update address

```
if ((source address found) and (receive port number != port_number))
then update address
```

15.3.2.7.3.3 Touching Process

```
if ((source address found) and (ageable) and (not touched))
then set touched
```

15.3.2.8 Packet Priority Handling

Packets are received on three ports, two of which are CPGMAC_SL Ethernet ports and the third port is the CPPI host port. Received packets have a received packet priority (0 to 7 with 7 being the highest priority). The received packet priority is determined as shown:

1. If the first packet LTYPE = 0x8100 then the received packet priority is the packet priority (VLAN tagged and priority tagged packets).
2. If the first packet LTYPE = 0x0800 and byte 14 (following the LTYPE) is equal to 0x45 then the received packet priority is the 6-bit TOS field in byte 15 (upper 6-bits) mapped through the port’s DSCP priority mapping register (IPV4 packet).
3. The received packet priority is the source (ingress) port priority (untagged non-IPV4 packet).

The received packet priority is mapped through the receive ports associated “packet priority to header packet priority mapping register” to obtain the header packet priority (the CPDMA Rx and Tx nomenclature is reversed from the CPGMAC_SL nomenclature for legacy reasons). The header packet priority is mapped through the “header priority to switch priority mapping register” to obtain the hardware switch priority (0 to 3 with 3 being the highest priority). The header packet priority is then used as the actual transmit packet priority if the VLAN information is to be sent on egress.

15.3.2.9 FIFO Memory Control

Each of the three CPSW_3G ports has an identical associated FIFO. Each FIFO contains a single logical receive (ingress) queue and four logical transmit queues (priority 0 through 3). Each FIFO memory contains 20,480 bytes (20k) total organized as 2560 by 64-bit words contained in a single memory instance. The FIFO memory is used for the associated port transmit and receive queues. The tx_max_blks field in the FIFO’s associated Max_Blks register determines the maximum number of 1k FIFO memory blocks to be allocated to the four logical transmit queues (transmit total).

The rx_max_blks field in the FIFO’s associated Max_Blks register determines the maximum number of 1k memory blocks to be allocated to the logical receive queue. The tx_max_blks value plus the rx_max_blks value must sum to 20 (the total number of blocks in the FIFO). If the sum were less than 20 then some memory blocks would be unused.The default is 17 (decimal) transmit blocks and three receive blocks. The FIFO’s follow the naming convention of the Ethernet ports.Host Port is Port0 and External Ports are Port1,2

15.3.2.10 FIFO Transmit Queue Control

There are four transmit queues in each transmit FIFO. Software has some flexibility in determining how packets are loaded into the queues and on how packet priorities are selected for transmission (how packets are removed and transmitted from queues). All ports on the switch have identical FIFO’s. For the purposes of the below the transmit FIFO is switch egress even though the port 0 transmit FIFO is connected to the CPDMA receive (also switch egress). The CPDMA nomenclature is reversed from the CPGMAC_SL nomenclature due to legacy reasons.

15.3.2.10.1 Normal Priority Mode

When operating in normal mode, lower priority frames are dropped before higher priority frames. The intention is to give preference to higher priority frames. Priority 3 is the highest priority and is allowed to fill the FIFO. Priority 2 will drop packets if the packet is going to take space in the last 2k available. Priority 1 will drop packets if the packet is going to take space in the last 4k available. Priority 0 will drop packets if the packet is going to take space in the last 6k available. If fewer than 4 priorities are to be implemented then the priorities should be mapped such that the highest priorities are used.

For example, if two priorities are going to be used then all packets should be mapped to priorities 3 and 2 and priorities 1 and 0 should be unused. Priority escalation may be used in normal priority mode if desired. Normal priority mode is configured as described below:

- Select normal priority mode by setting tx_in_sel[1:0] = 00 for all ports (default value in P0/1/2_Tx_In_Ctl)
- Configure priority mapping to use only the highest priorities if less than 4 priorities are used. Refer to the Packet Priority Handling section of this chapter.

15.3.2.10.2 Dual Mac Mode

When operating in dual mac mode the intention is to transfer packets between ports 0 and 1 and ports 0 and 2, but not between ports 1 and 2. Each CPGMAC_SL appears as a single MAC with no bridging between MAC’s. Each CPGMAC_SL has at least one unique (not the same) mac address.

Dual mac mode is configured as described below:

- Set the ale_vlan_aware bit in the ALE_Ctrl register. This bit configures the ALE to process in vlan aware mode.The CPSW_3G vlan aware bit (vlan_aware in CPSW_Ctrl) determines how packets VLAN’s are processed on CPGMAC_SL egress and does not affect how the ALE processes packets or the packet destination. The CPSW_3G vlan aware bit may be set or not as required (must be set if VLAN’s are to exit the switch).
- **Configure the Port 1 to Port 0 VLAN**

Add a VLAN Table Entry with ports 0 and 1 as members (clear the flood masks).

Add a VLAN/Unicast Address Table Entry with the Port1/0 VLAN and a port number of 0. Packets received on port 1 with this unicast address will be sent only to port 0 (egress). If multiple mac addresses are desired for this port then multiple entries of this type may be configured.

- **Configure the Port 2 to Port 0 VLAN**

Add a VLAN Table Entry with ports 0 and 2 as members (clear the flood masks).

Add a VLAN/Unicast Address Table Entry with the Port2/0 VLAN and a port number of 0. Packets received on port 2 with this unicast address will be sent only to port 0 (egress). If multiple mac addresses are desired for this port then multiple entries of this type may be configured.

- Packets from the host (port 0) to ports 1 and 2 should be directed. If directed packets are not desired then VLAN with addresses can be added for both destination ports.
- Select the dual mac mode on the port 0 FIFO by setting tx_in_sel[1:0] = 01 in P0_Tx_In_Ctl. The intention of this mode is to allow packets from both ethernet ports to be written into the FIFO without one port starving the other port.
- The priority levels may be configured such that packets received on port 1 egress on one CPDMA RX channel while packets received on port 2 egress on a different CPDMA RX channel.

15.3.2.10.3 Rate Limit Mode

Rate-limit mode is intended to allow some CPDMA transmit (switch ingress) channels and some CPGMAC_SL FIFO priorities (switch egress) to be rate-limited. Non rate-limited traffic (bulk traffic) is allowed on non rate-limited channels and FIFO priorities. The bulk traffic does not impact the rate-limited traffic. Rate-limited traffic must be configured to be sent to rate-limited queues (via packet priority handling).

The allocated rates for rate-limited traffic must not be oversubscribed. For example, if port 1 is sending 15% rate limited traffic to port 2 priority 3, and port 0 is also sending 10% rate-limited traffic to port 2 priority 3, then the port 2 priority 3 egress rate must be configured to be 25% plus a percent or two for margin. The switch must be configured to allow some percentage of non rate-limited traffic. Non-rate-limited traffic must be configured to be sent to non rate-limited queues. No packets from the host should be dropped, but non rate-limited traffic received on an ethernet port can be dropped. Rate-limited mode is configured as shown:

- Set tx_in_sel[1:0] = 10 in P1/2_Tx_In_Ctl to enable ports 1 and 2 transmit FIFO inputs to be configured for rate-limiting queues. Enabling a queue to be rate-limiting with this field affects only the packet being loaded into the FIFO, it does not configure the transmit for queue shaping.
- Configure the number of rate-limited queues for port 1 and 2 transmit FIFO’s by setting the tx_rate_en[3:0] field in P1/2_Tx_In_Ctl. Rate limited queues must be the highest number. For example, if there are two rate limited queues then 1100 would be written to this field for priorities 3 and 2. This field enables the FIFO to allow rate-limited traffic into rate-limited queues while discriminating against non rate-limited queues.
- Set p1_priN_shape_en and p2_priN_shape_en in the CPSW_3G PTYPE register. These bits determine which queues actually shape the output data stream. In general, the same priorities that are set in tx_rate_en are set in these bits as well, but the FIFO input and output enable bits are separate to allow rate-limiting from the host to non shaped channels if desired.
  When queue shaping is not enabled for a queue then packets are selected for egress based on priority. When queue shaping is enabled then packets are selected for egress based on queue percentages. If shaping is required on a single queue then it must be priority 3 (priorities 2, 1 and 0 are strict priority). If shaping is required on two queues then it must be on priorities 2 and 3 (priorities 1 and 0 are strict priority). If shaping is required on three queues then it must be priorities 3, 2, and 1 (priority 0 would then get the leftovers). Priority shaping follows the requirements in the IEEE P802.1Qav/D6.0 specification. Priority shaping is not compatible with priority escalation (escalation must be disabled).
- P0_Tx_In_Ctl[1:0] should remain at the default 00 value. Port 0 egress (CPDMA RX) should not be rate-limited.
- The CPDMA is configured for rate-limited transmit (switch ingress) channels by setting the highest bits of the tx_rlim[7:0] field in the CPDMA DMA_Ctrl register. If there are two rate limited channels then tx_rlim[7:0] = 11000000 (the rate limited channels must be the highest priorities). Also, tx_ptype in the DMA_Ctrl register must be set (fixed priority mode). Rate limited channels must go to rate-limited FIFO queues, and the FIFO queue rate must not be oversubscribed.

15.3.2.11 Packet Padding

VLAN tagged ingress packets of 64 to 67-bytes will be padded to 64-bytes on egress (all ports) if the VLAN is removed on egress.

15.3.2.12 Flow Control

There are two types of switch flow control – CPPI port flow control and Ethernet port flow control. The CPPI and Ethernet port naming conventions for data flow into and out of the switch are reversed. For the CPPI port (port 0), transmit operations move packets from external memory into the switch and then out to either or both Ethernet transmit ports (ports 1 and 2). CPPI receive operations move packets that were received on either or both Ethernet receive ports to external memory.

15.3.2.12.1 CPPI Port Flow Control

The CPPI port has flow control available for transmit (switch ingress). CPPI receive operations (switch egress) do not require flow control. CPPI Transmit flow control is initiated when enabled and triggered. CPPI transmit flow control is enabled by setting the p0_flow_en bit in the **CPSW_Flow_Ctrl** register. CPPI transmit flow control is enabled by default on reset because host packets should not be dropped in any mode of operation.

15.3.2.12.2 Ethernet Port Flow Control

The Ethernet ports have flow control available for transmit and receive. Transmit flow control stops the Ethernet port from transmitting packets to the wire (switch egress) in response to a received pause frame. Transmit flow control does not depend on FIFO usage.

The ethernet ports have flow control available for receive operations (packet ingress). Ethernet port receive flow control is initiated when enabled and triggered. Packets received on an ethernet port can be sent to the other ethernet port or the CPPI port (or both). Each destination port can trigger the receive ethernet port flow control. An ethernet destination port triggers another ethernet receive flow control when the destination port is full.

When a packet is received on an ethernet port interface with enabled flow control the below occurs:

- The packet will be sent to all ports that currently have room to take the entire packet.
- The packet will be retried until successful to all ports that indicate they don’t have room for the packet.

The flow control trigger to the CPGMAC_SL will be asserted until the packet has been sent, and there is room in the logical receive FIFO for packet runout from another flow control trigger (**rx_pkt_cnt** = 0). Ethernet port receive flow control is disabled by default on reset. Ethernet port receive flow control requires that the **rx_flow_en** bit in the associated CPGMAC_SL be set to one.

When receive flow control is enabled on a port, the port’s associated FIFO block allocation must be adjusted. The port RX allocation must increase from the default three blocks to accommodate the flow control runout. A corresponding decrease in the TX block allocation is required. If a sending port ignores a pause frame then packets may overrun on receive (and be dropped) but will not be dropped on transmit. If flow control is disabled for gmii ports, then any packets that are dropped are dropped on transmit and not on receive.

15.3.2.12.2.1 Receive Flow Control

When enabled and triggered, receive flow control is initiated to limit the CPGMAC_SL from further frame reception. Half-duplex mode receive flow control is collision based while full duplex mode issues 802.3X pause frames. In either case, receive flow control prevents frame reception by issuing the flow control appropriate for the current mode of operation. Receive flow control is enabled by the **rx_flow_en** bit in the **MACCTRL** register. Receive flow control is triggered (when enabled) when the **RX_FLOW_TRIGGER** input is asserted. The CPGMAC_SL is configured for collision or IEEE 802.3X flow control via the **fullduplex** bit in the **MACCTRL** register.

15.3.2.12.2.1.1 Collision Based Receive Buffer Flow Control

Collision-based receive buffer flow control provides a means of preventing frame reception when the port is operating in half-duplex mode (**fullduplex** is cleared in **MACCTRL)**. When receive flow control is enabled and triggered, the port will generate collisions for received frames. The jam sequence transmitted will be the twelve byte sequence C3.C3.C3.C3.C3.C3.C3.C3.C3.C3.C3.C3 (hex). The jam sequence will begin no later than approximately as the source address starts to be received. Note that these forced collisions will not be limited to a maximum of 16 consecutive collisions, and are independent of the normal back-off algorithm. Receive flow control does not depend on the value of the incoming frame destination address. A collision will be generated for any incoming packet, regardless of the destination address.

15.3.2.12.2.1.2 IEEE 802.3X Based Receive Flow Control

IEEE 802.3x based receive flow control provides a means of preventing frame reception when the port is operating in full-duplex mode (**fullduplex** is set in **MACCTRL)**. When receive flow control is enabled and triggered, the port will transmit a pause frame to request that the sending station stop transmitting for the period indicated within the transmitted pause frame.

The CPGMAC_SL will transmit a pause frame to the reserved multicast address at the first available opportunity (immediately if currently idle, or following the completion of the frame currently being transmitted). The pause frame will contain the maximum possible value for the pause time (0xFFFF). The MAC will count the receive pause frame time (decrements 0xFF00 downto zero) and retransmit an outgoing pause frame if the count reaches zero. When the flow control request is removed, the MAC will transmit a pause frame with a zero pause time to cancel the pause request.

Note that transmitted pause frames are only a request to the other end station to stop transmitting. Frames that are received during the pause interval will be received normally (provided the Rx FIFO is not full).

Pause frames will be transmitted if enabled and triggered regardless of whether or not the port is observing the pause time period from an incoming pause frame.

The CPGMAC_SL will transmit pause frames as described below:

- The 48-bit reserved multicast destination address 01.80.C2.00.00.01.
- The 48-bit source address — SL_SA(47:0).
- The 16-bit length/type field containing the value 88.08
- The 16-bit pause opcode equal to 00.01
- The 16-bit pause time value FF.FF. A pause-quantum is 512 bit-times. Pause frames sent to cancel a pause request will have a pause time value of 00.00.
- Zero padding to 64-byte data length (The CPGMAC_SL will transmit only 64 byte pause frames).
- The 32-bit frame-check sequence (CRC word).

All quantities above are hexadecimal and are transmitted most-significant byte first. The least-significant bit is transferred first in each byte.

If **rx_flow_en** is cleared to zero while the pause time is nonzero, then the pause time will be cleared to zero and a zero count pause frame will be sent.

15.3.2.12.2.2 Transmit Flow Control

Incoming pause frames are acted upon, when enabled, to prevent the CPGMAC_SL from transmitting any further frames. Incoming pause frames are only acted upon when the **fullduplex** and **tx_flow_en** bits in the **MACCTRL** register are set. Pause frames are not acted upon in half-duplex mode. Pause frame action will be taken if enabled, but normally the frame will be filtered and not transferred to memory.

MAC control frames will be transferred to memory if the **rx_cmf_en** (Copy MAC Frames) bit in the **MACCTRL** register is set. The **tx_flow_en** and **fullduplex** bits effect whether or not MAC control frames are acted upon, but they have no effect upon whether or not MAC control frames are transferred to memory or filtered.

Pause frames are a subset of MACCTRL Frames with an opcode field=0x0001. Incoming pause frames will only be acted upon by the port if:

- **tx_flow_en** is set in **MACCTRL**, and
- the frame’s length is 64 to **rx_maxlen** bytes inclusive, and
- the frame contains no crc error or align/code errors.

The pause time value from valid frames will be extracted from the two bytes following the opcode. The pause time will be loaded into the port’s transmit pause timer and the transmit pause time period will begin.

If a valid pause frame is received during the transmit pause time period of a previous transmit pause frame then:

- if the destination address is not equal to the reserved multicast address or any enabled or disabled unicast address, then the transmit pause timer will immediately expire, or
- if the new pause time value is zero then the transmit pause timer will immediately expire, else
- the port transmit pause timer will immediately be set to the new pause frame pause time value. (Any remaining pause time from the previous pause frame will be discarded).

If **tx_flow_en** in **MACCTRL** is cleared, then the pause-timer will immediately expire.

The port will not start the transmission of a new data frame any sooner than 512-bit times after a pause frame with a non-zero pause time has finished being received (**GMII_RXDV** going inactive). No transmission will begin until the pause timer has expired (the port may transmit pause frames in order to initiate outgoing flow control). Any frame already in transmission when a pause frame is received will be completed and unaffected.

Incoming pause frames consist of the below:

- A 48-bit destination address equal to:
- The reserved multicast destination address 01.80.C2.00.00.01, or
- The SL_SA(47:0) input mac source address.
- The 48-bit source address of the transmitting device.
- The 16-bit length/type field containing the value 88.08
- The 16-bit pause opcode equal to 00.01
- The 16-bit pause_time. A pause-quantum is 512 bit-times.
- Padding to 64-byte data length.
- The 32-bit frame-check sequence (CRC word).

All quantities above are hexadecimal and are transmitted most-significant byte first. The least-significant bit is transferred first in each byte.

The padding is required to make up the frame to a minimum of 64 bytes. The standard allows pause frames longer than 64 bytes to be discarded or interpreted as valid pause frames. The CPGMAC_SL will recognize any pause frame between 64 bytes and **rx_maxlen** bytes in length.

15.3.2.13 Packet Drop Interface

The packet drop interface supports an external packet drop engine. The port 1 (and port 2) CPGMAC_SL receive FIFO VBUSP interface signals are CPSW_3G outputs. The receive packet interface has an associated packet drop input P1_RFIFO_DROP (P2_RFIFO_DROP). An external packet drop engine may “snoop” the received packet header and data to determine whether or not the packet should be dropped.

If the packet is to be dropped the external logic must assert the drop signal by no later than the second clock after the end of packet (or abort) indication from the CPGMAC_SL. The drop signal should remain asserted until the second clock after the end of packet (or abort) indication. If the packet is not to be dropped then the drop signal should remain deasserted. The CPGMAC_SL section contains more information on the receive FIFO VBUSP interface signals and end of packet indication.

15.3.2.14 Short Gap

The port 1 (and port 2) transmit inter-packet gap (IPG) may be shortened by eight bit times when enabled and triggered. The **tx_short_gap_en** bit in the **SL1_MACCTRL (SL2_MACCTRL)** register enables the gap to be shortened when triggered. The condition is triggered when the port 1 (port 2) transmit FIFO has a user defined number of FIFO blocks used. The port 1 transmit FIFO blocks used determines if the port 1 gap is shortened, and the port 2 transmit FIFO blocks used determines if the port 2 gap is shortened. The **CPSW_Gap_Thresh** register value determines the port 1 short gap threshold, and the **CPSW_Gap_Thresh** register value determines the port 2 short gap threshold.

15.3.2.15 Switch Latency

The CPSW_3G is a store and forward switch. The switch latency is defined as the amount of time between the end of packet reception of the received packet to the start of the output packet transmit.

Mode	Latency
Gig (1000)	880ns
100	1.3us
10	6.5us

15.3.2.16 Emulation Control

The emulation control input (EMUSUSP) and submodule emulation control registers allow CPSW_3G operation to be completely or partially suspended. There are three CPSW_3G submodules that contain emulation control registers (CPGMAC_SL1, CPGMAC_SL2, and CPDMA). The submodule emulation control registers must be accessed to facilitate CPSW_3G emulation control. The CPSW_3G module enters the emulation suspend state if all three submodules are configured for emulation suspend and the emulation suspend input is asserted.

A partial emulation suspend state is entered if one or two submodules is configured for emulation suspend and the emulation suspend input is asserted. Emulation suspend occurs at packet boundaries. The emulation control feature is implemented for compatibility with other peripherals.

CPGMAC_SL Emulation Control

The emulation control input (**TBEMUSUP**) and register bits (**soft** and **free** in the **EMCTRL** register) allow CPGMAC_SL operation to be suspended. When the emulation suspend state is entered, the CPGMAC_SL will stop processing receive and transmit frames at the next frame boundary. Any frame currently in reception or transmission will be completed normally without suspension. For receive, frames that are detected by the CPGMAC_SL after the suspend state is entered are ignored. Emulation control is implemented for compatibility with other peripherals.

CPDMA Emulation Control

The emulation control input (**TBEMUSUP**) and register bits (**soft** and **free** in the **EMCTRL** register) allow CPDMA operation to be suspended. When the emulation suspend state is entered, the CPDMA will stop processing receive and transmit frames at the next frame boundary. Any frame currently in reception or transmission will be completed normally without suspension. For transmission, any complete or partial frame in the tx cell fifo will be transmitted. For receive, frames that are detected by the CPDMA after the suspend state is entered are ignored. No statistics will be kept for ignored frames. Emulation control is implemented for compatibility with other peripherals.

The following table shows the operations of the emulation control input and register bits:

Table 15-17. Operations of Emulation Control Input and Register Bits

EMUSUSP	soft	free	Description
0	X	X	Normal Operation
1	0	0	Normal Operation
1	1	0	Emulation Suspend
1	X	1	Normal Operation

Emulation Suspend Input

The emulation suspend input described above comes from the Debug Subsystem. See Chapter 31, *Debug Subsystem*, to enable an emulation suspend event input for the Ethernet Subsystem (EMAC).

15.3.2.17 Software IDLE

The submodule software idle register bits enable CPSW_3G operation to be completely or partially suspended by software control. There are three CPSW_3G submodules that contain software idle register bits (CPGMAC_SL1, CPGMAC_SL2, and CPDMA). Each of the three submodules may be individually commanded to enter the idle state. The idle state is entered at packet boundaries, and no further packet operations will occur on an idled submodule until the idle command is removed. The CPSW_3G module enters the idle state when all three submodules are commanded to enter and have entered the idle state. Idle status is determined by reading or polling the three submodule idle bits. The CPSW_3G is in the idle state when all three submodules are in the idle state. The **CPSW_Soft_Idle** bit may be set if desired after the submodules are in the idle state. The **CPSW_Soft_Idle** bit causes packets to not be transferred from one FIFO to another FIFO internal to the switch.

15.3.2.18 Software Reset

The CPSW_3G software reset register, CPSW_3GSS software reset register and the three submodule software reset registers enable the CPSW_3GSS to be reset by software. There are three CPSW_3G submodules that contain software reset registers (CPGMAC_SL1, CPGMAC_SL2, and CPDMA). Each of the three submodules may be individually commanded to be reset by software.

For the CPDMA, the reset state is entered at packet boundaries, at which time the CPDMA reset occurs. The CPGMAC_SL soft reset is immediate. Submodule reset status is determined by reading or polling the submodule reset bit. If the submodule reset bit is read as a one, then the reset process has not yet completed. The submodule soft reset process could take up to 2ms each. The reset has completed if the submodule reset bit is read as a zero.

After all three submodules (in any order) have been reset and a read of each submodule reset bit indicates that the reset process is complete, the CPSW_3G software reset register bit may be written to complete the CPSW_3G module software reset operation. The CPSW_3G software reset bit controls the reset of the FIFO's, the statistics submodule, and the address lookup engine (ALE). The CPSW_3G software reset is immediate and will be indicated by reading a zero from the soft reset bit.

The CPSW_3GSS software reset bit controls the reset of the INT, REGS and CPPI. The CPSW_3GSS software reset is immediate and will be indicated by reading a zero from the soft reset bit.

15.3.2.19 FIFO Loopback

FIFO loopback mode is entered when the fifo_loopback bit in the CPSW_Ctrl register is set. FIFO loopback mode causes packets received on a port to be turned around and transmitted back on the same port. Port 0 receive is fixed on channel zero in FIFO loopback mode. The RXSOFOVERRUN statistic is incremented for each packet sent in FIFO loopback mode. Packets sent in with errors are returned with errors (they are not dropped). FIFO loopback is intended as a simple mechanism for test purposes. FIFO loopback should be performed in fullduplex mode only.

15.3.2.20 CPSW_3G Network Statistics

The CPSW_3G has a set of statistics that record events associated with frame traffic on selected switch ports. The statistics values are cleared to zero 38 clocks after the rising edge of VBUSP_RST_N. When one or more port enable bits (stat_port_en[2:0]) are set, all statistics registers are write to decrement. The value written will be subtracted from the register value with the result being stored in the register. If a value greater than the statistics value is written, then zero will be written to the register (writing 0xffffffff will clear a statistics location).

When all port enable bits are cleared to zero, all statistics registers are read/write (normal write direct, so writing 0x00000000 will clear a statistics location). All write accesses must be 32-bit accesses. In the below statistics descriptions, “the port” refers to any enabled port (with a corresponding set stat_port_en[2:0] bit).

The statistics interrupt (STAT_PEND) will be issued if enabled when any statistics value is greater than or equal to 0x80000000. The statistics interrupt is removed by writing to decrement any statistics value greater than 0x80000000. The statistics are mapped into internal memory space and are 32-bits wide. All statistics rollover from 0xFFFFFFFF to 0x00000000.

15.3.2.20.1 Rx-only Statistics Descriptions

15.3.2.20.1.1 Good Rx Frames (Offset = 0h)

The total number of good frames received on the port. A good frame is defined to be:

- Any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode
- Had a length of 64 to rx_maxlen bytes inclusive
- Had no CRC error, alignment error or code error.

For definitions of alignment, code and CRC errors, see Section 15.3.2.20.1.5, *Rx CRC Errors* and Section 15.3.2.20.1.6, *Rx Align/Code Errors*. Overruns have no effect upon this statistic.

15.3.2.20.1.2 Broadcast Rx Frames (Offset = 4h)

The total number of good broadcast frames received on the port. A good broadcast frame is defined to be:

- Any data or MAC control frame which was destined for only address 0xFFFFFFFFFFFF
- Had a length of 64 to rx_maxlen bytes inclusive
- Had no CRC error, alignment error or code error.

See the Rx Align/Code Errors and Rx CRC errors statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

15.3.2.20.1.3 Multicast Rx Frames (Offset = 8h)

The total number of good multicast frames received on the port. A good multicast frame is defined to be:

- Any data or MAC control frame which was destined for any multicast address other than 0xFFFFFFFFFFFF
- Had a length of 64 to rx_maxlen bytes inclusive
- Had no CRC error, alignment error or code error

See the Rx Align/Code Errors and Rx CRC errors statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

15.3.2.20.1.4 Pause Rx Frames (Offset = Ch)

The total number of IEEE 802.3X pause frames received by the port (whether acted upon or not). Such a frame:

- Contained any unicast, broadcast, or multicast address
- Contained the length/type field value 88.08 (hex) and the opcode 0x0001
- Was of length 64 to rx_maxlen bytes inclusive
- Had no CRC error, alignment error or code error
- Pause-frames had been enabled on the port (tx_flow_en = 1).

The port could have been in either half or full-duplex mode.

See the Rx Align/Code Errors and Rx CRC errors statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

15.3.2.20.1.5 Rx CRC Errors (Offset = 10h)

The total number of frames received on the port that experienced a CRC error. Such a frame:

- Was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode
- Was of length 64 to rx_maxlen bytes inclusive
- Had no code/align error,
- Had a CRC error

Overruns have no effect upon this statistic.

A CRC error is defined to be:

- A frame containing an even number of nibbles
- Failing the Frame Check Sequence test

15.3.2.20.1.6 Rx Align/Code Errors (Offset = 14h)

The total number of frames received on the port that experienced an alignment error or code error. Such a frame:

- Was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode

- Was of length 64 to rx_maxlen bytes inclusive
- Had either an alignment error or a code error

Overruns have no effect upon this statistic.

An alignment error is defined to be:

- A frame containing an odd number of nibbles
- Failing the Frame Check Sequence test if the final nibble is ignored

A code error is defined to be a frame which has been discarded because the port's MRXER pin driven with a one for at least one bit-time's duration at any point during the frame's reception.

Note: RFC 1757 etherStatsCRCAlignErrors Ref. 1.5 can be calculated by summing Rx Align/Code Errors and Rx CRC errors.

15.3.2.20.1.7 Oversize Rx Frames (Offset = 18h)

The total number of oversized frames received on the port. An oversized frame is defined to be:

- Was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode
- Was greater than rx_maxlen in bytes
- Had no CRC error, alignment error or code error

See the Rx Align/Code Errors and Rx CRC errors statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

15.3.2.20.1.8 Rx Jabbers (Offset = 1Ch)

The total number of jabber frames received on the port. A jabber frame:

- Was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode
- Was greater than rx_maxlen in bytes
- Had no CRC error, alignment error or code error

See the Rx Align/Code Errors and Rx CRC errors statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

15.3.2.20.1.9 Undersize (Short) Rx Frames (Offset = 20h)

The total number of undersized frames received on the port. An undersized frame is defined to be:

- Was any data frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode
- Was greater than rx_maxlen in bytes
- Had no CRC error, alignment error or code error

See the Rx Align/Code Errors and Rx CRC errors statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

15.3.2.20.1.10 Rx Fragments (Offset = 24h)

The total number of frame fragments received on the port. A frame fragment is defined to be:

- Any data frame (address matching does not matter)
- Less than 64 bytes long
- Having a CRC error, an alignment error, or a code error
- Not the result of a collision caused by half duplex, collision based flow control

See the Rx Align/Code Errors and Rx CRC errors statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

15.3.2.20.1.11 Rx Start of Frame Overruns (Offset = 84h)

The total number of frames received on the port that had a CPDMA start of frame (SOF) overrun or were dropped by due to FIFO resource limitations. SOF overrun frame is defined to be:

- Any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode
- Any length (including less than 64 bytes and greater than rx_maxlen bytes)
- The CPDMA had a start of frame overrun or the packet was dropped due to FIFO resource limitations

15.3.2.20.1.12 Rx Middle of Frame Overruns (Offset = 88h)

The total number of frames received on the port that had a CPDMA middle of frame (MOF) overrun. MOF overrun frame is defined to be:

- Any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode
- Any length (including less than 64 bytes and greater than rx_maxlen bytes)
- The CPDMA had a middle of frame overrun

15.3.2.20.1.13 Rx DMA Overruns (Offset = 8Ch)

The total number of frames received on the port that had either a DMA start of frame (SOF) overrun or a DMA MOF overrun. An Rx DMA overrun frame is defined to be:

- Any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode
- Any length (including less than 64 bytes and greater than rx_maxlen bytes)
- The CPGMAC_SL was unable to receive it because it did not have the DMA buffer resources to receive it (zero head descriptor pointer at the start or during the middle of the frame reception)

CRC errors, alignment errors and code errors have no effect upon this statistic.

15.3.2.20.1.14 Rx Octets (Offset = 30h)

The total number of bytes in all good frames received on the port. A good frame is defined to be:

- Any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode
- Of length 64 to rx_maxlen bytes inclusive
- Had no CRC error, alignment error or code error

See the Rx Align/Code Errors and Rx CRC errors statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

15.3.2.20.1.15 Net Octets (Offset = 80h)

The total number of bytes of frame data received and transmitted on the port. Each frame counted:

- was any data or MAC control frame destined for any unicast, broadcast or multicast address (address match does not matter)
- Any length (including less than 64 bytes and greater than rx_maxlen bytes)

Also counted in this statistic is:

- Every byte transmitted before a carrier-loss was experienced
- Every byte transmitted before each collision was experienced, (i.e. multiple retries are counted each time)
- Every byte received if the port is in half-duplex mode until a jam sequence was transmitted to initiate flow control. (The jam sequence was not counted to prevent double-counting)

Error conditions such as alignment errors, CRC errors, code errors, overruns and underruns do not affect the recording of bytes by this statistic.

The objective of this statistic is to give a reasonable indication of ethernet utilization

15.3.2.20.2 Tx-only Statistics Descriptions

The maximum and minimum transmit frame size is software controllable.

15.3.2.20.2.1 Good Tx Frames (Offset = 34h)

The total number of good frames received on the port. A good frame is defined to be:

- Any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode
- Any length
- Had no late or excessive collisions, no carrier loss and no underrun

15.3.2.20.2.2 Broadcast Tx Frames (Offset = 38h)

The total number of good broadcast frames received on the port. A good broadcast frame is defined to be:

- Any data or MAC control frame which was destined for only address 0xFFFFFFFFFFFF
- Any length
- Had no late or excessive collisions, no carrier loss and no underrun

15.3.2.20.2.3 Multicast Tx Frames (Offset = 3Ch)

The total number of good multicast frames received on the port. A good multicast frame is defined to be:

- Any data or MAC control frame which was destined for any multicast address other than 0xFFFFFFFFFFFF
- Any length
- Had no late or excessive collisions, no carrier loss and no underrun

15.3.2.20.2.4 Pause Tx Frames (Offset = 40h)

This statistic indicates the number of IEEE 802.3X pause frames transmitted by the port.

Pause frames cannot underrun or contain a CRC error because they are created in the transmitting MAC, so these error conditions have no effect upon the statistic. Pause frames sent by software will not be included in this count.

Since pause frames are only transmitted in full duplex carrier loss and collisions have no effect upon this statistic.

Transmitted pause frames are always 64 byte multicast frames so will appear in the Tx Multicast Frames and 64octet Frames statistics.

15.3.2.20.2.5 Collisions (Offset = 48h)

This statistic records the total number of times that the port experienced a collision. Collisions occur under two circumstances.

1. When a transmit data or MAC control frame:
   - Was destined for any unicast, broadcast or multicast address
   - Was any size
   - Had no carrier loss and no underrun
   - Experienced a collision. A jam sequence is sent for every non-late collision, so this statistic will increment on each occasion if a frame experiences multiple collisions (and increments on late collisions)

   CRC errors have no effect upon this statistic.
2. When the port is in half-duplex mode, flow control is active, and a frame reception begins.

15.3.2.20.2.6 Single Collision Tx Frames (Offset = 4Ch)

The total number of frames transmitted on the port that experienced exactly one collision. Such a frame:

- Was any data or MAC control frame destined for any unicast, broadcast or multicast address
- Was any size
- Had no carrier loss and no underrun
- Experienced one collision before successful transmission. The collision was not late.

CRC errors have no effect upon this statistic.

15.3.2.20.2.7 Multiple Collision Tx Frames (Offset = 50h)

The total number of frames transmitted on the port that experienced multiple collisions. Such a frame:

- Was any data or MAC control frame destined for any unicast, broadcast or multicast address
- Was any size
- Had no carrier loss and no underrun
- Experienced 2 to 15 collisions before being successfully transmitted. None of the collisions were late.

CRC errors have no effect upon this statistic.

15.3.2.20.2.8 Excessive Collisions (Offset = 54h)

The total number of frames for which transmission was abandoned due to excessive collisions. Such a frame:

- Was any data or MAC control frame destined for any unicast, broadcast or multicast address
- Was any size
- Had no carrier loss and no underrun
- Experienced 16 collisions before abandoning all attempts at transmitting the frame. None of the collisions were late.

CRC errors have no effect upon this statistic.

15.3.2.20.2.9 Late Collisions (Offset = 58h)

The total number of frames on the port for which transmission was abandoned because they experienced a late collision. Such a frame:

- Was any data or MAC control frame destined for any unicast, broadcast or multicast address
- Was any size
- Experienced a collision later than 512 bit-times into the transmission. There may have been up to 15 previous (non-late) collisions which had previously required the transmission to be re-attempted. The Late Collisions statistic dominates over the single, multiple and excessive Collisions statistics - if a late collision occurs the frame will not be counted in any of these other three statistics.

CRC errors have no effect upon this statistic.

15.3.2.20.2.10 Tx Underrun (Offset = 5Ch)

There should be no transmitted frames that experience underrun.

15.3.2.20.2.11 Deferred Tx Frames (Offset = 44h)

The total number of frames transmitted on the port that first experienced deferment. Such a frame:

- Was any data or MAC control frame destined for any unicast, broadcast or multicast address
- Was any size
- Had no carrier loss and no underrun
- Experienced no collisions before being successfully transmitted

- Found the medium busy when transmission was first attempted, so had to wait.

CRC errors have no effect upon this statistic.

15.3.2.20.2.12 Carrier Sense Errors (Offset = 60h)

The total number of frames received on the port that had a CPDMA middle of frame (MOF) overrun. MOF overrun frame is defined to be:

- Was any data or MAC control frame destined for any unicast, broadcast or multicast address
- Was any size
- The carrier sense condition was lost or never asserted when transmitting the frame (the frame is not retransmitted). This is a transmit only statistic. Carrier Sense is a don't care for received frames. Transmit frames with carrier sense errors are sent until completion and are not aborted.

CRC errors have no effect upon this statistic.

15.3.2.20.2.13 Tx Octets (Offset = 64h)

The total number of bytes in all good frames transmitted on the port. A good frame is defined to be:

- Any data or MAC control frame which was destined for any unicast, broadcast or multicast address
- Was any size
- Had no late or excessive collisions, no carrier loss and no underrun.

15.3.2.20.3 Rx- and Tx-Shared Statistics Descriptions

15.3.2.20.3.1 Rx + Tx 64 Octet Frames (Offset = 68h)

The total number of 64-byte frames received and transmitted on the port. Such a frame is defined to be:

- Any data or MAC control frame which was destined for any unicast, broadcast or multicast address
- Did not experience late collisions, excessive collisions, or carrier sense error
- Was exactly 64 bytes long. (If the frame was being transmitted and experienced carrier loss that resulted in a frame of this size being transmitted, then the frame will be recorded in this statistic).

CRC errors, code/align errors and overruns do not affect the recording of frames in this statistic.

15.3.2.20.3.2 Rx + Tx 65–127 Octet Frames (Offset = 6Ch)

The total number of frames of size 65 to 127 bytes received and transmitted on the port. Such a frame is defined to be:

- Any data or MAC control frame which was destined for any unicast, broadcast or multicast address
- Did not experience late collisions, excessive collisions, or carrier sense error
- Was 65 to 127 bytes long

CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.

15.3.2.20.3.3 Rx + Tx 128–255 Octet Frames (Offset = 70h)

The total number of frames of size 128 to 255 bytes received and transmitted on the port. Such a frame is defined to be:

- Any data or MAC control frame which was destined for any unicast, broadcast or multicast address
- Did not experience late collisions, excessive collisions, or carrier sense error
- Was 128 to 255 bytes long

CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.

15.3.2.20.3.4 Rx + Tx 256–511 Octet Frames (Offset = 74h)

The total number of frames of size 256 to 511 bytes received and transmitted on the port. Such a frame is defined to be:

- Any data or MAC control frame which was destined for any unicast, broadcast or multicast address
- Did not experience late collisions, excessive collisions, or carrier sense error
- Was 256 to 511 bytes long

CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.

15.3.2.20.3.5 Rx + Tx 512–1023 Octet Frames (Offset = 78h)

The total number of frames of size 512 to 1023 bytes received and transmitted on the port. Such a frame is defined to be:

- Any data or MAC control frame which was destined for any unicast, broadcast or multicast address
- Did not experience late collisions, excessive collisions, or carrier sense error
- Was 512 to 1023 bytes long

CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.

15.3.2.20.3.6 Rx + Tx 1024_Up Octet Frames (Offset = 7Ch)

The total number of frames of size 1024 to rx_maxlen bytes for receive or 1024 up for transmit on the port. Such a frame is defined to be:

- Any data or MAC control frame which was destined for any unicast, broadcast or multicast address
- Did not experience late collisions, excessive collisions, or carrier sense error
- Was 1024 to rx_maxlen bytes long on receive, or any size on transmit

CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.

Table 15-18. Rx Statistics Summary

Rx Statistic	Frame/ Oct	Rx/ Rx+Tx	Frame Type					Frame Size (bytes)								Event				
			MAC control		Data															
			Pause frame	Non-pause	Multi-cast	Broad-cast	Uni-cast	<64	64	65-127	128-255	256-511	512-1023	1024-rx_maxlen	>rx_maxlen	Flow Coll.	CRC Error	Align/ Code	Over-run	Addr. Disc.
Good Rx Frames	F	Rx	(y|	y|	y|	y|	y)	n	(y|	y|	y|	y|	y|	y)	n	-	n	n	-	n
Broadcast Rx Frames	F	Rx	(%|	%|	n	y)	n	n	(y|	y|	y|	y|	y|	y)	n	-	n	n	-	n
Multicast Rx Frames	F	Rx	(%|	%|	y)	n	n	n	(y|	y|	y|	y|	y|	y)	n	-	n	n	-	n
Pause Rx Frames	F	Rx	y	n	n	n	n	n	(y|	y|	y|	y|	y|	y)	n	-	n	n	-	-
Rx CRC Errors	F	Rx	(y|	y|	y|	y|	y)	n	(y|	y|	y|	y|	y|	y)	n	-	y	n	-	n
Rx Align/Code Errors	F	Rx	(y|	y|	y|	y|	y)	n	(y|	y|	y|	y|	y|	y)	n	-	-	y	-	n
Oversized Rx Frames	F	Rx	(y|	y|	y|	y|	y)	n	n	n	n	n	n	n	y	-	n	n	-	n
Rx Jabbers	F	Rx	(y|	y|	y|	y|	y)	n	n	n	n	n	n	n	y	-	(y|	y)	-	n
Undersized Rx Frames	F	Rx	n	n	(y|	y|	y)	y	n	n	n	n	n	n	n	-	n	n	-	n
Rx Fragments	F	Rx	n	n	(y|	y|	y)	y^^	n	n	n	n	n	n	n	-	(y|	y)	-	-
Rx Overruns	F	Rx	(y|	y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y|	y)	-	-	-	y	n
64octet Frames	F	Rx+Tx	(y|	y|	y|	y|	y)	n	y	n	n	n	n	n	n	-	-	-	-	n
65-127octet Frames	F	Rx+Tx	(y|	y|	y|	y|	y)	n	n	y	n	n	n	n	n	-	-	-	-	n
128-255octet Frames	F	Rx+Tx	(y|	y|	y|	y|	y)	n	n		y	n	n	n	n	-	-	-	-	n
256-511octet Frames	F	Rx+Tx	(y|	y|	y|	y|	y)	n	n	n	n	y	n	n	n	-	-	-	-	n
512-1023octet Frames	F	Rx+Tx	(y|	y|	y|	y|	y)	n	n	n	n	n	y	n	n	-	-	-	-	n
1024-UPoctet Frames	F	Rx+Tx	(y|	y|	y|	y|	y)	n	n	n	n	n	n	y	n	-	-	-	-	n
Rx Octets	O	Rx	(y|	y|	y|	y|	y)	n	(y|	y|	y|	y|	y|	y)	n	-	n	n	-	n
Net Octets	O	Rx+Tx	(y|	y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y|	y|	y)	-	-	-	-

Notes for the Rx Statistics Summary:

1. “AND” is assumed horizontally across the table between all conditions which form the statistic (marked y or n) except where (y|y), meaning “OR” is indicated. Parentheses are significant.
2. “-“ indicates conditions which are ignored in the formations of the statistic.
3. Statistics marked “Rx+Tx” are formed by summing the Rx and Tx statistics, each of which is formed independently.
4. The non-pause column refers to all MAC control frames (i.e. frames with length/type=88.08) with opcodes other than 0x0001. The pauseframe column refers to MAC frames with the opcode=0x0001.
5. The multicast, broadcast and unicast columns in the table refer to non-MACCTRL/non-pause frames (i.e. data frames).
6. “%” If either a MAC control frame or pause frame has a multicast or broadcast destination address then the appropriate statistics will be updated.
7. “y^” Frame fragments are not counted if less than 8 bytes.
8. flow coll. are half-duplex collisions forced by the MAC to achieve flow-control. A collision will be forced during the first 8 bytes so should not show in frame fragments. Some of the ‘-‘s in this column might in reality be ‘n’s.
9. The rx_overruns stat show above is for rx_mof_overruns and rx_sof_overruns added together.

Table 15-19. Tx Statistics Summary

Tx Statistic	Frame/ Oct	Tx/ Rx+ Tx	Frame Type					Frame Size (bytes)							Event									
			MAC control		Data											Collision Type								
			Pause (MAC)	Any (CPU)	Multi-cast	Broad-cast	Uni-cast	64	65-127	128-255	256-511	512-1023	1024-1535	> 1535	CRC Error	Flow	1	2-15	16	Late	No Carrier	Queued	Deferred	Under-run
Good Tx Frames	F	Tx	(y|	y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	-	-	n	n	n	-	-	n
Broadcast Tx Frames	F	Tx	n	(%|	n	y)	n	(y|	y|	y|	y|	y|	y|	y)	-	-	-	-	n	n	n	-	-	n
Multicast Tx Frames	F	Tx	(y|	%|	y)	n	n	y|	y|	y|	y|	y|	y|	y)	-	-	-	-	n	n	n	-	-	n
Pause Tx Frames	F	Tx	y	n	n	n	n	y	n	n	n	n	n	n	-	-	-	-	-	-	-	-	-	-
Collisions	F	Tx	n	(y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	(+	+	+	+	+)	n	-	-	-
Single Collision Tx Frames	F	Tx	n	(y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	y	n	n	n	n	-	-	-
Multiple Collision Tx Frames	F	Tx	n	(y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	n	y	n	n	n	-	-	-
Excessive Collisions	F	Tx	n	(y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	n	n	y	n	n	-	-	-
Late Collisions	F	Tx	n	(y|	y|	y|	y)	n	(y|	y|	y|	y|	y|	y)	-	-	-	-	-	y	-	-	-	-
Deferred Tx Frames	F	Tx	n	(y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	n	n	n	n	n	-	y	n
Carrier Sense Errors	F	Tx	(y|	y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	-	-	-	-	y	-	-	-
64octet Frames	F	Rx+ Tx	(y|	y|	y|	y|	y)	y	n	n	n	n	n	n	-	-	-	-	n	n	n	-	-	-
65-127octet Frames	F	Rx+ Tx	(y|	y|	y|	y|	y)	n	y	n	n	n	n	n	-	-	-	-	n	n	n	-	-	-
128-255octet Frames	F	Rx+ Tx	(y|	y|	y|	y|	y)	n	n	y	n	n	n	n	-	-	-	-	n	n	n	-	-	-
256-511octet Frames	F	Rx+ Tx	(y|	y|	y|	y|	y)	n	n	n	y	n	n	n	-	-	-	-	n	n	n	-	-	-
512-1023octet Frames	F	Rx+ Tx	(y|	y|	y|	y|	y)	n	n	n	n	y	n	n	-	-	-	-	n	n	n	-	-	-
1024-UPoctet Frames	F	Rx+ Tx	(y|	y|	y|	y|	y)	n	n	n	n	n	y	y	-	-	-	-	n	n	n	-	-	-
Tx Octets	O	Tx	(y|	y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	-	-	n	n	n	-	-	n
Net Octets	O	Rx+ Tx	(y|	y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	$	$	$	$	$	-	-	-

Notes for the Tx Statistics Summary:

1. "AND" is assumed horizontally across the table between all conditions which form the statistic (marked y or n) except where (y|y), meaning “OR” is indicated. Parentheses are significant.
2. “-“ indicates conditions which are ignored in the formations of the statistic.
3. Statistics marked “Rx+Tx” are formed by summing the Rx and Tx statistics, each of which is formed independently.
4. Pause (MAC) frames are issued in the MAC as perfect (no CRC error) 64 byte frames in full duplex only, so they cannot collide.
5. “%” If a CPU sourced MAC control frame has a multicast or broadcast destination address then the appropriate statistics will be updated.
6. “+” indicates collisions which are “summed” (i.e. every collision is counted in the Collisions statistic). Jam sequences used for halfduplex flow control are also counted.
7. “$” Every byte written on the wire during each retry attempt is also counted in addition to frames which experience no collisions or carrier loss.
8. The flow collision type is for half-duplex collisions forced by the MAC to achieve flow control. Some of the ‘-‘s in this column might in reality be ‘n’s. To prevent double-counting, Net Octets are unaffected by the jam sequence – the ‘received’ bytes, however, are counted. (See Table 15-18.)
9. When the transmit Tx FIFO is drained due to the MAC being disabled or link being lost, then the frames being purged will not appear in the Tx statistics.

15.3.3 Ethernet Mac Sliver (CPGMAC_SL)

The CPGMAC_SL peripheral shall be compliant to the IEEE Std 802.3 Specification. Half duplex mode is supported in 10/100 Mbps mode, but not in 1000 Mbps (gigabit) mode.

Features:

- Synchronous 10/100/1000 Mbit operation.
- G/MII Interface.
- Hardware Error handling including CRC.
- Full Duplex Gigabit operation (half duplex gigabit is not supported).
- EtherStats and 802.3Stats RMON statistics gathering support for external statistics collection module.
- Transmit CRC generation selectable on a per channel basis.
- Emulation Support.
- VLAN Aware Mode Support.
- Hardware flow control.
- Programmable Inter Packet Gap (IPG)

15.3.3.1 GMII/MII Media Independent Interface

The following sections cover operation of the Media Independent Interface in 10/100/1000 Mbps modes. An IEEE 802.3 compliant Ethernet MAC controls the interface.

15.3.3.1.1 Data Reception

15.3.3.1.1.1 Receive Control

Data received from the PHY is interpreted and output. Interpretation involves detection and removal of the preamble and start of frame delimiter, extraction of the address and frame length, data handling, error checking and reporting, cyclic redundancy checking (CRC), and statistics control signal generation.

15.3.3.1.1.2 Receive Inter-Frame Interval

The 802.3 required inter-packet gap (IPG) is 24 GMII clocks (96 bit times) for 10/100 Mbit modes, and 12 GMII clocks (96 bit times) for 1000 Mbit mode. However, the MAC can tolerate a reduced IPG (2 GMII clocks in 10/100 mode and 5 GMII clocks in 1000 mode) with a correct preamble and start frame delimiter.

This interval between frames must comprise (in the following order):

- An Inter-Packet Gap (IPG).
- A seven octet preamble (all octets 0x55).
- A one octet start frame delimiter (0x5D).

15.3.3.1.2 Data Transmission

The Gigabit Ethernet Mac Sliver (GMII) passes data to the PHY when enabled. Data is synchronized to the transmit clock rate. The smallest frame that can be sent is two bytes of data with four bytes of CRC (6 byte frame).

15.3.3.1.2.1 Transmit Control

A jam sequence is output if a collision is detected on a transmit packet. If the collision was late (after the first 64 bytes have been transmitted) the collision is ignored. If the collision is not late, the controller will back off before retrying the frame transmission. When operating in full duplex mode the carrier sense (CRS) and collision sensing modes are disabled.

15.3.3.1.2.2 CRC Insertion

The MAC generates and appends a 32-bit Ethernet CRC onto the transmitted data if the transmit packet header **pass_crc** bit is zero. For the CPMAC_SL generated CRC case, a CRC at the end of the input packet data is not allowed.

If the header word **pass_crc** bit is set, then the last four bytes of the TX data are transmitted as the frame CRC. The four CRC data bytes should be the last four bytes of the frame and should be included in the packet byte count value. The MAC performs no error checking on the outgoing CRC when the **pass_crc** bit is set.

15.3.3.1.2.3 TXER

The GMII_TXER signal is not used. If an underflow condition occurs on a transmitted frame, the frame CRC will be inverted to indicate the error to the network. Underflow is a hardware error.

15.3.3.1.2.4 Adaptive Performance Optimization (APO)

The Ethernet MAC port incorporates Adaptive Performance Optimization (APO) logic that may be enabled by setting the **tx_pace** bit in the **MACCTRL** register. Transmission pacing to enhance performance is enabled when set. Adaptive performance pacing introduces delays into the normal transmission of frames, delaying transmission attempts between stations, reducing the probability of collisions occurring during heavy traffic (as indicated by frame deferrals and collisions) thereby increasing the chance of successful transmission.

When a frame is deferred, suffers a single collision, multiple collisions or excessive collisions, the pacing counter is loaded with an initial value of 31. When a frame is transmitted successfully (without experiencing a deferral, single collision, multiple collision or excessive collision) the pacing counter is decremented by one, down to zero.

With pacing enabled, a new frame is permitted to immediately (after one IPG) attempt transmission only if the pacing counter is zero. If the pacing counter is non zero, the frame is delayed by the pacing delay, a delay of approximately four inter-packet gap delays. APO only affects the IPG preceding the first attempt at transmitting a frame. It does not affect the back-off algorithm for retransmitted frames.

15.3.3.1.2.5 Inter-Packet-Gap Enforcement

The measurement reference for the IPG of 96 bit times is changed depending on frame traffic conditions. If a frame is successfully transmitted without collision, and **GMII_CRS** is de-asserted within approximately 48 bit times of **GMII_TXEN** being de-asserted, then 96 bit times is measured from **GMII_TXEN**. If the frame suffered a collision, or if **GMII_CRS** is not de-asserted until more than approximately 48 bit times after **GMII_TXEN** s de-asserted, then 96 bit times (approximately, but not less) is measured from **GMII_CRS**.

The transmit IPG can be shortened by eight bit times when enabled and triggered. The **tx_short_gap_en** bit in the **MACCTRL** register enables the **TX_SHORT_GAP** input to determine whether the transmit IPG is shorted by eight bit times.

15.3.3.1.2.6 Back Off

The Gigabit Ethernet Mac Sliver (GMII) implements the 802.3 binary exponential back-off algorithm.

15.3.3.1.2.7 Programmable Transmit Inter-Packet Gap

The transmit inter-packet gap (IPG) is programmable through the **Tx_Gap** register. The default value is decimal 12. The transmit IPG may be increased to the maximum value of 0x1ff. Increasing the IPG is not compatible with transmit pacing. The short gap feature will override the increased gap value, so the short gap feature may not be compatible with an increased IPG.

15.3.3.1.2.8 Speed, Duplex, and Pause Frame Support Negotiation

The CPMAC_SL can operate in half duplex or full duplex in 10/100 Mbit modes, and can operate in full duplex only in 1000 Mbit mode. Pause frame support is included in 10/100/1000 Mbit modes as configured by the host.

15.3.3.2 Frame Classification

Received frames are proper (good) frames if they are between 64 and **rx_maxlen** in length (inclusive) and contain no errors (code/align/CRC).

Received frames are long frames if their frame count exceeds the value in the **rx_maxlen** register. The **rx_maxlen** register reset (default) value is 1518 (dec). Long received frames are either oversized or jabber frames. Long frames with no errors are oversized frames. Long frames with CRC, code, or alignment errors are jabber frames.

Received frames are short frames if their frame count is less than 64 bytes. Short frames that contain no errors are undersized frames. Short frames with CRC, code, or alignment errors are fragment frames.

A received long packet will always contain **rx_maxlen** number of bytes transferred to memory (if **rx_cef_en** = 1). An example with **rx_maxlen** = 1518 is below:

- If the frame length is 1518, then the packet is not a long packet and there will be 1518 bytes transferred to memory.
- If the frame length is 1519, there will be 1518 bytes transferred to memory. The last three bytes will be the first three CRC bytes.
- If the frame length is 1520, there will be 1518 bytes transferred to memory. The last two bytes will be the first two CRC bytes.
- If the frame length is 1521, there will be 1518 bytes transferred to memory. The last byte will be the first CRC byte.

If the frame length is 1522, there will be 1518 bytes transferred to memory. The last byte will be the last data byte.

15.3.4 Command IDLE

The cmd_idle bit in the MACCTRL register allows CPGMAC_SL operation to be suspended. When the idle state is commanded, the CPGMAC_SL will stop processing receive and transmit frames at the next frame boundary. Any frame currently in reception or transmission will be completed normally without suspension. Received frames that are detected after the suspend state is entered are ignored. Commanded idle is similar in operation to emulation control and clock stop.

15.3.5 RMII Interface

The CPRMII peripheral shall be compliant to the RMII specification document.

Features:

- Source Synchronous 10/100 Mbit operation.
- Full and Half Duplex support.

15.3.5.1 RMII Receive (RX)

The CPRMII receive (RX) interface converts the input data from the external RMII PHY (or switch) into the required MII (CPGMAC) signals. The carrier sense and collision signals are determined from the RMII input data stream and transmit inputs as defined in the RMII specification.

An asserted RMII_RXER on any di-bit in the received packet will cause an MII_RXER assertion to the CPGMAC during the packet. In 10Mbps mode, the error is not required to be duplicated on 10 successive clocks. Any di-bit which has an asserted RMII_RXER during any of the 10 replications of the data will cause the error to be propagated.

Any received packet that ends with an improper nibble boundary aligned RMII_CRS_DV toggle will issue an MII_RXER during the packet to the CPGMAC. Also, a change in speed or duplex mode during packet operations will cause packet corruption.

The CPRMII can accept receive packets with shortened preambles, but 0x55 followed by a 0x5d is the shortest preamble that will be recognized (1 preamble byte with the start of frame byte). At least one byte of preamble with the start of frame indicator is required to begin a packet. An asserted RMII_CRS_DV without at least a single correct preamble byte followed by the start of frame indicator will be ignored.

15.3.5.2 RMII Transmit (TX)

The CPRMII transmit (TX) interface converts the 3PSW MII input data into the RMII transmit format. The data is then output to the external RMII PHY.

The 3PSW does not source the transmit error (MII TXERR) signal. Any transmit frame from the CPGMAC with an error (ie. underrun) will be indicated as an error by an error CRC. Transmit error is assumed to be deasserted at all times and is not an input into the CPRMII module.Zeroes are output on RMII_TXD[1:0] for each clock that RMII_TXEN is deasserted.

15.3.6 RGMII Interface

The CPRGMII peripheral shall be compliant to the RGMII specification document.

Features:

- Supports 1000/100/10 Mbps Speed.
- MII mode is not supported.

If RGMII is used, and a 10Mbit operation is desired, in-band mode must be used and an ethernet PHY that supports in-band status signaling must be selected.

15.3.6.1 RGMII Receive (RX)

The CPRGMII receive (RX) interface converts the source synchronous DDR input data from the external RGMII PHY into the required G/MII (CPGMAC) signals.

15.3.6.2 In-Band Mode of Operation

The CPRGMII is operating in the in-band mode of operation when the **RGMII_RX_INBAND** input is asserted.RGMII_RX_INPUT is asserted by configuring the ext_en bit to 1 of the MACCTRL register. The link status, duplexity, and speed are determined from the RGMII input data stream as defined in the RGMII specification. The link speed is indicated as shown in the following table:

RGMII_SPEED(1:0)	Link Speed
00	10 Mbs mode
01	100 Mbs mode
10	1000 Mbs mode
11	reserved

15.3.6.3 Forced Mode of Operation

The CPRGMII is operating in the forced mode of operation when the **RGMII_RX_INBAND** input is deasserted by setting MACCTRL.EXT_EN to 0. In the forced mode of operation, the in-band data is ignored if present. In this mode, the contents of RGMII_CTL are meaningless. Link status, duplexity, and speed status should be determined from the external ethernet PHY via MDIO transactions.

15.3.6.4 RGMII Transmit (TX)

The CPRGMII transmit (TX) interface converts the CPGMAC G/MII input data into the DDR RGMII format. The DDR data is then output to the external PHY.

The CPGMAC does not source the transmit error (MTXERR) signal. Any transmit frame from the CPGMAC with an error (that is, underrun) will be indicated as an error by an error CRC. Transmit error is assumed to be deasserted at all times and is not an input into the CPRGMII module.

The RGMII0/1_ID_MODE bit value in the GMII_SEL register should only be set to 1 for 'no internal delay'. The device does not support internal delay mode for RGMII.

15.3.7 Common Platform Time Sync (CPTS)

The CPTS module is used to facilitate host control of time sync operations. It enables compliance with the IEEE 1588-2008(v2) standard for a precision clock synchronization protocol.

15.3.7.1 Architecture

Figure 15-10. CPTS Block Diagram

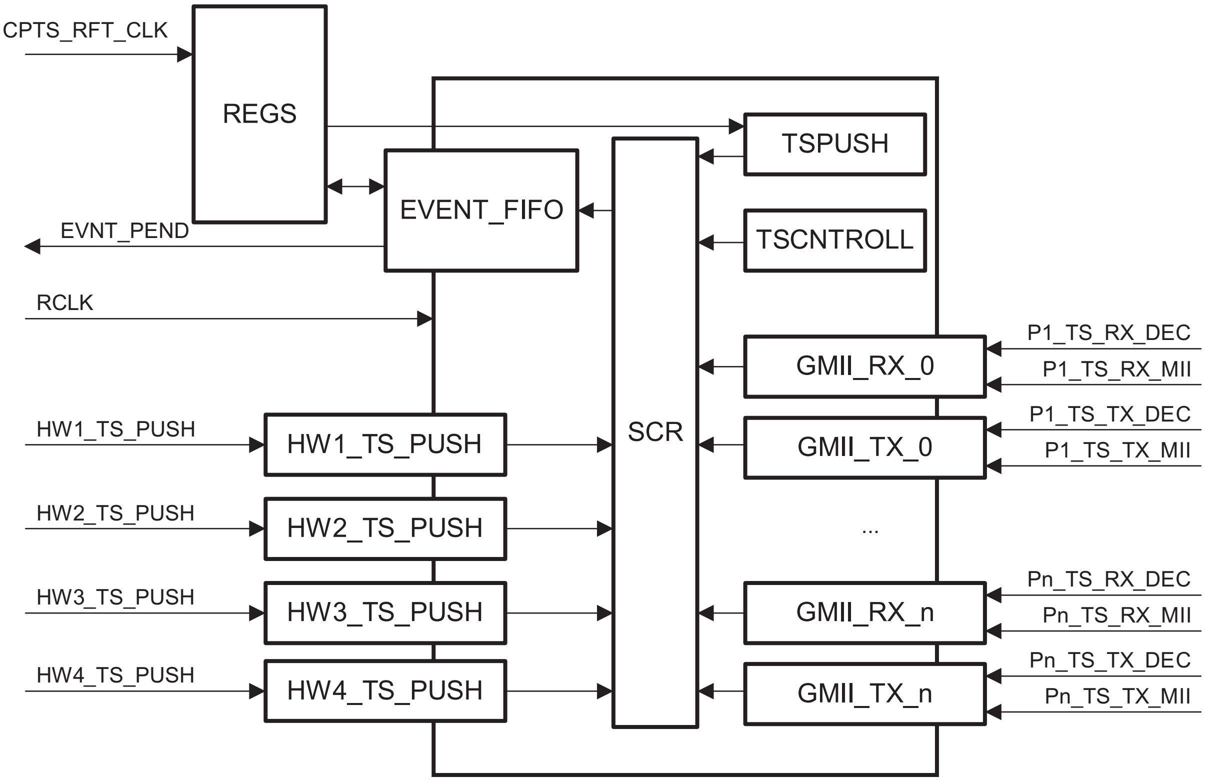


Figure 15-10 shows the architecture of the CPTS module inside the 3PSW Ethernet Subsystem. Time stamp values for every packet transmitted or received on either port of the 3PSW are recorded. At the same time, each packet is decoded to determine if it is a valid time sync event. If so, an event is loaded into the Event FIFO for processing containing the recorded time stamp value when the packet was transmitted or received.

In addition, both hardware (HWx_TS_PUSH) and software (TS_PUSH) can be used to read the current time stamp value though the Event FIFO

The reference clock used for the time stamp (RCLK) is sourced from one of the two sources, as shown in Figure 15-10. The source can be selected by configuring the CM_CPTS_RFT_CLKSEL register in the Control Module. For more details, see Chapter 7, *Control Module*.

15.3.7.2 Time Sync Overview

The CPTS module is used to facilitate host control of time sync operations. The CPTS collects time sync events and then presents them to the host for processing. There are five types of time sync events (ethernet receive event, ethernet transmit event, time stamp push event, time stamp rollover event, and time stamp half-rollover event). Each ethernet port can cause transmit and receive events. The time stamp push is initiated by software.

15.3.7.2.1 Time Sync Initialization

The CPTS module should be configured as shown:

- Complete the reset sequence (VBUSP_RST_N) to reset the module.
- Write the rftclk_sel[4:0] value in the RFTCLK_Sel register with the desired reference clock multiplexor value. This value is allowed to be written only when the cpts_en bit is cleared to zero.
- Write a one to the cpts_en bit in the TS_Ctrl register. The RCLK domain is in reset while this bit is low.
- Enable the interrupt by writing a one to the ts_pend_en bit in the TS_Int_En register (if using interrupts and not polling).

15.3.7.2.2 Time Stamp Value

The time stamp value is a 32-bit value that increments on each RCLK rising edge when CPTS_EN is set to one. When CPTS_EN is cleared to zero the time stamp value is reset to zero. If more than 32-bits of time stamp are required by the application, the host software must maintain the necessary number of upper bits. The upper time stamp value should be incremented by the host when the rollover event is detected.

For test purposes, the time stamp can be written via the time stamp load function (CPTS_LOAD_VAL and CPTS_LOAD_EN registers).

15.3.7.2.3 Event FIFO

All time sync events are pushed onto the Event FIFO. There are 16 locations in the event FIFO with no overrun indication supported. Software must service the event FIFO in a timely manner to prevent FIFO overrun.

15.3.7.2.4 Time Sync Events

Time Sync events are 64-bit values that are pushed onto the event FIFO and read in two 32-bit reads. CPTS_EVT_LOW and CPTS_EVT_HIGH are defined in and , respectively.

There are six types of sync events

- Time stamp push event
- Hardware time stamp push event
- Time stamp counter rollover event
- Time stamp counter half-rollover event
- Ethernet receive event
- Ethernet transmit event

15.3.7.2.4.1 Time Stamp Push Event

Software can obtain the current time stamp value (at the time of the write) by initiating a time stamp push event. The push event is initiated by setting the TS_PUSH bit of the CPTS_TS_PUSH register. The time stamp value is returned in the event, along with a time stamp push event code. Software should not push a second time stamp event on to the FIFO until the first time stamp value has been read from the event FIFO.

15.3.7.2.4.2 Time Stamp Counter Rollover Event

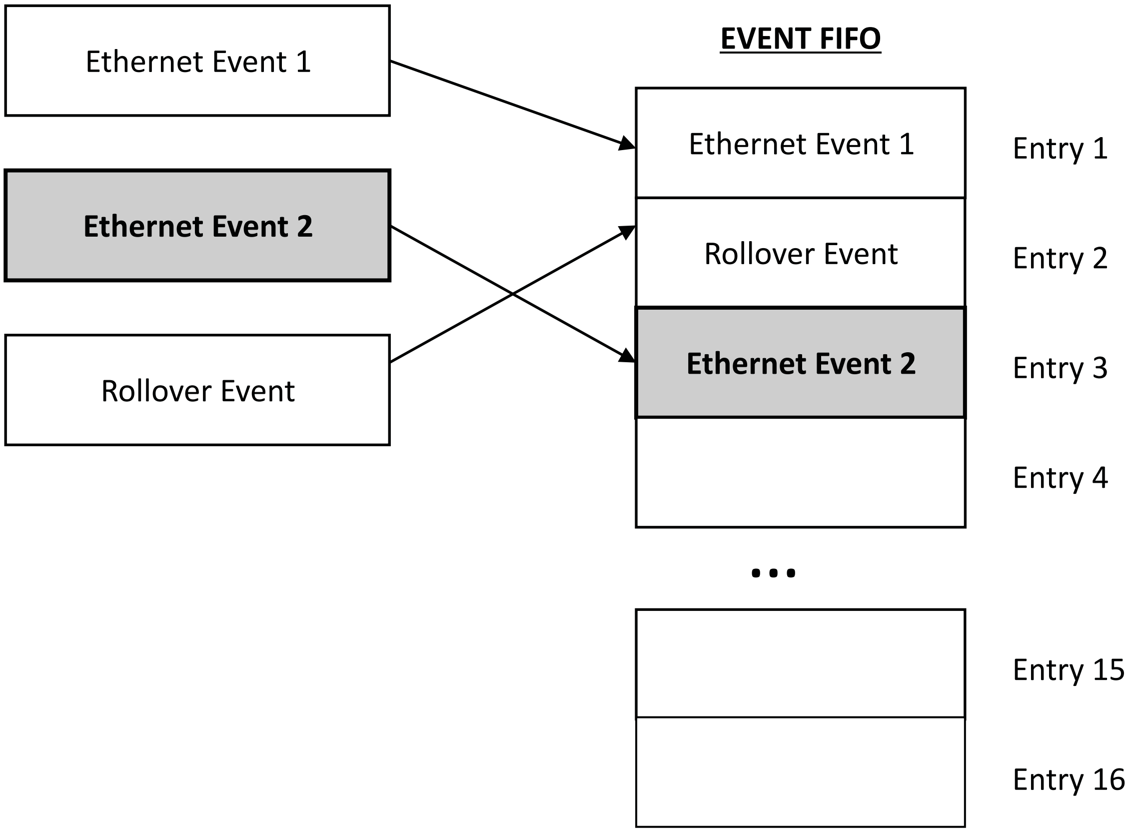
The CPTS module contains a 32-bit time stamp value. The counter upper bits are maintained by host software. The rollover event indicates to software that the time stamp counter has rolled over from 0xFFFF_FFFF to 0x0000_0000, and the software maintained upper count value should be incremented.

15.3.7.2.4.3 Time Stamp Counter Half-Rollover Event

The CPTS includes a time stamp counter half-rollover event. The half-rollover event indicates to software that the time stamp value has incremented from 0x7FFF_FFFF to 0x8000_0000. The half-rollover event is included to enable software to correct a misaligned event condition.The half-rollover event is included to enable software to determine the correct time for each event that contains a valid time stamp value – such as an Ethernet event. If an Ethernet event occurs around a counter rollover (full rollover), the rollover event could possibly be loaded into the event FIFO before the Ethernet event, even though the Ethernet event time was actually taken before the rollover. Figure 3 below shows a misalignment condition.

This misaligned event condition arises because an ethernet event time stamp occurs at the beginning of a packet and time passes before the packet is determined to be a valid synchronization packet. The misaligned event condition occurs if the rollover occurs in the middle, after the packet time stamp has been taken, but before the packet has been determined to be a valid time sync packet.

Figure 15-11. Event FIFO Misalignment Condition



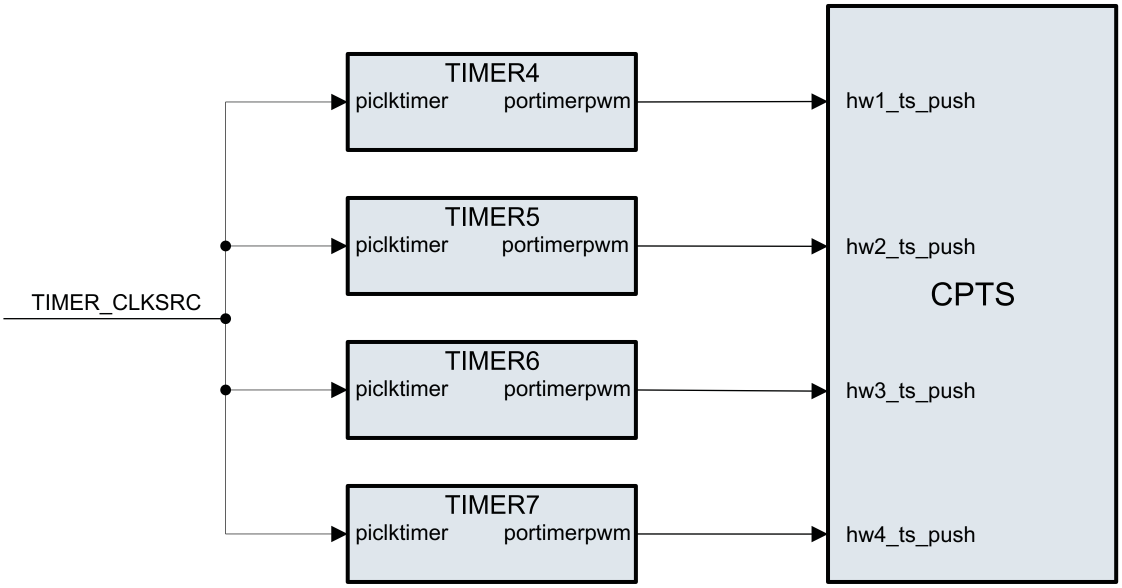
Host software must detect and correct for misaligned event conditions. For every event after a rollover and before a half-rollover, software must examine the time stamp most significant bit. If bit 31 of the time stamp value is low (0x0000_0000 through 0x7FFF_FFFF), then the event time stamp was taken after the rollover and no correction is required.

If the value is high (0x8000_0000 through 0xFFFF_FFFF), the time stamp value was taken before the rollover and a misalignment is detected. The misaligned case indicates to software that it must subtract one from the upper count value stored in software to calculate the correct time for the misaligned event.The misaligned event occurs only on the rollover boundary and not on the half-rollover boundary. Software only needs to check for misalignment from a rollover event to a half-rollover event.

15.3.7.2.4.4 Hardware Time Stamp Push Event

There are four hardware time stamp inputs (HW1/4_TS_PUSH) that can cause hardware time stamp push events to be loaded into the Event FIFO. Each hardware time stamp input is internally connected to the PORTIMERPWM output of each timer as shown in Figure 4.

Figure 15-12. HW1/4_TSP_PUSH Connection



The event is loaded into the event FIFO on the rising edge of the timer, and the PORT_NUMBER field in the EVT_HIGH register indicates the hardware time stamp input that caused the event.

Each hardware time stamp input must be asserted for at least 10 periods of the selected RCLK clock. Each input can be enabled or disabled by setting the respective bits in the ***CONTROL*** register.

Hardware time stamps are intended to be an extremely low frequency signals, such that the event FIFO does not overrun. Software must keep up with the event FIFO and ensure that there is no overrun, or events will be lost.

15.3.7.2.4.5 Ethernet Port Events

15.3.7.2.4.5.1 Ethernet Port Receive Event

Each ethernet port can generate a receive ethernet event. Receive ethernet events are generated for valid received time sync packets. There are two CPTS interfaces for each ethernet receive port. The first is the Px_TS_RX_MII interface and the second is the Px_TS_RX_DEC interface. Information from these interfaces is used to generates an ethernet receive event for each ethernet time sync packet received on the associated port.

The Px_TS_RX_MII interface issues a record signal (pX_ts_rx_mii_rec) along with a handle (pX_ts_rx_mii_hndl) for each packet (every packet) that is received on the associated ethernet port. The record signal is a single clock pulse indicating that a receive packet has been detected at the associated port MII interface. The handle value is incremented with each packet and rolls over to zero after 15.

There are 16 possible handle values so there can be a maximum of 16 packets “in flight” from the TS_RX_MII to the TS_RX_DEC block at any given time. A handle value is reused (not incremented) for any received packet that is shorter than about 31 octets (including preamble). Handle reuse on short packets prevents any possible overrun condition (more than 16 “in flight” packets) if multiple fragments are consecutively received.

Valid receive ethernet time sync events are signaled to the CPTS via the Px_TS_RX_DEC interface. When the pX_ts_rx_dec_evnt is asserted, a valid event is detected and will be loaded into the event FIFO. Only valid receive time sync packets are indicated on the Px_TS_RX_DEC interface. The pX_ts_rx_dec_hndl, pX_ts_rx_dec_msg_type, and pX_ts_rx_dec_seq_id signals are registered on an asserted pX_ts_rx_dec_evnt. When a Tx_TS_RX_DEC event is asserted, the handle value is used to retrieve the time stamp that was loaded with the same handle value from the Px_TS_RX_MII interface.

15.3.7.2.4.5.2 Ethernet Port Transmit Event

Each ethernet port can generate a transmit ethernet event. Transmit ethernet events are generated for valid transmitted time sync packets. There are two CPTS interfaces for each ethernet transmit port. The first is the Px_TS_TX_DEC interface and the second is the Px_TS_TX_MII interface. Information from these interfaces is used to generates an ethernet transmit event for each ethernet time sync packet transmitted on the associated port.

Valid ethernet transmit time sync events are signaled to the CPTS via the Px_TS_TX_DEC interface. When the pX_ts_tx_dec_evnt signal is asserted, a valid time sync event has been detected and will be loaded into the event FIFO. Only valid transmit time sync packets are indicated on the Px_TS_RX_DEC interface. The pX_ts_tx_dec_hndl, pX_ts_tx_dec_msg_type, pX_ts_tx_dec_seq_id signals are registered on an asserted pX_ts_tx_dec_evnt.

The time stamp for the event will be generated and signaled from the Px_TS_TX_MII interface when the packet is actually transmitted. The event will be loaded into the event FIFO when the time stamp is recorded as controlled by the Px_TS_TX_MII interface. The handle value is incremented with each time sync event packet and rolls over to zero after 7. There are 8 possible handle values so there can be a maximum of 8 time sync event packets “in flight” from the TS_TX_DEC to the TS_TX_MII block at any given time. The handle value increments only on time sync event packets.

The Px_TS_TX_MII interface issues a single clock record signal (pX_ts_tx_mii_rec) at the beginning of each transmitted packet. If the packet is a time sync event packet then a single clock event signal (pX_ts_tx_mii_evnt) along with a handle (pX_ts_rx_mii_hndl) will be issued before the next record signal for the next transmitted packet. The event signal will not be issued for packets that were not indicated as valid time sync event packets on the Px_TS_TX_DEC interface. If consecutive record indications occur without an interleaving event indication, then the packet associated with the first record was not a time sync event packet. The record signal is a single clock pulse indicating that a transmit packet egress has been detected at the associated port MII interface.

Table 15-20. Values of messageType field

Message Type	Value (hex)
Sync	0
Delay_Req	1
Pdelay_Req	2
Pdelay_Resp	3
Reserved	4-7
Follow_Up	8
Delay_Resp	9
Pdelay_Resp_Follow_Up	A
Announce	B
Signaling	C
Management	D
Reserved	E-F

15.3.7.3 Interrupt Handling

When an event is push onto the Event FIFO, an interrupt can be generated to indicate to software that a time sync event occurred. The following steps should be taken to process time sync events using interrupts:

- Enable the TS_PEND interrupt by setting the TS_PEND_EN bit of the CPTS_INT_EN register.
- Upon interrupt, read the CPTS_EVT_LOW and CPTS_EVT_HIGH register values.
- Set the EVT_POP field (bit zero) of the CPTS_EVT_POP register to pop the previously read value off of the event FIFO.
- Process the interrupt as required by the application software

Software has the option of processing more than a single event from the event FIFO in the interrupt service routine in the following way:

1. Enable the TS_PEND interrupt by setting the TS_PEND_EN bit of the CPTS_INT_EN register.
2. Upon interrupt enter the CPTS service routine.
3. Read the CPTS_EVT_LOW and CPTS_EVT_HIGH register values.
4. Set the EVT_POP bit of the CPTS_EVT_POP register to pop the previously read value off of the event FIFO.
5. Wait for an amount of time greater than eight CPTS_RFT_CLK periods
6. Read the ts_pend_raw bit in the CPTS_INTSTAT_RAW register to determine if another valid event is in the event FIFO. If it is asserted then goto step 3. Otherwise goto step 7.
7. Process the interrupt(s) as required by the application software

Software also has the option of disabling the interrupt and polling the ts_pend_raw bit of the CPTS_INTSTAT_RAW register to determine if a valid event is on the event FIFO.

15.3.8 MDIO

The MII Management I/F module implements the 802.3 serial management interface to interrogate and control two Ethernet PHYs simultaneously using a shared two-wire bus. Two user access registers to control and monitor up to two PHYs simultaneously.

15.3.8.1 MII Management Interface Frame Formats

The following tables show the read and write format of the 32-bit MII Management interface frames, respectively.

Table 15-21. MDIO Read Frame Format

Preamble	Start Delimiter	Operation Code	PHY Address	Register Address	Turnaround	Data
0xFFFF FFFF	01	10	AAAAA	RRRRR	Z0	DDDD.DDDD. DDDD.DDDD

Table 15-22. MDIO Write Frame Format

Preamble	Start Delimiter	Operation Code	PHY Address	Register Address	Turnaround	Data
0xFFFF FFFF	01	01	AAAAA	RRRRR	10	DDDD.DDDD. DDDD.DDDD

The default or idle state of the two wire serial interface is a logic one. All tri-state drivers should be disabled and the PHY's pull-up resistor will pull the **MDIO_DATA** line to a logic one. Prior to initiating any other transaction, the station management entity shall send a preamble sequence of 32 contiguous logic one bits on the **MDIO_DATA** line with 32 corresponding cycles on **MDIO_CLK** to provide the PHY with a pattern that it can use to establish synchronization. A PHY shall observe a sequence of 32 contiguous logic one bits on **MDIO_DATA** with 32 corresponding **MDIO_CLK** cycles before it responds to any other transaction.

Preamble

The start of a frame is indicated by a preamble, which consists of a sequence of 32 contiguous bits all of which are a “1”. This sequence provides the PHY a pattern to use to establish synchronization.

Start Delimiter

The preamble is followed by the start delimiter which is indicated by a “01” pattern. The pattern assures transitions from the default logic one state to zero and back to one.

Operation Code

The operation code for a read is “10”, while the operation code for a write is a “01”.

PHY Address

The PHY address is 5 bits allowing 32 unique values. The first bit transmitted is the MSB of the PHY address.

Register Address

The Register address is 5 bits allowing 32 registers to be addressed within each PHY. Refer to the 10/100 PHY address map for addresses of individual registers.

Turnaround

An idle bit time during which no device actively drives the MDIO_DATA signal shall be inserted between the register address field and the data field of a read frame in order to avoid contention. During a read frame, the PHY shall drive a zero bit onto MDIO_DATA for the first bit time following the idle bit and preceding the Data field. During a write frame, this field shall consist of a one bit followed by a zero bit.

Data

The Data field is 16 bits. The first bit transmitted and received is the MSB of the data word.

15.3.8.2 Functional Description

The MII Management I/F will remain idle until enabled by setting the enable bit in the MDIOCTRL register. The MII Management I/F will then continuously poll the link status from within the Generic Status Register of all possible 32 PHY addresses in turn recording the results in the MDIO link register.

The linksel bit in the MDIOUserPhySel register determines the status input that is used. A change in the link status of the two PHYs being monitored will set the appropriate bit in the MDIOLinkIntRaw register and the MDIOLinkIntMasked register, if enabled by the linkint_enable bit in the MDIOUserPhySel register.

The MDIO Alive register is updated by the MII Management I/F module if the PHY acknowledged the read of the generic status register. In addition, any PHY register read transactions initiated by the host also cause the MDIOAlive register to be updated.

At any time, the host can define a transaction for the MII Management interface module to undertake using the data, phyadr, regadr, and write fields in a MDIOUserAccess register. When the host sets the go bit in this register, the MII Management interface module will begin the transaction without any further intervention from the host. Upon completion, the MII Management interface will clear the go bit and set the userintraw bit in the MDIOUserIntRaw register corresponding to the MDIOUserAccess register being used.

The corresponding bit in the MDIOUserIntMasked register may also be set depending on the mask setting in the MDIOUserIntMaskSet and MDIOUserIntMaskClr registers. A round-robin arbitration scheme is used to schedule transactions which may queued by the host in different MDIOUserAccess registers. The host should check the status of the go bit in the MDIOUserAccess register before initiating a new transaction to ensure that the previous transaction has completed. The host can use the ack bit in the MDIOUserAccess register to determine the status of a read transaction.

It is necessary for software to use the MII Management interface module to setup the auto-negotiation parameters of each PHY attached to a MAC port, retrieve the negotiation results, and setup the MACCTRL register in the corresponding MAC.

15.4 Software Operation

15.4.1 Transmit Operation

After reset the host must write zeroes to all Tx DMA State head descriptor pointers. The Tx port may then be enabled. To initiate packet transmission the host constructs transmit queues in memory (one or more packets for transmission) and then writes the appropriate Tx DMA state head descriptor pointers. For each buffer added to a transmit queue, the host must initialize the Tx buffer descriptor values as follows:

1. Write the Next Descriptor Pointer with the 32-bit aligned address of the next descriptor in the queue (zero if last descriptor).
2. Write the Buffer Pointer with the byte aligned address of the buffer data.
3. Write the Buffer Length with the number of bytes in the buffer.
4. Write the Buffer Offset with the number of bytes in the offset to the data (nonzero with SOP only).
5. Set the SOP, EOP, and Ownership bits as appropriate.
6. Clear the End Of Queue bit.

The port begins Tx packet transmission on a given channel when the host writes the channel's Tx queue head descriptor pointer with the address of the first buffer descriptor in the queue (nonzero value). Each channel may have one or more queues, so each channel may have one or more head descriptor pointers. The first buffer descriptor for each Tx packet must have the Start of Packet (SOP) bit and the Ownership bit set to one by the host. The last buffer descriptor for each Tx packet must have the End of Packet (EOP) bit set to one by the host.

The port will transmit packets until all queued packets have been transmitted and the queue(s) are empty. When each packet transmission is complete, the port will clear the Ownership bit in the packet's SOP buffer descriptor and issue an interrupt to the host by writing the packet's last buffer descriptor address to the queue's Tx DMA State Completion Pointer. The interrupt is generated by the write, regardless of the value written.

When the last packet in a queue has been transmitted, the port sets the End Of Queue bit in the EOP buffer descriptor, clears the Ownership bit in the SOP Descriptor, zeroes the appropriate DMA state head descriptor pointer, and then issues a Tx interrupt to the host by writing to the queue's associated Tx completion pointer (address of the last buffer descriptor processed by the port). The port issues a maskable level interrupt (which may then be routed through external interrupt control logic to the host).

On interrupt from the port, the host processes the buffer queue, detecting transmitted packets by the status of the Ownership bit in the SOP buffer descriptor. If the Ownership bit is cleared to zero, then the packet has been transmitted and the host may reclaim the buffers associated with the packet. The host continues queue processing until the end of the queue or until a SOP buffer descriptor is read that contains a set Ownership bit indicating that the packet transmission is not complete.

The host determines that all packets in the queue have been transmitted when the last packet in the queue has a cleared Ownership bit in the SOP buffer descriptor, the End of Queue bit is set in the last packet EOP buffer descriptor, and the Next Descriptor Pointer of the last packet EOP buffer descriptor is zero. The host acknowledges an interrupt by writing the address of the last buffer descriptor to the queue's associated Tx Completion Pointer in the Tx DMA State.

If the host written buffer address value is different from the buffer address written by the port, then the level interrupt remains asserted. If the host written buffer address value is equal to the port written value, then the level interrupt is deasserted. The port write to the completion pointer actually stores the value in the state register (ram). The host written value is actually not written to the register location. The host written value is compared to the register contents (which was written by the port) and if the two values are equal, the interrupt is removed, otherwise the interrupt remains asserted. The host may process multiple packets previous to acknowledging an interrupt, or the host may acknowledge interrupts for every packet.

A misqueued packet condition may occur when the host adds a packet to a queue for transmission as the port finishes transmitting the previous last packet in the queue. The misqueued packet is detected by the host when queue processing detects a cleared Ownership bit in the SOP buffer descriptor, a set End of Queue bit in the EOP buffer descriptor, and a nonzero Next Descriptor Pointer in the EOP buffer descriptor.

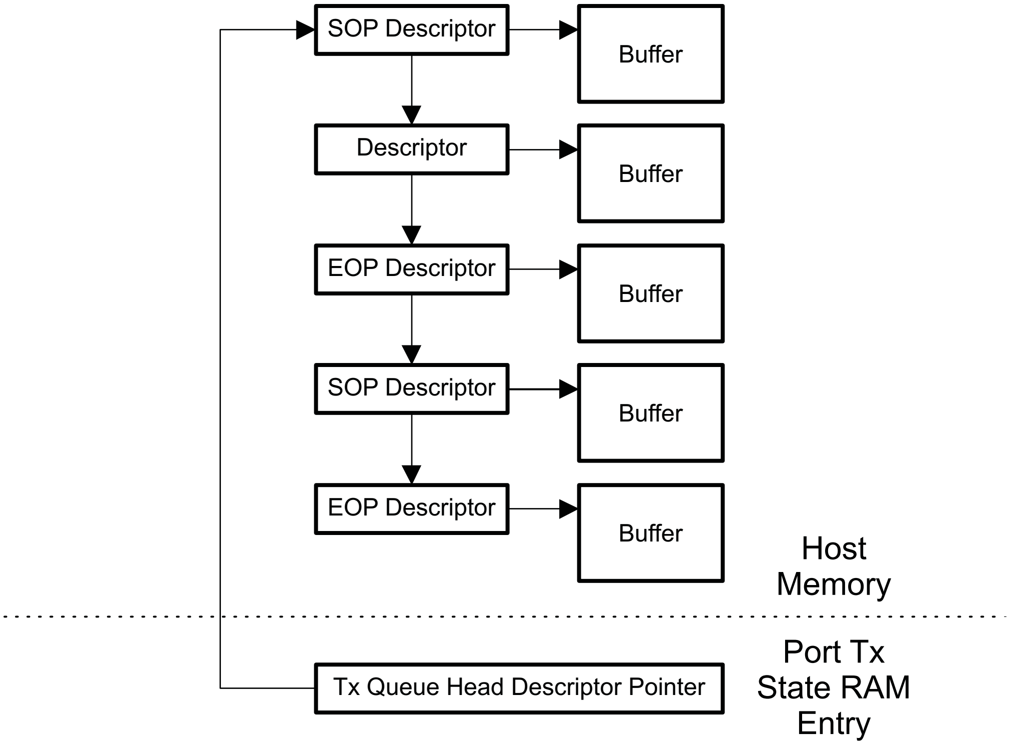
A misqueued packet means that the port read the last EOP buffer descriptor before the host added the new last packet to the queue, so the port determined queue empty just before the last packet was added. The host corrects the misqueued packet condition by initiating a new packet transfer for the misqueued packet by writing the misqueued packet's SOP buffer descriptor address to the appropriate DMA State Tx Queue head Descriptor Pointer.

The host may add packets to the tail end of an active Tx queue at any time by writing the Next Descriptor Pointer to the current last descriptor in the queue. If a Tx queue is empty (inactive), the host may initiate packet transmission at any time by writing the appropriate Tx DMA State head descriptor pointer.

The host software should always check for and reinitiate transmission for misqueued packets during queue processing on interrupt from the port. In order to preclude software underrun, the host should avoid adding buffers to an active queue for any Tx packet that is not complete and ready for transmission.

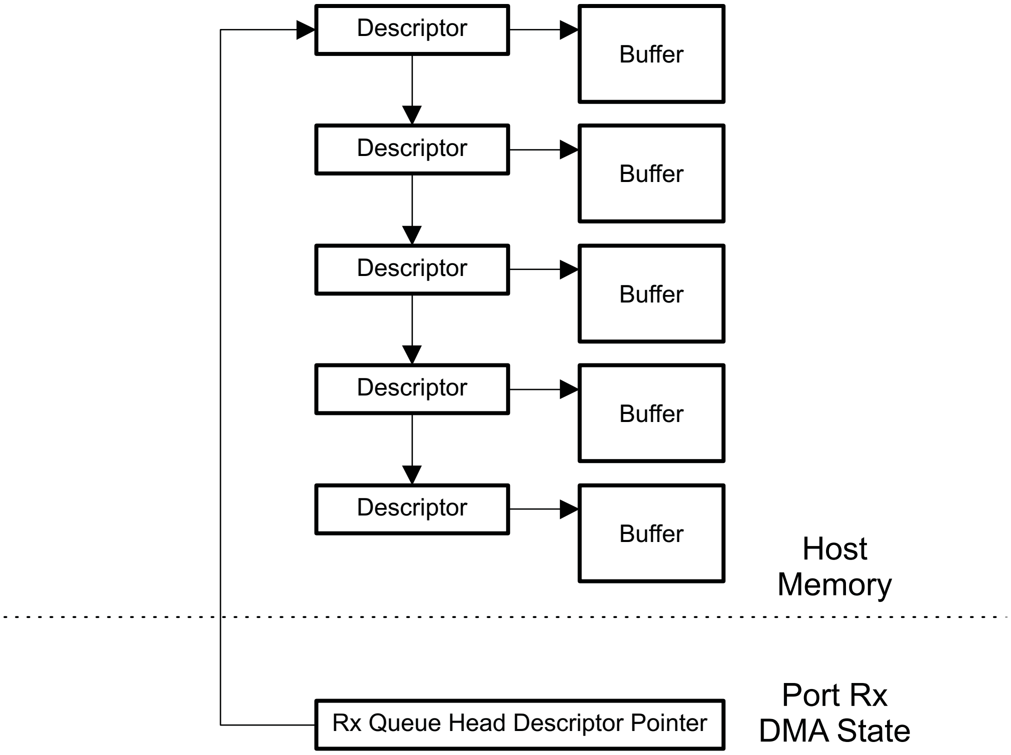
The port determines that a packet is the last packet in the queue by detecting the End of Packet bit set with a zero Next Descriptor Pointer in the packet buffer descriptor. If the End of Packet bit is set and the Next Descriptor Pointer is nonzero, then the queue still contains one or more packets to be transmitted. If the EOP bit is set with a zero Next Descriptor Pointer, then the port will set the EOQ bit in the packet's EOP buffer descriptor and then zero the appropriate head descriptor pointer previous to interrupting the port (by writing the completion pointer) when the packet transmission is complete.

Figure 15-13. Port TX State RAM Entry



15.4.2 Receive Operation

Figure 15-14. Port RX DMA State



After reset the host must write zeroes to all Rx DMA State head descriptor pointers. The Rx port may then be enabled. To initiate packet reception, the host constructs receive queues in memory and then writes the appropriate Rx DMA state head descriptor pointer. For each Rx buffer descriptor added to the queue, the host must initialize the Rx buffer descriptor values as follows:

- Write the Next Descriptor Pointer with the 32-bit aligned address of the next descriptor in the queue (zero if last descriptor)
- Write the Buffer Pointer with the byte aligned address of the buffer data
- Clear the Offset field
- Write the Buffer Length with the number of bytes in the buffer
- Clear the SOP, EOP, and EOQ bits
- Set the Ownership bit

The host enables packet reception on a given channel by writing the address of the first buffer descriptor in the queue (nonzero value) to the channel's head descriptor pointer in the channel's Rx DMA state. When packet reception begins on a given channel, the port fills each Rx buffer with data in order starting with the first buffer and proceeding through the Rx queue. If the Buffer Offset in the Rx DMA State is nonzero, then the port will begin writing data after the offset number of bytes in the SOP buffer. The port performs the following operations at the end of each packet reception:

- Overwrite the buffer length in the packet's EOP buffer descriptor with the number of bytes actually received in the packet's last buffer. The host initialized value is the buffer size. The overwritten value will be less than or equal to the host initialized value.
- Set the EOP bit in the packet's EOP buffer descriptor.
- Set the EOQ bit in the packet's EOP buffer descriptor if the current packet is the last packet in the queue.
- Overwrite the packet's SOP buffer descriptor Buffer Offset with the Rx DMA state value (the host initialized the buffer descriptor Buffer Offset value to zero). All non SOP buffer descriptors must have a zero Buffer Offset initialized by the host.

- Overwrite the packet's SOP buffer descriptor buffer length with the number of valid data bytes in the buffer. If the buffer is filled up, the buffer length will be the buffer size minus buffer offset.
- Set the SOP bit in the packet's SOP buffer descriptor.
- Write the SOP buffer descriptor Packet Length field.
- Clear the Ownership bit in the packet's SOP buffer descriptor.
- Issue an Rx host interrupt by writing the address of the packet's last buffer descriptor to the queue's Rx DMA State Completion Pointer. The interrupt is generated by the write to the Rx DMA State Completion Pointer address location, regardless of the value written.

On interrupt the host processes the Rx buffer queue detecting received packets by the status of the Ownership bit in each packet's SOP buffer descriptor. If the Ownership bit is cleared then the packet has been completely received and is available to be processed by the host.

The host may continue Rx queue processing until the end of the queue or until a buffer descriptor is read that contains a set Ownership bit indicating that the next packet's reception is not complete. The host determines that the Rx queue is empty when the last packet in the queue has a cleared Ownership bit in the SOP buffer descriptor, a set End of Queue bit in the EOP buffer descriptor, and the Next Descriptor Pointer in the EOP buffer descriptor is zero.

A misqueued buffer may occur when the host adds buffers to a queue as the port finishes the reception of the previous last packet in the queue. The misqueued buffer is detected by the host when queue processing detects a cleared Ownership bit in the SOP buffer descriptor, a set End of Queue bit in the EOP buffer descriptor, and a nonzero Next Descriptor Pointer in the EOP buffer descriptor.

A misqueued buffer means that the port read the last EOP buffer descriptor before the host added buffer descriptor(s) to the queue, so the port determined queue empty just before the host added more buffer descriptor(s). In the transmit case, the packet transmission is delayed by the time required for the host to determine the condition and reinitiate the transaction, but the packet is not actually lost. In the receive case, receive overrun condition may occur in the misqueued buffer case.

If a new packet reception is begun during the time that the port has determined the end of queue condition, then the received packet will overrun (start of packet overrun). If the misqueued buffer occurs during the middle of a packet reception then middle of packet overrun may occur. If the misqueued buffer occurs after the last packet has completed, and is corrected before the next packet reception begins, then overrun will not occur. The host acts on the misqueued buffer condition by writing the added buffer descriptor address to the appropriate Rx DMA State Head Descriptor Pointer.

15.4.3 Initializing the MDIO Module

The following steps are performed by the application software or device driver to initialize the MDIO device:

1. Configure the PREAMBLE and CLKDIV bits in the MDIO control register (MDIOCTRL).
2. Enable the MDIO module by setting the EN bit in MDIOCTRL.
3. The MDIO PHY alive status register (MDIOALIVE) can be read in polling fashion until a PHY connected to the system responded, and the MDIO PHY link status register (MDIOLINK) can determine whether this PHY already has a link.
4. Setup the appropriate PHY addresses in the MDIO user PHY select register (MDIOUSERPHYSEL*n*), and set the LINKINTENB bit to enable a link change event interrupt if desirable.

- If an interrupt on general MDIO register access is desired, set the corresponding bit in the MDIO user command complete interrupt mask set register (MDIOUSERINTMASKSET) to use the MDIO user access register (MDIOUSERACCESS*n*). Since only one PHY is used in this device, the application software can use one MDIOUSERACCESS*n* to trigger a completion interrupt; the other MDIOUSERACCESS*n* is not setup.

15.4.4 Writing Data to a PHY Register

The MDIO module includes a user access register (MDIOUSERACCESS*n*) to directly access a specified PHY device.To write a PHY register, perform the following:

1. Ensure that the GO bit in the MDIO user access register (MDIOUSERACCESS*n*) is cleared.

2. Write to the GO, WRITE, REGADR, PHYADR, and DATA bits in MDIOUSERACCESS*n* corresponding to the PHY and PHY register you want to write.
3. The write operation to the PHY is scheduled and completed by the MDIO module. Completion of the write operation can be determined by polling the GO bit in MDIOUSERACCESS*n* for a 0.
4. Completion of the operation sets the corresponding USERINTRAW bit (0 or 1) in the MDIO user command complete interrupt register (MDIOUSERINTRAW) corresponding to USERACCESS*n* used. If interrupts have been enabled on this bit using the MDIO user command complete interrupt mask set register (MDIOUSERINTMASKSET), then the bit is also set in the MDIO user command complete interrupt register (MDIOUSERINTMASKED) and an interrupt is triggered on the CPU.

15.4.5 Reading Data from a PHY Register

The MDIO module includes a user access register (MDIOUSERACCESS*n*) to directly access a specified PHY device. To read a PHY register, perform the following:

1. Ensure that the GO bit in the MDIO user access register (MDIOUSERACCESS*n*) is cleared.
2. Write to the GO, REGADR, and PHYADR bits in MDIOUSERACCESS*n* corresponding to the PHY and PHY register you want to read.
3. The read data value is available in the DATA bits in MDIOUSERACCESS*n* after the module completes the read operation on the serial bus. Completion of the read operation can be determined by polling the GO and ACK bits in MDIOUSERACCESS*n*. After the GO bit has cleared, the ACK bit is set on a successful read.
4. Completion of the operation sets the corresponding USERINTRAW bit (0 or 1) in the MDIO user command complete interrupt register (MDIOUSERINTRAW) corresponding to USERACCESS*n* used. If interrupts have been enabled on this bit using the MDIO user command complete interrupt mask set register (MDIOUSERINTMASKSET), then the bit is also set in the MDIO user command complete interrupt register (MDIOUSERINTMASKED) and an interrupt is triggered on the CPU.

15.4.6 Initialization and Configuration of CPSW

To configure the 3PSW Ethernet Subsystem for operation the host must perform the following:

- Select the Interface (GMII/RGMII/MII) Mode in the Control Module.
- Configure pads (PIN muxing) as per the Interface Selected using the appropriate pin muxing conf_xxx registers in the Control Module.
- Enable the 3PSW Ethernet Subsystem Clocks. See Section 15.2.2 and to enable the appropriate clocks.
- Configure the PRCM Registers CM_PER_CPSW_CLKSTCTRL and CM_PER_CPSW_CLKSTCTRL to enable power and clocks to the 3PSW Ethernet Subsystem. See Section 6.13.6 for register details.
- Apply soft reset to 3PSW Subsytem, CPSW_3G, CPGMAC_SL1/2, and CPDMA (see the soft reset registers in the following sections).
- Initialize the HDPs (Header Description Pointer) and CPs (Completion Pointer) to NULL
- Configure the Interrupts (see Chapter 8).
- Configure the CPSW_3G Control register.
- Configure the Statistics Port Enable register
- Configure the ALE. (See Section 15.3.2.7.)
- Configure the MDIO.
- Configure the CPDMA receive DMA controller.
- Configure the CPDMA transmit DMA controller.
- Configure the CPPI Tx and Rx Descriptors.
- Configure CPGMAC_SL1 and CPGMAC_SL2 as per the desired mode of operations.
- Start up RX and TX DMA (write to HDP of Rx and Tx).
- Wait for the completion of the transfer (HDP cleared to zero).

15.5 Ethernet Subsystem Registers

15.5.1 CPSW_ALE Registers

Table 15-23 lists the memory-mapped registers for the CPSW_ALE. All register offset addresses not listed in Table 15-23 should be considered as reserved locations and the register contents should not be modified.

Table 15-23. CPSW_ALE Registers

Offset	Acronym	Register Name	Section
0h	CPSW_ALE_IDVER	ADDRESS LOOKUP ENGINE ID/VERSION REGISTER	Section 15.5.1.1
8h	CPSW_ALE_CTRL	ADDRESS LOOKUP ENGINE CONTROL REGISTER	Section 15.5.1.2
10h	CPSW_ALE_PRESCALE	ADDRESS LOOKUP ENGINE PRESCALE REGISTER	Section 15.5.1.3
18h	CPSW_ALE_UNKNOWN_VLAN	ADDRESS LOOKUP ENGINE UNKNOWN VLAN REGISTER	Section 15.5.1.4
20h	CPSW_ALE_TBLCTL	ADDRESS LOOKUP ENGINE TABLE CONTROL	Section 15.5.1.5
34h	CPSW_ALE_TBLW2	ADDRESS LOOKUP ENGINE TABLE WORD 2 REGISTER	Section 15.5.1.6
38h	CPSW_ALE_TBLW1	ADDRESS LOOKUP ENGINE TABLE WORD 1 REGISTER	Section 15.5.1.7
3Ch	CPSW_ALE_TBLW0	ADDRESS LOOKUP ENGINE TABLE WORD 0 REGISTER	Section 15.5.1.8
40h to 54h	CPSW_ALE_PORTCTL_0 to CPSW_ALE_PORTCTL_5	ADDRESS LOOKUP ENGINE PORT x CONTROL REGISTER	Section 15.5.1.9

15.5.1.1 CPSW_ALE_IDVER Register (offset = 0h) [reset = 290104h]

CPSW_ALE_IDVER is shown in Figure 15-15 and described in Table 15-24.

ADDRESS LOOKUP ENGINE ID/VERSION REGISTER

Figure 15-15. CPSW_ALE_IDVER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">IDENT</td></tr>
<tr><td colspan="16">R-29h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">MAJ_VER</td><td colspan="8">MINOR_VER</td></tr>
<tr><td colspan="8">R-1h</td><td colspan="8">R-4h</td></tr>
</table>

Table 15-24. CPSW_ALE_IDVER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	IDENT	R	29h	ALE Identification Value
15-8	MAJ_VER	R	1h	ALE Major Version Value
7-0	MINOR_VER	R	4h	ALE Minor Version Value

15.5.1.2 CPSW_ALE_CTRL Register (offset = 8h) [reset = 0h]

CPSW_ALE_CTRL is shown in Figure 15-16 and described in Table 15-25.

ADDRESS LOOKUP ENGINE CONTROL REGISTER

Figure 15-16. CPSW_ALE_CTRL Register

31	30	29	28	27	26	25	24
EN_ALE	CLR_TBL	AGE_OUT_NOW	RESERVED				
R/W-0h	R/W-0h	R/W-0h	R-0h				
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							EN_P0_UNI_FLOOD
R-0h							R/W-0h
7	6	5	4	3	2	1	0
LEARN_NO_VID	EN_VID0_MODE	EN_OUI_DENY	BYPASS	RATE_LIMIT_TX	VLAN_AWARE	EN_AUTH_MODE	EN_RATE_LIMIT
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-25. CPSW_ALE_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	EN_ALE	R/W	0h	Enable ALE 0h = Drop all packets 1h = Enable ALE packet processing
30	CLR_TBL	R/W	0h	Clear ALE address table - Setting this bit causes the ALE hardware to write all table bit values to zero. Software must perform a clear table operation as part of the ALE setup/configuration process. Setting this bit causes all ALE accesses to be held up for 64 clocks while the clear is performed. Access to all ALE registers will be blocked (wait states) until the 64 clocks have completed. This bit cannot be read as one because the read is blocked until the clear table is completed at which time this bit is cleared to zero.
29	AGE_OUT_NOW	R/W	0h	Age Out Address Table Now - Setting this bit causes the ALE hardware to remove (free up) any ageable table entry that does not have a set touch bit. This bit is cleared when the age out process has completed. This bit may be read. The age out process takes 4096 clocks best case (no ale packet processing during ageout) and 66550 clocks absolute worst case.
28-9	RESERVED	R	0h	
8	EN_P0_UNI_FLOOD	R/W	0h	Enable Port 0 (Host Port) unicast flood 0h = Do not flood unknown unicast packets to host port (p0) 1h = Flood unknown unicast packets to host port (p0)
7	LEARN_NO_VID	R/W	0h	Learn No VID 0h = VID is learned with the source address 1h = VID is not learned with the source address (source address is not tied to VID).
6	EN_VID0_MODE	R/W	0h	Enable VLAN ID = 0 Mode 0h = Process the packet with VID = PORT_VLAN[11 to 0] 1h = Process the packet with VID = 0.

Table 15-25. CPSW_ALE_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	EN_OUI_DENY	R/W	0h	Enable OUI Deny Mode - When set this bit indicates that a packet with a non OUI table entry matching source address will be dropped to the host unless the destination address matches a multicast table entry with the super bit set.
4	BYPASS	R/W	0h	ALE Bypass - When set, all packets received on ports 0 and 1 are sent to the host (only to the host).
3	RATE_LIMIT_TX	R/W	0h	Rate Limit Transmit mode 0h = Broadcast and multicast rate limit counters are received port based 1h = Broadcast and multicast rate limit counters are transmit port based
2	VLAN_AWARE	R/W	0h	ALE VLAN Aware - Determines what is done if VLAN not found. 0h = Flood if VLAN not found 1h = Drop packet if VLAN not found
1	EN_AUTH_MODE	R/W	0h	Enable MAC Authorization Mode - Mac authorization mode requires that all table entries be made by the host software. There are no learned address in authorization mode and the packet will be dropped if the source address is not found (and the destination address is not a multicast address with the super table entry bit set). 0h = The ALE is not in MAC authorization mode. 1h = The ALE is in MAC authorization mode.
0	EN_RATE_LIMIT	R/W	0h	Enable Broadcast and Multicast Rate Limit 0h = Broadcast/Multicast rates not limited 1h = Broadcast/Multicast packet reception limited to the port control register rate limit fields

15.5.1.3 CPSW_ALE_PRESCALE Register (offset = 10h) [reset = 0h]

CPSW_ALE_PRESCALE is shown in Figure 15-17 and described in Table 15-26.

ADDRESS LOOKUP ENGINE PRESCALE REGISTER

Figure 15-17. CPSW_ALE_PRESCALE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												PRESCALE																			
R-0h												R/W-0h																			

Table 15-26. CPSW_ALE_PRESCALE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-0	PRESCALE	R/W	0h	ALE Prescale Register - The input clock is divided by this value for use in the multicast/broadcast rate limiters. The minimum operating value is 0x10. The prescaler is off when the value is zero.

15.5.1.4 CPSW_ALE_UNKNOWN_VLAN Register (offset = 18h) [reset = 0h]

CPSW_ALE_UNKNOWN_VLAN is shown in Figure 15-18 and described in Table 15-27.

ADDRESS LOOKUP ENGINE UNKNOWN VLAN REGISTER

Figure 15-18. CPSW_ALE_UNKNOWN_VLAN Register

31	30	29	28	27	26	25	24
RESERVED		UNKNOWN_FORCE_UNTAGGED_EGRESS					
R-X		R/W-X					

23	22	21	20	19	18	17	16
RESERVED		UNKNOWN_REG_MCAST_FLOOD_MASK					
R-X		R/W-X					

15	14	13	12	11	10	9	8
RESERVED		UNKNOWN_MCAST_FLOOD_MASK					
R-X		R/W-X					

7	6	5	4	3	2	1	0
RESERVED		UNKNOWN_VLAN_MEMBER_LIST					
R-0h		R/W-0h					

Table 15-27. CPSW_ALE_UNKNOWN_VLAN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	X	
29-24	UNKNOWN_FORCE_UNTAGGED_EGRESS	R/W	X	Unknown VLAN Force Untagged Egress.
23-22	RESERVED	R	X	
21-16	UNKNOWN_REG_MCAST_FLOOD_MASK	R/W	X	Unknown VLAN Registered Multicast Flood Mask
15-14	RESERVED	R	X	
13-8	UNKNOWN_MCAST_FLOOD_MASK	R/W	X	Unknown VLAN Multicast Flood Mask
7-6	RESERVED	R	0h	
5-0	UNKNOWN_VLAN_MEMBER_LIST	R/W	0h	Unknown VLAN Member List

15.5.1.5 CPSW_ALE_TBLCTL Register (offset = 20h) [reset = 0h]

CPSW_ALE_TBLCTL is shown in Figure 15-19 and described in Table 15-28.

ADDRESS LOOKUP ENGINE TABLE CONTROL

Figure 15-19. CPSW_ALE_TBLCTL Register

31	30	29	28	27	26	25	24
WRITE_RDZ	RESERVED						
R/W-X	R-X						
23	22	21	20	19	18	17	16
RESERVED							
R-X							
15	14	13	12	11	10	9	8
RESERVED						ENTRY_POINTER	
R-X						R/W-0h	
7	6	5	4	3	2	1	0
ENTRY_POINTER							
R/W-0h							

Table 15-28. CPSW_ALE_TBLCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	WRITE_RDZ	R/W	X	Write Bit - This bit is always read as zero. Writing a 1 to this bit causes the three table word register values to be written to the entry_pointer location in the address table. Writing a 0 to this bit causes the three table word register values to be loaded from the entry_pointer location in the address table so that they may be subsequently read. A read of any ALE address location will be stalled until the read or write has completed.
30-10	RESERVED	R	X	
9-0	ENTRY_POINTER	R/W	0h	Table Entry Pointer - The entry_pointer contains the table entry value that will be read/written with accesses to the table word registers.

15.5.1.6 CPSW_ALE_TBLW2 Register (offset = 34h) [reset = 0h]

CPSW_ALE_TBLW2 is shown in Figure 15-20 and described in Table 15-29.

ADDRESS LOOKUP ENGINE TABLE WORD 2 REGISTER

Figure 15-20. CPSW_ALE_TBLW2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-X															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ENTRY71_64							
R-X								R/W-0h							

Table 15-29. CPSW_ALE_TBLW2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	X	
7-0	ENTRY71_64	R/W	0h	Table entry bits 71:64

15.5.1.7 CPSW_ALE_TBLW1 Register (offset = 38h) [reset = 0h]

CPSW_ALE_TBLW1 is shown in Figure 15-21 and described in Table 15-30.

ADDRESS LOOKUP ENGINE TABLE WORD 1 REGISTER

Figure 15-21. CPSW_ALE_TBLW1 Register

31–0
ENTRY63_32
R/W-0h

Table 15-30. CPSW_ALE_TBLW1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ENTRY63_32	R/W	0h	Table entry bits 63:32

15.5.1.8 CPSW_ALE_TBLW0 Register (offset = 3Ch) [reset = 0h]

CPSW_ALE_TBLW0 is shown in Figure 15-22 and described in Table 15-31.

ADDRESS LOOKUP ENGINE TABLE WORD 0 REGISTER

Figure 15-22. CPSW_ALE_TBLW0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENTRY31_0																															
R/W-0h																															

Table 15-31. CPSW_ALE_TBLW0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ENTRY31_0	R/W	0h	Table entry bits 31:0

15.5.1.9 CPSW_ALE_PORTCTL_0 to CPSW_ALE_PORTCTL_5 Register (offset = 40h to 54h) [reset = 0h]

CPSW_ALE_PORTCTL_0 to CPSW_ALE_PORTCTL_5 is shown in Figure 15-23 and described in Table 15-32.

ADDRESS LOOKUP ENGINE PORT x CONTROL REGISTER

Figure 15-23. CPSW_ALE_PORTCTL_0 to CPSW_ALE_PORTCTL_5 Register

31	30	29	28	27	26	25	24
BCAST_LIMIT							
R/W-0h							
23	22	21	20	19	18	17	16
MCAST_LIMIT							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		NO_SA_UPDATE	NO_LEARN	VID_INGRESS_CHECK	DROP_UNTAGGED	PORT_STATE	
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

Table 15-32. CPSW_ALE_PORTCTL_0 to CPSW_ALE_PORTCTL_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BCAST_LIMIT	R/W	0h	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field
23-16	MCAST_LIMIT	R/W	0h	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field.
15-6	RESERVED	R	0h	
5	NO_SA_UPDATE	R/W	0h	No Souce Address Update - When set the port is disabled from updating the source port number in an ALE table entry.
4	NO_LEARN	R/W	0h	No Learn Mode - When set the port is disabled from learning an address.
3	VID_INGRESS_CHECK	R/W	0h	VLAN ID Ingress Check - If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped.
2	DROP_UNTAGGED	R/W	0h	Drop Untagged Packets - Drop non-VLAN tagged ingress packets.
1-0	PORT_STATE	R/W	0h	0h (R/W) = Disabled 1h (R/W) = Blocked 2h (R/W) = Learn 3h (R/W) = Forward

15.5.2 CPSW_CPDMA Registers

Table 15-33 lists the memory-mapped registers for the CPSW_CPDMA. All register offset addresses not listed in Table 15-33 should be considered as reserved locations and the register contents should not be modified.

Table 15-33. CPSW_CPDMA Registers

Offset	Acronym	Register Name	Section
0h	CPSW_TX_IDVER	CPDMA_REGS TX IDENTIFICATION AND VERSION REGISTER	Section 15.5.2.1
4h	CPSW_TX_CTRL	CPDMA_REGS TX CONTROL REGISTER	Section 15.5.2.2
8h	CPSW_TX_TEARDOWN	CPDMA_REGS TX TEARDOWN REGISTER	Section 15.5.2.3
10h	CPSW_RX_IDVER	CPDMA_REGS RX IDENTIFICATION AND VERSION REGISTER	Section 15.5.2.4
14h	CPSW_RX_CTRL	CPDMA_REGS RX CONTROL REGISTER	Section 15.5.2.5
18h	CPSW_RX_TEARDOWN	CPDMA_REGS RX TEARDOWN REGISTER	Section 15.5.2.6
1Ch	CPSW_CPDMA_SOFT_RESET	CPDMA_REGS SOFT RESET REGISTER	Section 15.5.2.7
20h	CPSW_DMACTRL	CPDMA_REGS CPDMA CONTROL REGISTER	Section 15.5.2.8
24h	CPSW_DMASTS	CPDMA_REGS CPDMA STATUS REGISTER	Section 15.5.2.9
28h	CPSW_RX_BUFFER_OFFSET	CPDMA_REGS RECEIVE BUFFER OFFSET	Section 15.5.2.10
2Ch	CPSW_EMCTRL	CPDMA_REGS EMULATION CONTROL	Section 15.5.2.11
30h	CPSW_TX_PRI0_RATE	CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 0 RATE	Section 15.5.2.12
34h	CPSW_TX_PRI1_RATE	CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 1 RATE	Section 15.5.2.13
38h	CPSW_TX_PRI2_RATE	CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 2 RATE	Section 15.5.2.14
3Ch	CPSW_TX_PRI3_RATE	CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 3 RATE	Section 15.5.2.15
40h	CPSW_TX_PRI4_RATE	CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 4 RATE	Section 15.5.2.16
44h	CPSW_TX_PRI5_RATE	CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 5 RATE	Section 15.5.2.17
48h	CPSW_TX_PRI6_RATE	CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 6 RATE	Section 15.5.2.18
4Ch	CPSW_TX_PRI7_RATE	CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 7 RATE	Section 15.5.2.19
80h	CPSW_TX_INTSTAT_RAW	CPDMA_INT TX INTERRUPT STATUS REGISTER (RAW VALUE)	Section 15.5.2.20
84h	CPSW_TX_INTSTAT_MASKED	CPDMA_INT TX INTERRUPT STATUS REGISTER (MASKED VALUE)	Section 15.5.2.21
88h	CPSW_TX_INTMASK_SET	CPDMA_INT TX INTERRUPT MASK SET REGISTER	Section 15.5.2.22
8Ch	CPSW_TX_INTMASK_CLR	CPDMA_INT TX INTERRUPT MASK CLEAR REGISTER	Section 15.5.2.23
90h	CPSW_CPDMA_IN_VECTOR	CPDMA_INT INPUT VECTOR (READ ONLY)	Section 15.5.2.24
94h	CPSW_CPDMA_EOI_VECTOR	CPDMA_INT END OF INTERRUPT VECTOR	Section 15.5.2.25
A0h	CPSW_RX_INTSTAT_RAW	CPDMA_INT RX INTERRUPT STATUS REGISTER (RAW VALUE)	Section 15.5.2.26
A4h	CPSW_RX_INTSTAT_MASKED	CPDMA_INT RX INTERRUPT STATUS REGISTER (MASKED VALUE)	Section 15.5.2.27
A8h	CPSW_RX_INTMASK_SET	CPDMA_INT RX INTERRUPT MASK SET REGISTER	Section 15.5.2.28
ACh	CPSW_RX_INTMASK_CLR	CPDMA_INT RX INTERRUPT MASK CLEAR REGISTER	Section 15.5.2.29
B0h	CPSW_DMA_INTSTAT_RAW	CPDMA_INT DMA INTERRUPT STATUS REGISTER (RAW VALUE)	Section 15.5.2.30
B4h	CPSW_DMA_INTSTAT_MASKED	CPDMA_INT DMA INTERRUPT STATUS REGISTER (MASKED VALUE)	Section 15.5.2.31
B8h	CPSW_DMA_INTMASK_SET	CPDMA_INT DMA INTERRUPT MASK SET REGISTER	Section 15.5.2.32
BCh	CPSW_DMA_INTMASK_CLR	CPDMA_INT DMA INTERRUPT MASK CLEAR REGISTER	Section 15.5.2.33

Table 15-33. CPSW_CPDMA Registers (continued)

Offset	Acronym	Register Name	Section
C0h	CPSW_RX0_PENDTHRESH	CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 0	Section 15.5.2.34
C4h	CPSW_RX1_PENDTHRESH	CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 1	Section 15.5.2.35
C8h	CPSW_RX2_PENDTHRESH	CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 2	Section 15.5.2.36
CCh	CPSW_RX3_PENDTHRESH	CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 3	Section 15.5.2.37
D0h	CPSW_RX4_PENDTHRESH	CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 4	Section 15.5.2.38
D4h	CPSW_RX5_PENDTHRESH	CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 5	Section 15.5.2.39
D8h	CPSW_RX6_PENDTHRESH	CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 6	Section 15.5.2.40
DCh	CPSW_RX7_PENDTHRESH	CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 7	Section 15.5.2.41
E0h	CPSW_RX0_FREEBUFFER	CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 0	Section 15.5.2.42
E4h	CPSW_RX1_FREEBUFFER	CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 1	Section 15.5.2.43
E8h	CPSW_RX2_FREEBUFFER	CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 2	Section 15.5.2.44
ECh	CPSW_RX3_FREEBUFFER	CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 3	Section 15.5.2.45
F0h	CPSW_RX4_FREEBUFFER	CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 4	Section 15.5.2.46
F4h	CPSW_RX5_FREEBUFFER	CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 5	Section 15.5.2.47
F8h	CPSW_RX6_FREEBUFFER	CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 6	Section 15.5.2.48
FCh	CPSW_RX7_FREEBUFFER	CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 7	Section 15.5.2.49

15.5.2.1 CPSW_TX_IDVER Register (offset = 0h) [reset = 180108h]

CPSW_TX_IDVER is shown in Figure 15-24 and described in Table 15-34.

CPDMA_REGS TX IDENTIFICATION AND VERSION REGISTER

Figure 15-24. CPSW_TX_IDVER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">TX_IDENT</td></tr>
<tr><td colspan="16">R-18h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">TX_MAJOR_VER</td><td colspan="8">TX_MINOR_VER</td></tr>
<tr><td colspan="8">R-1h</td><td colspan="8">R-8h</td></tr>
</table>

Table 15-34. CPSW_TX_IDVER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TX_IDENT	R	18h	TX Identification Value
15-8	TX_MAJOR_VER	R	1h	TX Major Version Value - The value read is the major version number
7-0	TX_MINOR_VER	R	8h	TX Minor Version Value - The value read is the minor version number

15.5.2.2 CPSW_TX_CTRL Register (offset = 4h) [reset = 0h]

CPSW_TX_CTRL is shown in Figure 15-25 and described in Table 15-35.

CPDMA_REGS TX CONTROL REGISTER

Figure 15-25. CPSW_TX_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TX_EN
R-0h							R/W-0h

Table 15-35. CPSW_TX_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TX_EN	R/W	0h	TX Enable 0h = Disabled 1h = Enabled

15.5.2.3 CPSW_TX_TEARDOWN Register (offset = 8h) [reset = 0h]

CPSW_TX_TEARDOWN is shown in Figure 15-26 and described in Table 15-36.

CPDMA_REGS TX TEARDOWN REGISTER

Figure 15-26. CPSW_TX_TEARDOWN Register

31	30	29	28	27	26	25	24
TX_TDN_RDY	RESERVED						
R-0h	R-0h						
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					TX_TDN_CH		
R-0h					R/W-0h		

Table 15-36. CPSW_TX_TEARDOWN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TX_TDN_RDY	R	0h	Tx Teardown Ready - read as zero, but is always assumed to be one (unused).
30-3	RESERVED	R	0h	
2-0	TX_TDN_CH	R/W	0h	Tx Teardown Channel - Transmit channel teardown is commanded by writing the encoded value of the transmit channel to be torn down. The teardown register is read as zero. 000 - teardown transmit channel 0 ... 111 - teardown transmit channel 7

15.5.2.4 CPSW_RX_IDVER Register (offset = 10h) [reset = C0107h]

CPSW_RX_IDVER is shown in Figure 15-27 and described in Table 15-37.

CPDMA_REGS RX IDENTIFICATION AND VERSION REGISTER

Figure 15-27. CPSW_RX_IDVER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RX_IDENT															
R-Ch															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_MAJOR_VER								RX_MINOR_VER							
R-1h								R-7h							

Table 15-37. CPSW_RX_IDVER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RX_IDENT	R	Ch	RX Identification Value
15-8	RX_MAJOR_VER	R	1h	RX Major Version Value
7-0	RX_MINOR_VER	R	7h	RX Minor Version Value

15.5.2.5 CPSW_RX_CTRL Register (offset = 14h) [reset = 0h]

CPSW_RX_CTRL is shown in Figure 15-28 and described in Table 15-38.

CPDMA_REGS RX CONTROL REGISTER

Figure 15-28. CPSW_RX_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							RX_EN
R-0h							R/W-0h

Table 15-38. CPSW_RX_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	RX_EN	R/W	0h	RX DMA Enable 0h = Disabled 1h = Enabled

15.5.2.6 CPSW_RX_TEARDOWN Register (offset = 18h) [reset = 0h]

CPSW_RX_TEARDOWN is shown in Figure 15-29 and described in Table 15-39.

CPDMA_REGS RX TEARDOWN REGISTER

Figure 15-29. CPSW_RX_TEARDOWN Register

31	30	29	28	27	26	25	24
RX_TDN_RDY	RESERVED						
R-0h	R-0h						
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					RX_TDN_CH		
R-0h					R/W-0h		

Table 15-39. CPSW_RX_TEARDOWN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RX_TDN_RDY	R	0h	Teardown Ready - read as zero, but is always assumed to be one (unused).
30-3	RESERVED	R	0h	
2-0	RX_TDN_CH	R/W	0h	Rx Teardown Channel -Receive channel teardown is commanded by writing the encoded value of the receive channel to be torn down. The teardown register is read as zero. 000 - teardown receive channel 0 ... 111 - teardown receive channel 7

15.5.2.7 CPSW_CPDMA_SOFT_RESET Register (offset = 1Ch) [reset = 0h]

CPSW_CPDMA_SOFT_RESET is shown in Figure 15-30 and described in Table 15-40.

CPDMA_REGS SOFT RESET REGISTER

Figure 15-30. CPSW_CPDMA_SOFT_RESET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							SOFT_RESET
R-0h							R/W-0h

Table 15-40. CPSW_CPDMA_SOFT_RESET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	SOFT_RESET	R/W	0h	Software reset - Writing a one to this bit causes the CPDMA logic to be reset. Software reset occurs when the RX and TX DMA Controllers are in an idle state to avoid locking up the VBUSP bus. After writing a one to this bit, it may be polled to determine if the reset has occurred. If a one is read, the reset has not yet occurred. If a zero is read then reset has occurred.

15.5.2.8 CPSW_DMACTRL Register (offset = 20h) [reset = 0h]

CPSW_DMACTRL is shown in Figure 15-31 and described in Table 15-41.

CPDMA_REGS CPDMA CONTROL REGISTER

Figure 15-31. CPSW_DMACTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TX_RLIM							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED			RX_CEF	CMD_IDLE	RX_OFFLEN_BLOCK	RX_OWNERSHIP	TX_PTYPE
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-41. CPSW_DMACTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	TX_RLIM	R/W	0h	Transmit Rate Limit Channel Bus 00000000 - no rate-limited channels 10000000 - channel 7 is rate-limited 11000000 - channels 7 downto 6 are rate-limited 11100000 - channels 7 downto 5 are rate-limited 11110000 - channels 7 downto 4 are rate-limited 11111000 - channels 7 downto 3 are rate-limited 11111100 - channels 7 downto 2 are rate-limited 11111110 - channels 7 downto 1 are rate-limited 11111111 - channels 7 downto 0 are rate-limited all others invalid - this bus must be set msb towards lsb. tx_ptype must be set if any tx_rlim bit is set for fixed priority.
7-5	RESERVED	R	0h	
4	RX_CEF	R/W	0h	RX Copy Error Frames Enable - Enables DMA overrun frames to be transferred to memory (up to the point of overrun). The overrun error bit will be set in the frame EOP buffer descriptor. Overrun frame data will be filtered when rx_cef is not set. Frames coming from the receive FIFO with other error bits set are not effected by this bit. 0h = Frames containing overrun errors are filtered 1h = Frames containing overrun errors are transferred to memory
3	CMD_IDLE	R/W	0h	Command Idle 0h = Idle not commanded 1h = Idle Commanded (read idle in DMAStatus)
2	RX_OFFLEN_BLOCK	R/W	0h	Receive Offset/Length word write block. 0h = Do not block the DMA writes to the receive buffer descriptor offset/buffer length word. The offset/buffer length word is written as specified in CPPI 3.0. 1h = Block all CPDMA DMA controller writes to the receive buffer descriptor offset/buffer length words during CPPI packet processing. when this bit is set, the CPDMA will never write the third word to any receive buffer descriptor.

Table 15-41. CPSW_DMACTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	RX_OWNERSHIP	R/W	0h	Receive Ownership Write Bit Value. 0h = The CPDMA writes the receive ownership bit to zero at the end of packet processing as specified in CPPI 3.0. 1h = The CPDMA writes the receive ownership bit to one at the end of packet processing. Users who do not use the ownership mechanism can use this mode to preclude the necessity of software having to set this bit each timethe buffer descriptor is used.
0	TX_PTYPE	R/W	0h	Transmit Queue Priority Type 0h = The queue uses a round robin scheme to select the next channel for transmission. 1h = The queue uses a fixed (channel 7 highest priority) priority scheme to select the next channel for transmission

15.5.2.9 CPSW_DMASTS Register (offset = 24h) [reset = 0h]

CPSW_DMASTS is shown in Figure 15-32 and described in Table 15-42.

CPDMA_REGS CPDMA STATUS REGISTER

Figure 15-32. CPSW_DMASTS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>IDLE</td><td colspan="7">RESERVED</td></tr>
<tr><td>R-0h</td><td colspan="7">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">TX_HOST_ERR_CODE</td><td>RESERVED</td><td colspan="3">TX_ERR_CH</td></tr>
<tr><td colspan="4">R-0h</td><td>R-0h</td><td colspan="3">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="4">RX_HOST_ERR_CODE</td><td>RESERVED</td><td colspan="3">RX_ERR_CH</td></tr>
<tr><td colspan="4">R-0h</td><td>R-0h</td><td colspan="3">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
</table>

Table 15-42. CPSW_DMASTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IDLE	R	0h	Idle Status Bit - Indicates when set that the CPDMA is not transferring a packet on transmit or receive.
30-24	RESERVED	R	0h	
23-20	TX_HOST_ERR_CODE	R	0h	TX Host Error Code - This field is set to indicate CPDMA detected TX DMA related host errors. The host should read this field after a HOST_ERR_INT to determine the error. Host error Interrupts require hardware reset in order to recover. A zero packet length is an error, but it is not detected. 0000 - No error 0001 - SOP error. 0010 - Ownership bit not set in SOP buffer. 0011 - Zero Next Buffer Descriptor Pointer Without EOP 0100 - Zero Buffer Pointer. 0101 - Zero Buffer Length 0110 - Packet Length Error (sum of buffers is less than packet length) 0111 - reserved ... 1111 - reserved
19	RESERVED	R	0h	
18-16	TX_ERR_CH	R	0h	TX Host Error Channel - This field indicates which TX channel (if applicable) the host error occurred on. This field is cleared to zero on a host read. 000 - The host error occurred on TX channel 0 ... 111 - The host error occurred on TX channel 7
15-12	RX_HOST_ERR_CODE	R	0h	RX Host Error Code - This field is set to indicate CPDMA detected RX DMA related host errors. The host should read this field after a HOST_ERR_INT to determine the error. Host error Interrupts require hardware reset in order to recover. 0000 - No error 0001 - reserved 0010 - Ownership bit not set in input buffer. 0011 - reserved 0100 - Zero Buffer Pointer. 0101 - Zero buffer length on non-SOP descriptor 0110 - SOP buffer length not greater than offset ... 1111 - reserved
11	RESERVED	R	0h	

Table 15-42. CPSW_DMASTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-8	RX_ERR_CH	R	0h	RX Host Error Channel - This field indicates which RX channel the host error occurred on. This field is cleared to zero on a host read. 000 - The host error occurred on RX channel 0 ... 111 - The host error occurred on RX channel 7
7-0	RESERVED	R	0h	

15.5.2.10 CPSW_RX_BUFFER_OFFSET Register (offset = 28h) [reset = 0h]

CPSW_RX_BUFFER_OFFSET is shown in Figure 15-33 and described in Table 15-43.

CPDMA_REGS RECEIVE BUFFER OFFSET

Figure 15-33. CPSW_RX_BUFFER_OFFSET Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">RX_BUFFER_OFFSET</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">R/W-0h</td></tr>
</table>

Table 15-43. CPSW_RX_BUFFER_OFFSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	RX_BUFFER_OFFSET	R/W	0h	Receive Buffer Offset Value - The rx_buffer_offset will be written by the port into each frame SOP buffer descriptor buffer_offset field. The frame data will begin after the rx_buffer_offset value of bytes. A value of 0x0000 indicates that there are no unused bytes at the beginning of the data and that valid data begins on the first byte of the buffer. A value of 0x000F (decimal 15) indicates that the first 15 bytes of the buffer are to be ignored by the port and that valid buffer data starts on byte 16 of the buffer. This value is used for all channels.

15.5.2.11 CPSW_EMCTRL Register (offset = 2Ch) [reset = 0h]

CPSW_EMCTRL is shown in Figure 15-34 and described in Table 15-44.

CPDMA_REGS EMULATION CONTROL

Figure 15-34. CPSW_EMCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						SOFT	FREE
R-0h						R/W-0h	R/W-0h

Table 15-44. CPSW_EMCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	SOFT	R/W	0h	Emulation Soft Bit
0	FREE	R/W	0h	Emulation Free Bit

15.5.2.12 CPSW_TX_PRI0_RATE Register (offset = 30h) [reset = 0h]

CPSW_TX_PRI0_RATE is shown in Figure 15-35 and described in Table 15-45.

CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 0 RATE

Figure 15-35. CPSW_TX_PRI0_RATE Register

31	30	29	28	27	26	25	24
RESERVED		PRIN_IDLE_CNT					
R-0h		R/W-0h					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRIN_IDLE_CNT							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED		PRIN_SEND_CNT					
R-0h		R/W-0h					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRIN_SEND_CNT							
R/W-0h							

Table 15-45. CPSW_TX_PRI0_RATE Register Field Descriptions

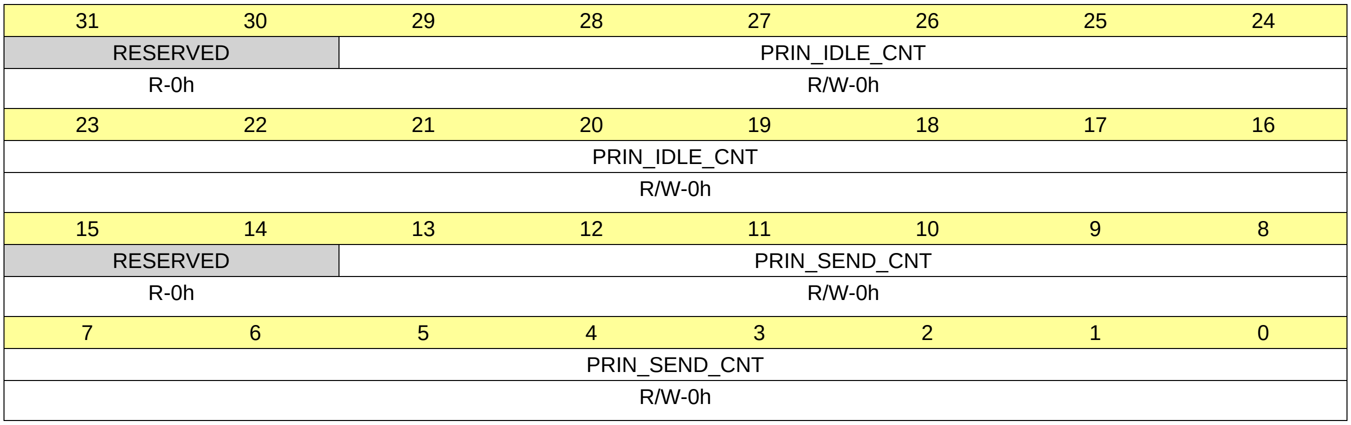
Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	
29-16	PRIN_IDLE_CNT	R/W	0h	Priority (7:0) idle count
15-14	RESERVED	R	0h	
13-0	PRIN_SEND_CNT	R/W	0h	Priority (7:0) send count

15.5.2.13 CPSW_TX_PRI1_RATE Register (offset = 34h) [reset = 0h]

CPSW_TX_PRI1_RATE is shown in Figure 15-36 and described in Table 15-46.

CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 1 RATE

Figure 15-36. CPSW_TX_PRI1_RATE Register



<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">PRIN_IDLE_CNT</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="6">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">PRIN_IDLE_CNT</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">PRIN_SEND_CNT</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="6">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">PRIN_SEND_CNT</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 15-46. CPSW_TX_PRI1_RATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	
29-16	PRIN_IDLE_CNT	R/W	0h	Priority (7:0) idle count
15-14	RESERVED	R	0h	
13-0	PRIN_SEND_CNT	R/W	0h	Priority (7:0) send count

15.5.2.14 CPSW_TX_PRI2_RATE Register (offset = 38h) [reset = 0h]

CPSW_TX_PRI2_RATE is shown in Figure 15-37 and described in Table 15-47.

CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 2 RATE

Figure 15-37. CPSW_TX_PRI2_RATE Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">PRIN_IDLE_CNT</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="6">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">PRIN_IDLE_CNT</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">PRIN_SEND_CNT</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="6">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">PRIN_SEND_CNT</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 15-47. CPSW_TX_PRI2_RATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	
29-16	PRIN_IDLE_CNT	R/W	0h	Priority (7:0) idle count
15-14	RESERVED	R	0h	
13-0	PRIN_SEND_CNT	R/W	0h	Priority (7:0) send count

15.5.2.15 CPSW_TX_PRI3_RATE Register (offset = 3Ch) [reset = 0h]

CPSW_TX_PRI3_RATE is shown in Figure 15-38 and described in Table 15-48.

CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 3 RATE

Figure 15-38. CPSW_TX_PRI3_RATE Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">PRIN_IDLE_CNT</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="6">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">PRIN_IDLE_CNT</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">PRIN_SEND_CNT</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="6">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">PRIN_SEND_CNT</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 15-48. CPSW_TX_PRI3_RATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	
29-16	PRIN_IDLE_CNT	R/W	0h	Priority (7:0) idle count
15-14	RESERVED	R	0h	
13-0	PRIN_SEND_CNT	R/W	0h	Priority (7:0) send count

15.5.2.16 CPSW_TX_PRI4_RATE Register (offset = 40h) [reset = 0h]

CPSW_TX_PRI4_RATE is shown in Figure 15-39 and described in Table 15-49.

CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 4 RATE

Figure 15-39. CPSW_TX_PRI4_RATE Register

31	30	29	28	27	26	25	24
RESERVED		PRIN_IDLE_CNT					
R-0h		R/W-0h					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRIN_IDLE_CNT							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED		PRIN_SEND_CNT					
R-0h		R/W-0h					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRIN_SEND_CNT							
R/W-0h							

Table 15-49. CPSW_TX_PRI4_RATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	
29-16	PRIN_IDLE_CNT	R/W	0h	Priority (7:0) idle count
15-14	RESERVED	R	0h	
13-0	PRIN_SEND_CNT	R/W	0h	Priority (7:0) send count

15.5.2.17 CPSW_TX_PRI5_RATE Register (offset = 44h) [reset = 0h]

CPSW_TX_PRI5_RATE is shown in Figure 15-40 and described in Table 15-50.

CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 5 RATE

Figure 15-40. CPSW_TX_PRI5_RATE Register

31	30	29	28	27	26	25	24
RESERVED		PRIN_IDLE_CNT					
R-0h		R/W-0h					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRIN_IDLE_CNT							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED		PRIN_SEND_CNT					
R-0h		R/W-0h					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRIN_SEND_CNT							
R/W-0h							

Table 15-50. CPSW_TX_PRI5_RATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	
29-16	PRIN_IDLE_CNT	R/W	0h	Priority (7:0) idle count
15-14	RESERVED	R	0h	
13-0	PRIN_SEND_CNT	R/W	0h	Priority (7:0) send count

15.5.2.18 CPSW_TX_PRI6_RATE Register (offset = 48h) [reset = 0h]

CPSW_TX_PRI6_RATE is shown in Figure 15-41 and described in Table 15-51.

CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 6 RATE

Figure 15-41. CPSW_TX_PRI6_RATE Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">PRIN_IDLE_CNT</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="6">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">PRIN_IDLE_CNT</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">PRIN_SEND_CNT</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="6">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">PRIN_SEND_CNT</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 15-51. CPSW_TX_PRI6_RATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	
29-16	PRIN_IDLE_CNT	R/W	0h	Priority (7:0) idle count
15-14	RESERVED	R	0h	
13-0	PRIN_SEND_CNT	R/W	0h	Priority (7:0) send count

15.5.2.19 CPSW_TX_PRI7_RATE Register (offset = 4Ch) [reset = 0h]

CPSW_TX_PRI7_RATE is shown in Figure 15-42 and described in Table 15-52.

CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 7 RATE

Figure 15-42. CPSW_TX_PRI7_RATE Register

31	30	29	28	27	26	25	24
RESERVED		PRIN_IDLE_CNT					
R-0h		R/W-0h					
23	22	21	20	19	18	17	16
PRIN_IDLE_CNT							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED		PRIN_SEND_CNT					
R-0h		R/W-0h					
7	6	5	4	3	2	1	0
PRIN_SEND_CNT							
R/W-0h							

Table 15-52. CPSW_TX_PRI7_RATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	
29-16	PRIN_IDLE_CNT	R/W	0h	Priority (7:0) idle count
15-14	RESERVED	R	0h	
13-0	PRIN_SEND_CNT	R/W	0h	Priority (7:0) send count

15.5.2.20 CPSW_TX_INTSTAT_RAW Register (offset = 80h) [reset = 0h]

CPSW_TX_INTSTAT_RAW is shown in Figure 15-43 and described in Table 15-53.

CPDMA_INT TX INTERRUPT STATUS REGISTER (RAW VALUE)

Figure 15-43. CPSW_TX_INTSTAT_RAW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>TX7_PEND</td><td>TX6_PEND</td><td>TX5_PEND</td><td>TX4_PEND</td><td>TX3_PEND</td><td>TX2_PEND</td><td>TX1_PEND</td><td>TX0_PEND</td></tr>
<tr><td>R-0h</td><td>R-0h</td><td>R-0h</td><td>R-0h</td><td>R-0h</td><td>R-0h</td><td>R-0h</td><td>R-0h</td></tr>
</table>

Table 15-53. CPSW_TX_INTSTAT_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	TX7_PEND	R	0h	TX7_PEND raw int read (before mask).
6	TX6_PEND	R	0h	TX6_PEND raw int read (before mask).
5	TX5_PEND	R	0h	TX5_PEND raw int read (before mask).
4	TX4_PEND	R	0h	TX4_PEND raw int read (before mask).
3	TX3_PEND	R	0h	TX3_PEND raw int read (before mask).
2	TX2_PEND	R	0h	TX2_PEND raw int read (before mask).
1	TX1_PEND	R	0h	TX1_PEND raw int read (before mask).
0	TX0_PEND	R	0h	TX0_PEND raw int read (before mask).

15.5.2.21 CPSW_TX_INTSTAT_MASKED Register (offset = 84h) [reset = 0h]

CPSW_TX_INTSTAT_MASKED is shown in Figure 15-44 and described in Table 15-54.

CPDMA_INT TX INTERRUPT STATUS REGISTER (MASKED VALUE)

Figure 15-44. CPSW_TX_INTSTAT_MASKED Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TX7_PEND	TX6_PEND	TX5_PEND	TX4_PEND	TX3_PEND	TX2_PEND	TX1_PEND	TX0_PEND
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 15-54. CPSW_TX_INTSTAT_MASKED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	TX7_PEND	R	0h	TX7_PEND masked interrupt read.
6	TX6_PEND	R	0h	TX6_PEND masked interrupt read.
5	TX5_PEND	R	0h	TX5_PEND masked interrupt read.
4	TX4_PEND	R	0h	TX4_PEND masked interrupt read.
3	TX3_PEND	R	0h	TX3_PEND masked interrupt read.
2	TX2_PEND	R	0h	TX2_PEND masked interrupt read.
1	TX1_PEND	R	0h	TX1_PEND masked interrupt read.
0	TX0_PEND	R	0h	TX0_PEND masked interrupt read.

15.5.2.22 CPSW_TX_INTMASK_SET Register (offset = 88h) [reset = 0h]

CPSW_TX_INTMASK_SET is shown in Figure 15-45 and described in Table 15-55.

CPDMA_INT TX INTERRUPT MASK SET REGISTER

Figure 15-45. CPSW_TX_INTMASK_SET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TX7_MASK	TX6_MASK	TX5_MASK	TX4_MASK	TX3_MASK	TX2_MASK	TX1_MASK	TX0_MASK
W-0h	R-0h	W-0h	R-0h	W-0h	R-0h	W-0h	R-0h

Table 15-55. CPSW_TX_INTMASK_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	TX7_MASK	W	0h	TX Channel 7 Mask - Write one to enable interrupt.
6	TX6_MASK	R	0h	TX Channel 6 Mask - Write one to enable interrupt.
5	TX5_MASK	W	0h	TX Channel 5 Mask - Write one to enable interrupt.
4	TX4_MASK	R	0h	TX Channel 4 Mask - Write one to enable interrupt.
3	TX3_MASK	W	0h	TX Channel 3 Mask - Write one to enable interrupt.
2	TX2_MASK	R	0h	TX Channel 2 Mask - Write one to enable interrupt.
1	TX1_MASK	W	0h	TX Channel 1 Mask - Write one to enable interrupt.
0	TX0_MASK	R	0h	TX Channel 0 Mask - Write one to enable interrupt.

15.5.2.23 CPSW_TX_INTMASK_CLR Register (offset = 8Ch) [reset = 0h]

CPSW_TX_INTMASK_CLR is shown in Figure 15-46 and described in Table 15-56.

CPDMA_INT TX INTERRUPT MASK CLEAR REGISTER

Figure 15-46. CPSW_TX_INTMASK_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
TX7_MASK	TX6_MASK	TX5_MASK	TX4_MASK	TX3_MASK	TX2_MASK	TX1_MASK	TX0_MASK
W-0h	R-0h	W-0h	R-0h	W-0h	R-0h	W-0h	R-0h

Table 15-56. CPSW_TX_INTMASK_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	TX7_MASK	W	0h	TX Channel 7 Mask - Write one to disable interrupt.
6	TX6_MASK	R	0h	TX Channel 6 Mask - Write one to disable interrupt.
5	TX5_MASK	W	0h	TX Channel 5 Mask - Write one to disable interrupt.
4	TX4_MASK	R	0h	TX Channel 4 Mask - Write one to disable interrupt.
3	TX3_MASK	W	0h	TX Channel 3 Mask - Write one to disable interrupt.
2	TX2_MASK	R	0h	TX Channel 2 Mask - Write one to disable interrupt.
1	TX1_MASK	W	0h	TX Channel 1 Mask - Write one to disable interrupt.
0	TX0_MASK	R	0h	TX Channel 0 Mask - Write one to disable interrupt.

15.5.2.24 CPSW_CPDMA_IN_VECTOR Register (offset = 90h) [reset = 0h]

CPSW_CPDMA_IN_VECTOR is shown in Figure 15-47 and described in Table 15-57.

CPDMA_INT INPUT VECTOR (READ ONLY)

Figure 15-47. CPSW_CPDMA_IN_VECTOR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA_IN_VECTOR																															
R-0h																															

Table 15-57. CPSW_CPDMA_IN_VECTOR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DMA_IN_VECTOR	R	0h	DMA Input Vector - The value of DMA_In_Vector is reset to zero, but will change to the IN_VECTOR bus value one clock after reset is deasserted. Thereafter, this value will change to a new IN_VECTOR value one clock after the IN_VECTOR value changes.

15.5.2.25 CPSW_CPDMA_EOI_VECTOR Register (offset = 94h) [reset = 0h]

CPSW_CPDMA_EOI_VECTOR is shown in Figure 15-48 and described in Table 15-58.

CPDMA_INT END OF INTERRUPT VECTOR

Figure 15-48. CPSW_CPDMA_EOI_VECTOR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="11">RESERVED</td><td colspan="5">DMA_EOI_VECTOR</td></tr>
<tr><td colspan="11">R-0h</td><td colspan="5">R/W-0h</td></tr>
</table>

Table 15-58. CPSW_CPDMA_EOI_VECTOR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4-0	DMA_EOI_VECTOR	R/W	0h	DMA End of Interrupt Vector - The EOI_VECTOR(4:0) pins reflect the value written to this location one CLK cycle after a write to this location. The EOI_WR signal is asserted for a single clock cycle after a latency of two CLK cycles when a write is performed to this location.

15.5.2.26 CPSW_RX_INTSTAT_RAW Register (offset = A0h) [reset = 0h]

CPSW_RX_INTSTAT_RAW is shown in Figure 15-49 and described in Table 15-59.

CPDMA_INT RX INTERRUPT STATUS REGISTER (RAW VALUE)

Figure 15-49. CPSW_RX_INTSTAT_RAW Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RX7_THRESH_PEND	RX6_THRESH_PEND	RX5_THRESH_PEND	RX4_THRESH_PEND	RX3_THRESH_PEND	RX2_THRESH_PEND	RX1_THRESH_PEND	RX0_THRESH_PEND
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RX7_PEND	RX6_PEND	RX5_PEND	RX4_PEND	RX3_PEND	RX2_PEND	RX1_PEND	RX0_PEND
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 15-59. CPSW_RX_INTSTAT_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	RX7_THRESH_PEND	R	0h	RX7_THRESH_PEND raw int read (before mask).
14	RX6_THRESH_PEND	R	0h	RX6_THRESH_PEND raw int read (before mask).
13	RX5_THRESH_PEND	R	0h	RX5_THRESH_PEND raw int read (before mask).
12	RX4_THRESH_PEND	R	0h	RX4_THRESH_PEND raw int read (before mask).
11	RX3_THRESH_PEND	R	0h	RX3_THRESH_PEND raw int read (before mask).
10	RX2_THRESH_PEND	R	0h	RX2_THRESH_PEND raw int read (before mask).
9	RX1_THRESH_PEND	R	0h	RX1_THRESH_PEND raw int read (before mask).
8	RX0_THRESH_PEND	R	0h	RX0_THRESH_PEND raw int read (before mask).
7	RX7_PEND	R	0h	RX7_PEND raw int read (before mask).
6	RX6_PEND	R	0h	RX6_PEND raw int read (before mask).
5	RX5_PEND	R	0h	RX5_PEND raw int read (before mask).
4	RX4_PEND	R	0h	RX4_PEND raw int read (before mask).
3	RX3_PEND	R	0h	RX3_PEND raw int read (before mask).
2	RX2_PEND	R	0h	RX2_PEND raw int read (before mask).
1	RX1_PEND	R	0h	RX1_PEND raw int read (before mask).
0	RX0_PEND	R	0h	RX0_PEND raw int read (before mask).

15.5.2.27 CPSW_RX_INTSTAT_MASKED Register (offset = A4h) [reset = 0h]

CPSW_RX_INTSTAT_MASKED is shown in Figure 15-50 and described in Table 15-60.

CPDMA_INT RX INTERRUPT STATUS REGISTER (MASKED VALUE)

Figure 15-50. CPSW_RX_INTSTAT_MASKED Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RX7_THRESH_PEND	RX6_THRESH_PEND	RX5_THRESH_PEND	RX4_THRESH_PEND	RX3_THRESH_PEND	RX2_THRESH_PEND	RX1_THRESH_PEND	RX0_THRESH_PEND
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RX7_PEND	RX6_PEND	RX5_PEND	RX4_PEND	RX3_PEND	RX2_PEND	RX1_PEND	RX0_PEND
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 15-60. CPSW_RX_INTSTAT_MASKED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	RX7_THRESH_PEND	R	0h	RX7_THRESH_PEND masked int read.
14	RX6_THRESH_PEND	R	0h	RX6_THRESH_PEND masked int read.
13	RX5_THRESH_PEND	R	0h	RX5_THRESH_PEND masked int read.
12	RX4_THRESH_PEND	R	0h	RX4_THRESH_PEND masked int read.
11	RX3_THRESH_PEND	R	0h	RX3_THRESH_PEND masked int read.
10	RX2_THRESH_PEND	R	0h	RX2_THRESH_PEND masked int read.
9	RX1_THRESH_PEND	R	0h	RX1_THRESH_PEND masked int read.
8	RX0_THRESH_PEND	R	0h	RX0_THRESH_PEND masked int read.
7	RX7_PEND	R	0h	RX7_PEND masked int read.
6	RX6_PEND	R	0h	RX6_PEND masked int read.
5	RX5_PEND	R	0h	RX5_PEND masked int read.
4	RX4_PEND	R	0h	RX4_PEND masked int read.
3	RX3_PEND	R	0h	RX3_PEND masked int read.
2	RX2_PEND	R	0h	RX2_PEND masked int read.
1	RX1_PEND	R	0h	RX1_PEND masked int read.
0	RX0_PEND	R	0h	RX0_PEND masked int read.

15.5.2.28 CPSW_RX_INTMASK_SET Register (offset = A8h) [reset = 0h]

CPSW_RX_INTMASK_SET is shown in Figure 15-51 and described in Table 15-61.

CPDMA_INT RX INTERRUPT MASK SET REGISTER

Figure 15-51. CPSW_RX_INTMASK_SET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RX7_THRESH_PEND_MASK	RX6_THRESH_PEND_MASK	RX5_THRESH_PEND_MASK	RX4_THRESH_PEND_MASK	RX3_THRESH_PEND_MASK	RX2_THRESH_PEND_MASK	RX1_THRESH_PEND_MASK	RX0_THRESH_PEND_MASK
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RX7_PEND_MASK	RX6_PEND_MASK	RX5_PEND_MASK	RX4_PEND_MASK	RX3_PEND_MASK	RX2_PEND_MASK	RX1_PEND_MASK	RX0_PEND_MASK
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-61. CPSW_RX_INTMASK_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	RX7_THRESH_PEND_MASK	R/W	0h	RX Channel 7 Threshold Pending Int. Mask - Write one to enable Int.
14	RX6_THRESH_PEND_MASK	R/W	0h	RX Channel 6 Threshold Pending Int. Mask - Write one to enable Int.
13	RX5_THRESH_PEND_MASK	R/W	0h	RX Channel 5 Threshold Pending Int. Mask - Write one to enable Int.
12	RX4_THRESH_PEND_MASK	R/W	0h	RX Channel 4 Threshold Pending Int. Mask - Write one to enable Int.
11	RX3_THRESH_PEND_MASK	R/W	0h	RX Channel 3 Threshold Pending Int. Mask - Write one to enable Int.
10	RX2_THRESH_PEND_MASK	R/W	0h	RX Channel 2 Threshold Pending Int. Mask - Write one to enable Int.
9	RX1_THRESH_PEND_MASK	R/W	0h	RX Channel 1 Threshold Pending Int. Mask - Write one to enable Int.
8	RX0_THRESH_PEND_MASK	R/W	0h	RX Channel 0 Threshold Pending Int. Mask - Write one to enable Int.
7	RX7_PEND_MASK	R/W	0h	RX Channel 7 Pending Int. Mask - Write one to enable Int.
6	RX6_PEND_MASK	R/W	0h	RX Channel 6 Pending Int. Mask - Write one to enable Int.
5	RX5_PEND_MASK	R/W	0h	RX Channel 5 Pending Int. Mask - Write one to enable Int.
4	RX4_PEND_MASK	R/W	0h	RX Channel 4 Pending Int. Mask - Write one to enable Int.
3	RX3_PEND_MASK	R/W	0h	RX Channel 3 Pending Int. Mask - Write one to enable Int.
2	RX2_PEND_MASK	R/W	0h	RX Channel 2 Pending Int. Mask - Write one to enable Int.
1	RX1_PEND_MASK	R/W	0h	RX Channel 1 Pending Int. Mask - Write one to enable Int.
0	RX0_PEND_MASK	R/W	0h	RX Channel 0 Pending Int. Mask - Write one to enable Int.

15.5.2.29 CPSW_RX_INTMASK_CLR Register (offset = ACh) [reset = 0h]

CPSW_RX_INTMASK_CLR is shown in Figure 15-52 and described in Table 15-62.

CPDMA_INT RX INTERRUPT MASK CLEAR REGISTER

Figure 15-52. CPSW_RX_INTMASK_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RX7_THRESH_PEND_MASK	RX6_THRESH_PEND_MASK	RX5_THRESH_PEND_MASK	RX4_THRESH_PEND_MASK	RX3_THRESH_PEND_MASK	RX2_THRESH_PEND_MASK	RX1_THRESH_PEND_MASK	RX0_THRESH_PEND_MASK
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RX7_PEND_MASK	RX6_PEND_MASK	RX5_PEND_MASK	RX4_PEND_MASK	RX3_PEND_MASK	RX2_PEND_MASK	RX1_PEND_MASK	RX0_PEND_MASK
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-62. CPSW_RX_INTMASK_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	RX7_THRESH_PEND_MASK	R/W	0h	RX Channel 7 Threshold Pending Int. Mask - Write one to disable Int.
14	RX6_THRESH_PEND_MASK	R/W	0h	RX Channel 6 Threshold Pending Int. Mask - Write one to disable Int.
13	RX5_THRESH_PEND_MASK	R/W	0h	RX Channel 5 Threshold Pending Int. Mask - Write one to disable Int.
12	RX4_THRESH_PEND_MASK	R/W	0h	RX Channel 4 Threshold Pending Int. Mask - Write one to disable Int.
11	RX3_THRESH_PEND_MASK	R/W	0h	RX Channel 3 Threshold Pending Int. Mask - Write one to disable Int.
10	RX2_THRESH_PEND_MASK	R/W	0h	RX Channel 2 Threshold Pending Int. Mask - Write one to disable Int.
9	RX1_THRESH_PEND_MASK	R/W	0h	RX Channel 1 Threshold Pending Int. Mask - Write one to disable Int.
8	RX0_THRESH_PEND_MASK	R/W	0h	RX Channel 0 Threshold Pending Int. Mask - Write one to disable Int.
7	RX7_PEND_MASK	R/W	0h	RX Channel 7 Pending Int. Mask - Write one to disable Int.
6	RX6_PEND_MASK	R/W	0h	RX Channel 6 Pending Int. Mask - Write one to disable Int.
5	RX5_PEND_MASK	R/W	0h	RX Channel 5 Pending Int. Mask - Write one to disable Int.
4	RX4_PEND_MASK	R/W	0h	RX Channel 4 Pending Int. Mask - Write one to disable Int.
3	RX3_PEND_MASK	R/W	0h	RX Channel 3 Pending Int. Mask - Write one to disable Int.
2	RX2_PEND_MASK	R/W	0h	RX Channel 2 Pending Int. Mask - Write one to disable Int.
1	RX1_PEND_MASK	R/W	0h	RX Channel 1 Pending Int. Mask - Write one to disable Int.
0	RX0_PEND_MASK	R/W	0h	RX Channel 0 Pending Int. Mask - Write one to disable Int.

15.5.2.30 CPSW_DMA_INTSTAT_RAW Register (offset = B0h) [reset = 0h]

CPSW_DMA_INTSTAT_RAW is shown in Figure 15-53 and described in Table 15-63.

CPDMA_INT DMA INTERRUPT STATUS REGISTER (RAW VALUE)

Figure 15-53. CPSW_DMA_INTSTAT_RAW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>HOST_PEND</td><td>STAT_PEND</td></tr>
<tr><td colspan="6">R-0h</td><td>R-0h</td><td>R-0h</td></tr>
</table>

Table 15-63. CPSW_DMA_INTSTAT_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	HOST_PEND	R	0h	Host Pending Interrupt - raw int read (before mask).
0	STAT_PEND	R	0h	Statistics Pending Interrupt - raw int read (before mask).

15.5.2.31 CPSW_DMA_INTSTAT_MASKED Register (offset = B4h) [reset = 0h]

CPSW_DMA_INTSTAT_MASKED is shown in Figure 15-54 and described in Table 15-64.

CPDMA_INT DMA INTERRUPT STATUS REGISTER (MASKED VALUE)

Figure 15-54. CPSW_DMA_INTSTAT_MASKED Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>HOST_PEND</td><td>STAT_PEND</td></tr>
<tr><td colspan="6">R-0h</td><td>R-0h</td><td>R-0h</td></tr>
</table>

Table 15-64. CPSW_DMA_INTSTAT_MASKED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	HOST_PEND	R	0h	Host Pending Interrupt - masked interrupt read.
0	STAT_PEND	R	0h	Statistics Pending Interrupt - masked interrupt read.

15.5.2.32 CPSW_DMA_INTMASK_SET Register (offset = B8h) [reset = 0h]

CPSW_DMA_INTMASK_SET is shown in Figure 15-55 and described in Table 15-65.

CPDMA_INT DMA INTERRUPT MASK SET REGISTER

Figure 15-55. CPSW_DMA_INTMASK_SET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED						HOST_ERR_INT_MASK	STAT_INT_MASK
R-0h						W-0h	R-0h

Table 15-65. CPSW_DMA_INTMASK_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	HOST_ERR_INT_MASK	W	0h	Host Error Interrupt Mask - Write one to enable interrupt.
0	STAT_INT_MASK	R	0h	Statistics Interrupt Mask - Write one to enable interrupt.

15.5.2.33 CPSW_DMA_INTMASK_CLR Register (offset = BCh) [reset = 0h]

CPSW_DMA_INTMASK_CLR is shown in Figure 15-56 and described in Table 15-66.

CPDMA_INT DMA INTERRUPT MASK CLEAR REGISTER

Figure 15-56. CPSW_DMA_INTMASK_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						HOST_ERR_INT_MASK	STAT_INT_MASK
R-0h						R/W-0h	R/W-0h

Table 15-66. CPSW_DMA_INTMASK_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	HOST_ERR_INT_MASK	R/W	0h	Host Error Interrupt Mask - Write one to disable interrupt.
0	STAT_INT_MASK	R/W	0h	Statistics Interrupt Mask - Write one to disable interrupt.

15.5.2.34 CPSW_RX0_PENDTHRESH Register (offset = C0h) [reset = 0h]

CPSW_RX0_PENDTHRESH is shown in Figure 15-57 and described in Table 15-67.

CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 0

Figure 15-57. CPSW_RX0_PENDTHRESH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED								RX_PENDTHRESH							
R-0h								R/W-0h							

Table 15-67. CPSW_RX0_PENDTHRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	RX_PENDTHRESH	R/W	0h	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).

15.5.2.35 CPSW_RX1_PENDTHRESH Register (offset = C4h) [reset = 0h]

CPSW_RX1_PENDTHRESH is shown in Figure 15-58 and described in Table 15-68.

CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 1

Figure 15-58. CPSW_RX1_PENDTHRESH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RX_PENDTHRESH							
R-0h								R/W-0h							

Table 15-68. CPSW_RX1_PENDTHRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	RX_PENDTHRESH	R/W	0h	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).

15.5.2.36 CPSW_RX2_PENDTHRESH Register (offset = C8h) [reset = 0h]

CPSW_RX2_PENDTHRESH is shown in Figure 15-59 and described in Table 15-69.

CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 2

Figure 15-59. CPSW_RX2_PENDTHRESH Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">RX_PENDTHRESH</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 15-69. CPSW_RX2_PENDTHRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	RX_PENDTHRESH	R/W	0h	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).

15.5.2.37 CPSW_RX3_PENDTHRESH Register (offset = CCh) [reset = 0h]

CPSW_RX3_PENDTHRESH is shown in Figure 15-60 and described in Table 15-70.

CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 3

Figure 15-60. CPSW_RX3_PENDTHRESH Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">RX_PENDTHRESH</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 15-70. CPSW_RX3_PENDTHRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	RX_PENDTHRESH	R/W	0h	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).

15.5.2.38 CPSW_RX4_PENDTHRESH Register (offset = D0h) [reset = 0h]

CPSW_RX4_PENDTHRESH is shown in Figure 15-61 and described in Table 15-71.

CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 4

Figure 15-61. CPSW_RX4_PENDTHRESH Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">RX_PENDTHRESH</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 15-71. CPSW_RX4_PENDTHRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	RX_PENDTHRESH	R/W	0h	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).

15.5.2.39 CPSW_RX5_PENDTHRESH Register (offset = D4h) [reset = 0h]

CPSW_RX5_PENDTHRESH is shown in Figure 15-62 and described in Table 15-72.

CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 5

Figure 15-62. CPSW_RX5_PENDTHRESH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED								RX_PENDTHRESH							
R-0h								R/W-0h							

Table 15-72. CPSW_RX5_PENDTHRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	RX_PENDTHRESH	R/W	0h	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).

15.5.2.40 CPSW_RX6_PENDTHRESH Register (offset = D8h) [reset = 0h]

CPSW_RX6_PENDTHRESH is shown in Figure 15-63 and described in Table 15-73.

CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 6

Figure 15-63. CPSW_RX6_PENDTHRESH Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">RX_PENDTHRESH</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 15-73. CPSW_RX6_PENDTHRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	RX_PENDTHRESH	R/W	0h	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).

15.5.2.41 CPSW_RX7_PENDTHRESH Register (offset = DCh) [reset = 0h]

CPSW_RX7_PENDTHRESH is shown in Figure 15-64 and described in Table 15-74.

CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 7

Figure 15-64. CPSW_RX7_PENDTHRESH Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">RX_PENDTHRESH</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 15-74. CPSW_RX7_PENDTHRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	RX_PENDTHRESH	R/W	0h	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).

15.5.2.42 CPSW_RX0_FREEBUFFER Register (offset = E0h) [reset = 0h]

CPSW_RX0_FREEBUFFER is shown in Figure 15-65 and described in Table 15-75.

CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 0

Figure 15-65. CPSW_RX0_FREEBUFFER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">RX_FREEBUFFER</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">W-0h</td></tr>
</table>

Table 15-75. CPSW_RX0_FREEBUFFER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	RX_FREEBUFFER	W	0h	Rx Free Buffer Count - This field contains the count of free buffers available. The rx_pendthresh value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.

15.5.2.43 CPSW_RX1_FREEBUFFER Register (offset = E4h) [reset = 0h]

CPSW_RX1_FREEBUFFER is shown in Figure 15-66 and described in Table 15-76.

CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 1

Figure 15-66. CPSW_RX1_FREEBUFFER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">RX_FREEBUFFER</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">W-0h</td></tr>
</table>

Table 15-76. CPSW_RX1_FREEBUFFER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	RX_FREEBUFFER	W	0h	Rx Free Buffer Count - This field contains the count of free buffers available. The rx_pendthresh value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.

15.5.2.44 CPSW_RX2_FREEBUFFER Register (offset = E8h) [reset = 0h]

CPSW_RX2_FREEBUFFER is shown in Figure 15-67 and described in Table 15-77.

CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 2

Figure 15-67. CPSW_RX2_FREEBUFFER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">RX_FREEBUFFER</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">W-0h</td></tr>
</table>

Table 15-77. CPSW_RX2_FREEBUFFER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	RX_FREEBUFFER	W	0h	Rx Free Buffer Count - This field contains the count of free buffers available. The rx_pendthresh value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.

15.5.2.45 CPSW_RX3_FREEBUFFER Register (offset = ECh) [reset = 0h]

CPSW_RX3_FREEBUFFER is shown in Figure 15-68 and described in Table 15-78.

CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 3

Figure 15-68. CPSW_RX3_FREEBUFFER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">RX_FREEBUFFER</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">W-0h</td></tr>
</table>

Table 15-78. CPSW_RX3_FREEBUFFER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	RX_FREEBUFFER	W	0h	Rx Free Buffer Count - This field contains the count of free buffers available. The rx_pendthresh value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.

15.5.2.46 CPSW_RX4_FREEBUFFER Register (offset = F0h) [reset = 0h]

CPSW_RX4_FREEBUFFER is shown in Figure 15-69 and described in Table 15-79.

CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 4

Figure 15-69. CPSW_RX4_FREEBUFFER Register

31-16	15-0
RESERVED	RX_FREEBUFFER
R-0h	W-0h

Table 15-79. CPSW_RX4_FREEBUFFER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	RX_FREEBUFFER	W	0h	Rx Free Buffer Count - This field contains the count of free buffers available. The rx_pendthresh value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.

15.5.2.47 CPSW_RX5_FREEBUFFER Register (offset = F4h) [reset = 0h]

CPSW_RX5_FREEBUFFER is shown in Figure 15-70 and described in Table 15-80.

CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 5

Figure 15-70. CPSW_RX5_FREEBUFFER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">RX_FREEBUFFER</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">W-0h</td></tr>
</table>

Table 15-80. CPSW_RX5_FREEBUFFER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	RX_FREEBUFFER	W	0h	Rx Free Buffer Count - This field contains the count of free buffers available. The rx_pendthresh value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.

15.5.2.48 CPSW_RX6_FREEBUFFER Register (offset = F8h) [reset = 0h]

CPSW_RX6_FREEBUFFER is shown in Figure 15-71 and described in Table 15-81.

CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 6

Figure 15-71. CPSW_RX6_FREEBUFFER Register

31-16	15-0
RESERVED	RX_FREEBUFFER
R-0h	W-0h

Table 15-81. CPSW_RX6_FREEBUFFER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	RX_FREEBUFFER	W	0h	Rx Free Buffer Count - This field contains the count of free buffers available. The rx_pendthresh value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.

15.5.2.49 CPSW_RX7_FREEBUFFER Register (offset = FCh) [reset = 0h]

CPSW_RX7_FREEBUFFER is shown in Figure 15-72 and described in Table 15-82.

CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 7

Figure 15-72. CPSW_RX7_FREEBUFFER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">RX_FREEBUFFER</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">W-0h</td></tr>
</table>

Table 15-82. CPSW_RX7_FREEBUFFER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	RX_FREEBUFFER	W	0h	Rx Free Buffer Count - This field contains the count of free buffers available. The rx_pendthresh value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.

15.5.3 CPSW_CPTS Registers

Table 15-83 lists the memory-mapped registers for the CPSW_CPTS. All register offset addresses not listed in Table 15-83 should be considered as reserved locations and the register contents should not be modified.

Table 15-83. CPSW_CPTS Registers

Offset	Acronym	Register Name	Section
0h	CPSW_CPTS_IDVER	IDENTIFICATION AND VERSION REGISTER	Section 15.5.3.1
4h	CPSW_CPTS_CTRL	TIME SYNC CONTROL REGISTER	Section 15.5.3.2
8h	CPSW_RFTCLK_SEL	RFTCLK SELECT REGISTER	Section 15.5.3.3
Ch	CPSW_CPTS_PUSH	TIME STAMP EVENT PUSH REGISTER	Section 15.5.3.4
10h	CPSW_CPTS_LOAD_VAL	TIME STAMP LOAD VALUE REGISTER	Section 15.5.3.5
14h	CPSW_CPTS_LOAD_EN	TIME STAMP LOAD ENABLE REGISTER	Section 15.5.3.6
18h	CPSW_CPTS_COMP_VAL	TIME STAMP COMPARISON VALUE REGISTER	Section 15.5.3.7
1Ch	CPSW_CPTS_COMP_LENGTH	TIME STAMP COMPARISON LENGTH REGISTER	Section 15.5.3.8
20h	CPSW_CPTS_INTSTAT_RAW	TIME SYNC INTERRUPT STATUS RAW REGISTER	Section 15.5.3.9
24h	CPSW_CPTS_INTSTAT_MASKED	TIME SYNC INTERRUPT STATUS MASKED REGISTER	Section 15.5.3.10
28h	CPSW_CPTS_INT_EN	TIME SYNC INTERRUPT ENABLE REGISTER	Section 15.5.3.11
30h	CPSW_CPTS_EVT_POP	EVENT INTERRUPT POP REGISTER	Section 15.5.3.12
34h	CPSW_CPTS_EVT_LOW	LOWER 32-BITS OF THE EVENT VALUE	Section 15.5.3.13
38h	CPSW_CPTS_EVT_MID	MIDDLE 32-BITS OF THE EVENT VALUE	Section 15.5.3.14
3Ch	CPSW_CPTS_EVT_HIGH	UPPER 32-BITS OF THE EVENT VALUE	Section 15.5.3.15

15.5.3.1 CPSW_CPTS_IDVER Register (offset = 0h) [reset = 4E8A0101h]

CPSW_CPTS_IDVER is shown in Figure 15-73 and described in Table 15-84.

IDENTIFICATION AND VERSION REGISTER

Figure 15-73. CPSW_CPTS_IDVER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TX_IDENT															
R-4E8Ah															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTL_VER					MAJOR_VER			MINOR_VER							
R-0h					R-1h			R-1h							

Table 15-84. CPSW_CPTS_IDVER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TX_IDENT	R	4E8Ah	TX Identification Value
15-11	RTL_VER	R	0h	RTL Version Value
10-8	MAJOR_VER	R	1h	Major Version Value
7-0	MINOR_VER	R	1h	Minor Version Value

15.5.3.2 CPSW_CPTS_CTRL Register (offset = 4h) [reset = 0h]

CPSW_CPTS_CTRL is shown in Figure 15-74 and described in Table 15-85.

TIME SYNC CONTROL REGISTER

Figure 15-74. CPSW_CPTS_CTRL Register

31	30	29	28	27	26	25	24
TS_SYNC_SEL				RESERVED			
R/W-0h				R-0h			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED				HW4_TS_PUSH_EN	HW3_TS_PUSH_EN	HW2_TS_PUSH_EN	HW1_TS_PUSH_EN
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					TS_COMP_POLARITY	INT_TEST	CPTS_EN
R-0h					R/W-1h	R/W-0h	R/W-0h

Table 15-85. CPSW_CPTS_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	TS_SYNC_SEL	R/W	0h	TS_SYNC output timestamp counter bit select 0000 - TS_SYNC disabled 0001 to 1111 - TS_SYNC is timestamp counter bits 31 (1111) down to 17 (0001)
27-12	RESERVED	R	0h	
11	HW4_TS_PUSH_EN	R/W	0h	Hardware push 4 enable
10	HW3_TS_PUSH_EN	R/W	0h	Hardware push 3 enable
9	HW2_TS_PUSH_EN	R/W	0h	Hardware push 2 enable
8	HW1_TS_PUSH_EN	R/W	0h	Hardware push 1 enable
7-3	RESERVED	R	0h	
2	TS_COMP_POLARITY	R/W	1h	TS_COMP Polarity 0h (R/W) = TS_COMP is asserted low 1h (R/W) = TS_COMP is asserted high
1	INT_TEST	R/W	0h	Interrupt Test - When set, this bit allows the raw interrupt to be written to facilitate interrupt test.
0	CPTS_EN	R/W	0h	Time Sync Enable - When disabled (cleared to zero), the RCLK domain is held in reset. 0h (R/W) = Time Sync Disabled 1h (R/W) = Time Sync Enabled

15.5.3.3 CPSW_RFTCLK_SEL Register (offset = 8h) [reset = 0h]

CPSW_RFTCLK_SEL is shown in Figure 15-75 and described in Table 15-86.

RFTCLK SELECT REGISTER

Figure 15-75. CPSW_RFTCLK_SEL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="11">RESERVED</td><td colspan="5">RFTCLK_SEL</td></tr>
<tr><td colspan="11">R-0h</td><td colspan="5">R/W-0h</td></tr>
</table>

Table 15-86. CPSW_RFTCLK_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4-0	RFTCLK_SEL	R/W	0h	Reference Clock Select - This signal is used to control an external multiplexor that selects one of up to 32 clocks for time sync reference (RFTCLK). This RFTCLK_SEL value can be written only when the CPTS_EN bit is cleared to zero in the CPTS_CONTROL register.

15.5.3.4 CPSW_CPTS_PUSH Register (offset = Ch) [reset = 0h]

CPSW_CPTS_PUSH is shown in Figure 15-76 and described in Table 15-87.

TIME STAMP EVENT PUSH REGISTER

Figure 15-76. CPSW_CPTS_PUSH Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TS_PUSH
R-0h							W-0h

Table 15-87. CPSW_CPTS_PUSH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TS_PUSH	W	0h	Time stamp event push - When a logic high is written to this bit a time stamp event is pushed onto the event FIFO. The time stamp value is the time of the write of this register, not the time of the event read. The time stamp value can then be read on interrupt via the event registers. Software should not push a second time stamp event onto the event FIFO until the first time stamp value has been read from the event FIFO (there should be only one time stamp event in the event FIFO at any given time). This bit is write only and always reads zero.

15.5.3.5 CPSW_CPTS_LOAD_VAL Register (offset = 10h) [reset = 0h]

CPSW_CPTS_LOAD_VAL is shown in Figure 15-77 and described in Table 15-88.

TIME STAMP LOAD VALUE REGISTER

Figure 15-77. CPSW_CPTS_LOAD_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS_LOAD_VAL																															
R/W-0h																															

Table 15-88. CPSW_CPTS_LOAD_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TS_LOAD_VAL	R/W	0h	Time Stamp Load Value - Writing the ts_load_en bit causes the value contained in this register to be written into the time stamp. The time stamp value is read by initiating a time stamp push event, not by reading this register. When reading this register, the value read is not the time stamp, but is the value that was last written to this register.

15.5.3.6 CPSW_CPTS_LOAD_EN Register (offset = 14h) [reset = 0h]

CPSW_CPTS_LOAD_EN is shown in Figure 15-78 and described in Table 15-89.

TIME STAMP LOAD ENABLE REGISTER

Figure 15-78. CPSW_CPTS_LOAD_EN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TS_LOAD_EN
R-0h							W-0h

Table 15-89. CPSW_CPTS_LOAD_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TS_LOAD_EN	W	0h	Time Stamp Load - Writing a one to this bit enables the time stamp value to be written via the ts_load_val [31:0] register. This bit is write only and is cleared by the hardware after one clock..

15.5.3.7 CPSW_CPTS_COMP_VAL Register (offset = 18h) [reset = 0h]

CPSW_CPTS_COMP_VAL is shown in Figure 15-79 and described in Table 15-90.

TIME STAMP COMPARISON VALUE REGISTER

Figure 15-79. CPSW_CPTS_COMP_VAL Register

31–0
TS_COMP_VAL
R/W-0h

Table 15-90. CPSW_CPTS_COMP_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TS_COMP_VAL	R/W	0h	Time Stamp Comparison Value Writing a non-zero value to the TS_Comp_Length[15:0] register causes a pulse of TS_Comp_Length RCLK periods on the TS_COMP output and a comparison event when the time_stamp counter value is equivalent to ts_comp_val.

15.5.3.8 CPSW_CPTS_COMP_LENGTH Register (offset = 1Ch) [reset = 0h]

CPSW_CPTS_COMP_LENGTH is shown in Figure 15-80 and described in Table 15-91.

TIME STAMP COMPARISON LENGTH REGISTER

Figure 15-80. CPSW_CPTS_COMP_LENGTH Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">TS_COMP_LENGTH</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">R/W-0h</td></tr>
</table>

Table 15-91. CPSW_CPTS_COMP_LENGTH Register Field Descriptions

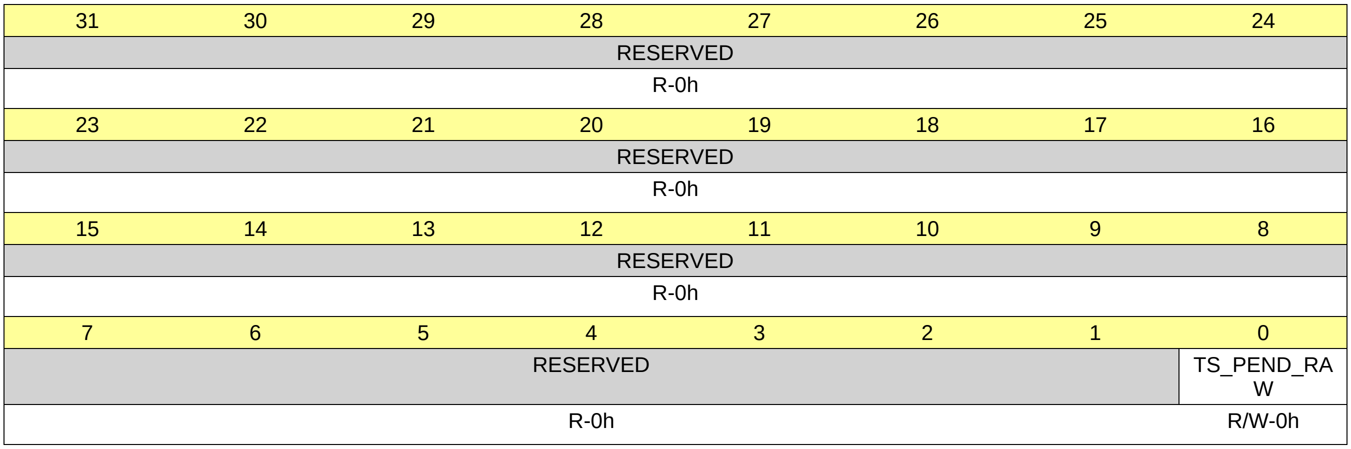
Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	TS_COMP_LENGTH	R/W	0h	Time Stamp Comparison Length Writing a non-zero value to this field enables the time stamp comparison event and output. This value should be zero when the TS_Comp_Val register is written.

15.5.3.9 CPSW_CPTS_INTSTAT_RAW Register (offset = 20h) [reset = 0h]

CPSW_CPTS_INTSTAT_RAW is shown in Figure 15-81 and described in Table 15-92.

TIME SYNC INTERRUPT STATUS RAW REGISTER

Figure 15-81. CPSW_CPTS_INTSTAT_RAW Register



31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TS_PEND_RAW
R-0h							R/W-0h

Table 15-92. CPSW_CPTS_INTSTAT_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TS_PEND_RAW	R/W	0h	TS_PEND_RAW int read (before enable). Writable when int_test = 1 A one in this bit indicates that there is one or more events in the event FIFO.

15.5.3.10 CPSW_CPTS_INTSTAT_MASKED Register (offset = 24h) [reset = 0h]

CPSW_CPTS_INTSTAT_MASKED is shown in Figure 15-82 and described in Table 15-93.

TIME SYNC INTERRUPT STATUS MASKED REGISTER

Figure 15-82. CPSW_CPTS_INTSTAT_MASKED Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							TS_PEND
R-0h							R-0h

Table 15-93. CPSW_CPTS_INTSTAT_MASKED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TS_PEND	R	0h	TS_PEND masked interrupt read (after enable).

15.5.3.11 CPSW_CPTS_INT_EN Register (offset = 28h) [reset = 0h]

CPSW_CPTS_INT_EN is shown in Figure 15-83 and described in Table 15-94.

TIME SYNC INTERRUPT ENABLE REGISTER

Figure 15-83. CPSW_CPTS_INT_EN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							TS_PEND_EN
R-0h							R/W-0h

Table 15-94. CPSW_CPTS_INT_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TS_PEND_EN	R/W	0h	TS_PEND masked interrupt enable.

15.5.3.12 CPSW_CPTS_EVT_POP Register (offset = 30h) [reset = 0h]

CPSW_CPTS_EVT_POP is shown in Figure 15-84 and described in Table 15-95.

EVENT INTERRUPT POP REGISTER

Figure 15-84. CPSW_CPTS_EVT_POP Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							EVT_POP
R-0h							W-0h

Table 15-95. CPSW_CPTS_EVT_POP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	EVT_POP	W	0h	Event Pop - When a logic high is written to this bit an event is popped off the event FIFO. The event FIFO pop occurs as part of the interrupt process after the event has been read in the Event_Low and Event_High registers. Popping an event discards the event and causes the next event, if any, to be moved to the top of the FIFO ready to be read by software on interrupt.

15.5.3.13 CPSW_CPTS_EVT_LOW Register (offset = 34h) [reset = 0h]

CPSW_CPTS_EVT_LOW is shown in Figure 15-85 and described in Table 15-96.

LOWER 32-BITS OF THE EVENT VALUE

Figure 15-85. CPSW_CPTS_EVT_LOW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">TIME_STAMP</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 15-96. CPSW_CPTS_EVT_LOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TIME_STAMP	R	0h	Time Stamp - The timestamp is valid for transmit, receive, and time stamp push event types. The timestamp value is not valid for counter roll event types.

15.5.3.14 CPSW_CPTS_EVT_MID Register (offset = 38h) [reset = 0h]

CPSW_CPTS_EVT_MID is shown in Figure 15-86 and described in Table 15-97.

MIDDLE 32-BITS OF THE EVENT VALUE

Figure 15-86. CPSW_CPTS_EVT_MID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED			PORT_NUMBER					EVT_TYPE				MESSAGE_TYPE			
R-0h			R-0h					R-0h				R-0h			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEQUENCE_ID															
R-0h															

Table 15-97. CPSW_CPTS_EVT_MID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	PORT_NUMBER	R	0h	Port Number - indicates the port number of an ethernet event or the hardware push pin number (1 to 4).
23-20	EVT_TYPE	R	0h	Time Sync Event Type 0h (R/W) = Time Stamp Push Event 1h (R/W) = Time Stamp Rollover Event 2h (R/W) = Time Stamp Half Rollover Event 3h (R/W) = Hardware Time Stamp Push Event 4h (R/W) = Ethernet Receive Event 5h (R/W) = Ethernet Transmit Event
19-16	MESSAGE_TYPE	R	0h	Message type - The message type value that was contained in an ethernet transmit or receive time sync packet. This field is valid only for ethernet transmit or receive events.
15-0	SEQUENCE_ID	R	0h	Sequence ID - The 16-bit sequence id is the value that was contained in an ethernet transmit or receivetime sync packet. This field is valid only for ethernet transmit or receive events.

15.5.3.15 CPSW_CPTS_EVT_HIGH Register (offset = 3Ch) [reset = 0h]

CPSW_CPTS_EVT_HIGH is shown in Figure 15-87 and described in Table 15-98.

UPPER 32-BITS OF THE EVENT VALUE

Figure 15-87. CPSW_CPTS_EVT_HIGH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								DOMAIN							
R-0h																								R-0h							

Table 15-98. CPSW_CPTS_EVT_HIGH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	DOMAIN	R	0h	Domain The 8-bit domain is the value that was contained in an Ethernet transmit or receive time sync packet. This field is valid only for Ethernet transmit or receive events.

15.5.4 CPSW_STATS Registers

For a full description of the CPSW_STATS registers, see Section 15.3.2.20, *CPSW_3G Network Statistics*. The registers are summarized in Table 15-99.

Table 15-99. CPSW_STATS REGISTERS

Offset	Acronym	Register Name	Section
00h		Good Rx Frames	Section 15.3.2.20.1.1
04h		Broadcast Rx Frames	Section 15.3.2.20.1.2
08h		Multicast Rx Frames	Section 15.3.2.20.1.3
0Ch		Pause Rx Frames	Section 15.3.2.20.1.4
10h		Rx CRC Errors	Section 15.3.2.20.1.5
14h		Rx Align/Code Errors	Section 15.3.2.20.1.6
18h		Oversize Rx Frames	Section 15.3.2.20.1.7
1Ch		Rx Jabbers	Section 15.3.2.20.1.8
20h		Undersize (Short) Rx Frames	Section 15.3.2.20.1.9
24h		Rx Fragments	Section 15.3.2.20.1.10
30h		Rx Octets	Section 15.3.2.20.1.14
34h		Good Tx Frames	Section 15.3.2.20.2.1
38h		Broadcast Tx Frames	Section 15.3.2.20.2.2
3Ch		Multicast Tx Frames	Section 15.3.2.20.2.3
40h		Pause Tx Frames	Section 15.3.2.20.2.4
44h		Deferred Tx Frames	Section 15.3.2.20.2.11
48h		Collisions	Section 15.3.2.20.2.5
4Ch		Single Collision Tx Frames	Section 15.3.2.20.2.6
50h		Multiple Collision Tx Frames	Section 15.3.2.20.2.7
54h		Excessive Collisions	Section 15.3.2.20.2.8
58h		Late Collisions	Section 15.3.2.20.2.9
5Ch		Tx Underrun	Section 15.3.2.20.2.10
60h		Carrier Sense Errors	Section 15.3.2.20.2.12
64h		Tx Octets	Section 15.3.2.20.2.13
68h		Rx + Tx 64 Octet Frames	Section 15.3.2.20.3.1
6Ch		Rx + Tx 65–127 Octet Frames	Section 15.3.2.20.3.2
70h		Rx + Tx 128–255 Octet Frames	Section 15.3.2.20.3.3
74h		Rx + Tx 256–511 Octet Frames	Section 15.3.2.20.3.4
78h		Rx + Tx 512–1023 Octet Frames	Section 15.3.2.20.3.5
7Ch		Rx + Tx 1024_Up Octet Frames	Section 15.3.2.20.3.6
80h		Net Octets	Section 15.3.2.20.1.15
84h		Rx Start of Frame Overruns	Section 15.3.2.20.1.11
88h		Rx Middle of Frame Overruns	Section 15.3.2.20.1.12
8Ch		Rx DMA Overruns	Section 15.3.2.20.1.13

15.5.5 CPDMA_STATERAM Registers

Table 15-100 lists the memory-mapped registers for the CPDMA_STATERAM. All register offset addresses not listed in Table 15-100 should be considered as reserved locations and the register contents should not be modified.

Table 15-100. CPDMA_STATERAM REGISTERS

Offset	Acronym	Register Name	Section
0h	CPSW_STATERAM_TX0_HDP	CPDMA_STATERAM TX CHANNEL 0 HEAD DESC POINTER *	Section 15.5.5.1
4h	CPSW_STATERAM_TX1_HDP	CPDMA_STATERAM TX CHANNEL 1 HEAD DESC POINTER *	Section 15.5.5.2
8h	CPSW_STATERAM_TX2_HDP	CPDMA_STATERAM TX CHANNEL 2 HEAD DESC POINTER *	Section 15.5.5.3
Ch	CPSW_STATERAM_TX3_HDP	CPDMA_STATERAM TX CHANNEL 3 HEAD DESC POINTER *	Section 15.5.5.4
10h	CPSW_STATERAM_TX4_HDP	CPDMA_STATERAM TX CHANNEL 4 HEAD DESC POINTER *	Section 15.5.5.5
14h	CPSW_STATERAM_TX5_HDP	CPDMA_STATERAM TX CHANNEL 5 HEAD DESC POINTER *	Section 15.5.5.6
18h	CPSW_STATERAM_TX6_HDP	CPDMA_STATERAM TX CHANNEL 6 HEAD DESC POINTER *	Section 15.5.5.7
1Ch	CPSW_STATERAM_TX7_HDP	CPDMA_STATERAM TX CHANNEL 7 HEAD DESC POINTER *	Section 15.5.5.8
20h	CPSW_STATERAM_RX0_HDP	CPDMA_STATERAM RX 0 CHANNEL 0 HEAD DESC POINTER *	Section 15.5.5.9
24h	CPSW_STATERAM_RX1_HDP	CPDMA_STATERAM RX 1 CHANNEL 1 HEAD DESC POINTER *	Section 15.5.5.10
28h	CPSW_STATERAM_RX2_HDP	CPDMA_STATERAM RX 2 CHANNEL 2 HEAD DESC POINTER *	Section 15.5.5.11
2Ch	CPSW_STATERAM_RX3_HDP	CPDMA_STATERAM RX 3 CHANNEL 3 HEAD DESC POINTER *	Section 15.5.5.12
30h	CPSW_STATERAM_RX4_HDP	CPDMA_STATERAM RX 4 CHANNEL 4 HEAD DESC POINTER *	Section 15.5.5.13
34h	CPSW_STATERAM_RX5_HDP	CPDMA_STATERAM RX 5 CHANNEL 5 HEAD DESC POINTER *	Section 15.5.5.14
38h	CPSW_STATERAM_RX6_HDP	CPDMA_STATERAM RX 6 CHANNEL 6 HEAD DESC POINTER *	Section 15.5.5.15
3Ch	CPSW_STATERAM_RX7_HDP	CPDMA_STATERAM RX 7 CHANNEL 7 HEAD DESC POINTER *	Section 15.5.5.16
40h	CPSW_STATERAM_TX0_CP	CPDMA_STATERAM TX CHANNEL 0 COMPLETION POINTER REGISTER	Section 15.5.5.17
44h	CPSW_STATERAM_TX1_CP	CPDMA_STATERAM TX CHANNEL 1 COMPLETION POINTER REGISTER *	Section 15.5.5.18
48h	CPSW_STATERAM_TX2_CP	CPDMA_STATERAM TX CHANNEL 2 COMPLETION POINTER REGISTER *	Section 15.5.5.19
4Ch	CPSW_STATERAM_TX3_CP	CPDMA_STATERAM TX CHANNEL 3 COMPLETION POINTER REGISTER *	Section 15.5.5.20
50h	CPSW_STATERAM_TX4_CP	CPDMA_STATERAM TX CHANNEL 4 COMPLETION POINTER REGISTER *	Section 15.5.5.21
54h	CPSW_STATERAM_TX5_CP	CPDMA_STATERAM TX CHANNEL 5 COMPLETION POINTER REGISTER *	Section 15.5.5.22
58h	CPSW_STATERAM_TX6_CP	CPDMA_STATERAM TX CHANNEL 6 COMPLETION POINTER REGISTER *	Section 15.5.5.23
5Ch	CPSW_STATERAM_TX7_CP	CPDMA_STATERAM TX CHANNEL 7 COMPLETION POINTER REGISTER *	Section 15.5.5.24
60h	CPSW_STATERAM_RX0_CP	CPDMA_STATERAM RX CHANNEL 0 COMPLETION POINTER REGISTER *	Section 15.5.5.25
64h	CPSW_STATERAM_RX1_CP	CPDMA_STATERAM RX CHANNEL 1 COMPLETION POINTER REGISTER *	Section 15.5.5.26
68h	CPSW_STATERAM_RX2_CP	CPDMA_STATERAM RX CHANNEL 2 COMPLETION POINTER REGISTER *	Section 15.5.5.27
6Ch	CPSW_STATERAM_RX3_CP	CPDMA_STATERAM RX CHANNEL 3 COMPLETION POINTER REGISTER *	Section 15.5.5.28

Table 15-100. CPDMA_STATERAM REGISTERS (continued)

Offset	Acronym	Register Name	Section
70h	CPSW_STATERAM_RX4_CP	CPDMA_STATERAM RX CHANNEL 4 COMPLETION POINTER REGISTER *	Section 15.5.5.29
74h	CPSW_STATERAM_RX5_CP	CPDMA_STATERAM RX CHANNEL 5 COMPLETION POINTER REGISTER *	Section 15.5.5.30
78h	CPSW_STATERAM_RX6_CP	CPDMA_STATERAM RX CHANNEL 6 COMPLETION POINTER REGISTER *	Section 15.5.5.31
7Ch	CPSW_STATERAM_RX7_CP	CPDMA_STATERAM RX CHANNEL 7 COMPLETION POINTER REGISTER *	Section 15.5.5.32

15.5.5.1 CPSW_STATERAM_TX0_HDP Register (offset = 0h) [reset = 0h]

CPSW_STATERAM_TX0_HDP is shown in Figure 15-88 and described in Table 15-101.

CPDMA_STATERAM TX CHANNEL 0 HEAD DESC POINTER *

Figure 15-88. CPSW_STATERAM_TX0_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															
R/W-0h																															

Table 15-101. CPSW_STATERAM_TX0_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_HDP	R/W	0h	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

15.5.5.2 CPSW_STATERAM_TX1_HDP Register (offset = 4h) [reset = 0h]

CPSW_STATERAM_TX1_HDP is shown in Figure 15-89 and described in Table 15-102.

CPDMA_STATERAM TX CHANNEL 1 HEAD DESC POINTER *

Figure 15-89. CPSW_STATERAM_TX1_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															
R/W-0h																															

Table 15-102. CPSW_STATERAM_TX1_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_HDP	R/W	0h	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

15.5.5.3 CPSW_STATERAM_TX2_HDP Register (offset = 8h) [reset = 0h]

CPSW_STATERAM_TX2_HDP is shown in Figure 15-90 and described in Table 15-103.

CPDMA_STATERAM TX CHANNEL 2 HEAD DESC POINTER *

Figure 15-90. CPSW_STATERAM_TX2_HDP Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">TX_HDP</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 15-103. CPSW_STATERAM_TX2_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_HDP	R/W	0h	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

15.5.5.4 CPSW_STATERAM_TX3_HDP Register (offset = Ch) [reset = 0h]

CPSW_STATERAM_TX3_HDP is shown in Figure 15-91 and described in Table 15-104.

CPDMA_STATERAM TX CHANNEL 3 HEAD DESC POINTER *

Figure 15-91. CPSW_STATERAM_TX3_HDP Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

TX_HDP

R/W-0h

Table 15-104. CPSW_STATERAM_TX3_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_HDP	R/W	0h	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

15.5.5.5 CPSW_STATERAM_TX4_HDP Register (offset = 10h) [reset = 0h]

CPSW_STATERAM_TX4_HDP is shown in Figure 15-92 and described in Table 15-105.

CPDMA_STATERAM TX CHANNEL 4 HEAD DESC POINTER *

Figure 15-92. CPSW_STATERAM_TX4_HDP Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">TX_HDP</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 15-105. CPSW_STATERAM_TX4_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_HDP	R/W	0h	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

15.5.5.6 CPSW_STATERAM_TX5_HDP Register (offset = 14h) [reset = 0h]

CPSW_STATERAM_TX5_HDP is shown in Figure 15-93 and described in Table 15-106.

CPDMA_STATERAM TX CHANNEL 5 HEAD DESC POINTER *

Figure 15-93. CPSW_STATERAM_TX5_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															
R/W-0h																															

Table 15-106. CPSW_STATERAM_TX5_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_HDP	R/W	0h	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

15.5.5.7 CPSW_STATERAM_TX6_HDP Register (offset = 18h) [reset = 0h]

CPSW_STATERAM_TX6_HDP is shown in Figure 15-94 and described in Table 15-107.

CPDMA_STATERAM TX CHANNEL 6 HEAD DESC POINTER *

Figure 15-94. CPSW_STATERAM_TX6_HDP Register

31–0
TX_HDP
R/W-0h

Table 15-107. CPSW_STATERAM_TX6_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_HDP	R/W	0h	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

15.5.5.8 CPSW_STATERAM_TX7_HDP Register (offset = 1Ch) [reset = 0h]

CPSW_STATERAM_TX7_HDP is shown in Figure 15-95 and described in Table 15-108.

CPDMA_STATERAM TX CHANNEL 7 HEAD DESC POINTER *

Figure 15-95. CPSW_STATERAM_TX7_HDP Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

TX_HDP

R/W-0h

Table 15-108. CPSW_STATERAM_TX7_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_HDP	R/W	0h	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

15.5.5.9 CPSW_STATERAM_RX0_HDP Register (offset = 20h) [reset = 0h]

CPSW_STATERAM_RX0_HDP is shown in Figure 15-96 and described in Table 15-109.

CPDMA_STATERAM RX 0 CHANNEL 0 HEAD DESC POINTER *

Figure 15-96. CPSW_STATERAM_RX0_HDP Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">RX_HDP</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 15-109. CPSW_STATERAM_RX0_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_HDP	R/W	0h	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

15.5.5.10 CPSW_STATERAM_RX1_HDP Register (offset = 24h) [reset = 0h]

CPSW_STATERAM_RX1_HDP is shown in Figure 15-97 and described in Table 15-110.

CPDMA_STATERAM RX 1 CHANNEL 1 HEAD DESC POINTER *

Figure 15-97. CPSW_STATERAM_RX1_HDP Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">RX_HDP</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 15-110. CPSW_STATERAM_RX1_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_HDP	R/W	0h	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

15.5.5.11 CPSW_STATERAM_RX2_HDP Register (offset = 28h) [reset = 0h]

CPSW_STATERAM_RX2_HDP is shown in Figure 15-98 and described in Table 15-111.

CPDMA_STATERAM RX 2 CHANNEL 2 HEAD DESC POINTER *

Figure 15-98. CPSW_STATERAM_RX2_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															
R/W-0h																															

Table 15-111. CPSW_STATERAM_RX2_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_HDP	R/W	0h	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

15.5.5.12 CPSW_STATERAM_RX3_HDP Register (offset = 2Ch) [reset = 0h]

CPSW_STATERAM_RX3_HDP is shown in Figure 15-99 and described in Table 15-112.

CPDMA_STATERAM RX 3 CHANNEL 3 HEAD DESC POINTER *

Figure 15-99. CPSW_STATERAM_RX3_HDP Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">RX_HDP</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 15-112. CPSW_STATERAM_RX3_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_HDP	R/W	0h	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

15.5.5.13 CPSW_STATERAM_RX4_HDP Register (offset = 30h) [reset = 0h]

CPSW_STATERAM_RX4_HDP is shown in Figure 15-100 and described in Table 15-113.

CPDMA_STATERAM RX 4 CHANNEL 4 HEAD DESC POINTER *

Figure 15-100. CPSW_STATERAM_RX4_HDP Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">RX_HDP</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 15-113. CPSW_STATERAM_RX4_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_HDP	R/W	0h	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

15.5.5.14 CPSW_STATERAM_RX5_HDP Register (offset = 34h) [reset = 0h]

CPSW_STATERAM_RX5_HDP is shown in Figure 15-101 and described in Table 15-114.

CPDMA_STATERAM RX 5 CHANNEL 5 HEAD DESC POINTER *

Figure 15-101. CPSW_STATERAM_RX5_HDP Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">RX_HDP</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 15-114. CPSW_STATERAM_RX5_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_HDP	R/W	0h	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

15.5.5.15 CPSW_STATERAM_RX6_HDP Register (offset = 38h) [reset = 0h]

CPSW_STATERAM_RX6_HDP is shown in Figure 15-102 and described in Table 15-115.

CPDMA_STATERAM RX 6 CHANNEL 6 HEAD DESC POINTER *

Figure 15-102. CPSW_STATERAM_RX6_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															
R/W-0h																															

Table 15-115. CPSW_STATERAM_RX6_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_HDP	R/W	0h	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

15.5.5.16 CPSW_STATERAM_RX7_HDP Register (offset = 3Ch) [reset = 0h]

CPSW_STATERAM_RX7_HDP is shown in Figure 15-103 and described in Table 15-116.

CPDMA_STATERAM RX 7 CHANNEL 7 HEAD DESC POINTER *

Figure 15-103. CPSW_STATERAM_RX7_HDP Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">RX_HDP</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 15-116. CPSW_STATERAM_RX7_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_HDP	R/W	0h	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

15.5.5.17 CPSW_STATERAM_TX0_CP Register (offset = 40h) [reset = 0h]

CPSW_STATERAM_TX0_CP is shown in Figure 15-104 and described in Table 15-117.

CPDMA_STATERAM TX CHANNEL 0 COMPLETION POINTER REGISTER

Figure 15-104. CPSW_STATERAM_TX0_CP Register

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="32">TX_CP</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 15-117. CPSW_STATERAM_TX0_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CP	R/W	0h	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.

15.5.5.18 CPSW_STATERAM_TX1_CP Register (offset = 44h) [reset = 0h]

CPSW_STATERAM_TX1_CP is shown in Figure 15-105 and described in Table 15-118.

CPDMA_STATERAM TX CHANNEL 1 COMPLETION POINTER REGISTER *

Figure 15-105. CPSW_STATERAM_TX1_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															
R/W-0h																															

Table 15-118. CPSW_STATERAM_TX1_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CP	R/W	0h	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.

15.5.5.19 CPSW_STATERAM_TX2_CP Register (offset = 48h) [reset = 0h]

CPSW_STATERAM_TX2_CP is shown in Figure 15-106 and described in Table 15-119.

CPDMA_STATERAM TX CHANNEL 2 COMPLETION POINTER REGISTER *

Figure 15-106. CPSW_STATERAM_TX2_CP Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">TX_CP</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 15-119. CPSW_STATERAM_TX2_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CP	R/W	0h	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.

15.5.5.20 CPSW_STATERAM_TX3_CP Register (offset = 4Ch) [reset = 0h]

CPSW_STATERAM_TX3_CP is shown in Figure 15-107 and described in Table 15-120.

CPDMA_STATERAM TX CHANNEL 3 COMPLETION POINTER REGISTER *

Figure 15-107. CPSW_STATERAM_TX3_CP Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">TX_CP</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 15-120. CPSW_STATERAM_TX3_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CP	R/W	0h	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.

15.5.5.21 CPSW_STATERAM_TX4_CP Register (offset = 50h) [reset = 0h]

CPSW_STATERAM_TX4_CP is shown in Figure 15-108 and described in Table 15-121.

CPDMA_STATERAM TX CHANNEL 4 COMPLETION POINTER REGISTER *

Figure 15-108. CPSW_STATERAM_TX4_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															
R/W-0h																															

Table 15-121. CPSW_STATERAM_TX4_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CP	R/W	0h	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.

15.5.5.22 CPSW_STATERAM_TX5_CP Register (offset = 54h) [reset = 0h]

CPSW_STATERAM_TX5_CP is shown in Figure 15-109 and described in Table 15-122.

CPDMA_STATERAM TX CHANNEL 5 COMPLETION POINTER REGISTER *

Figure 15-109. CPSW_STATERAM_TX5_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															
R/W-0h																															

Table 15-122. CPSW_STATERAM_TX5_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CP	R/W	0h	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.

15.5.5.23 CPSW_STATERAM_TX6_CP Register (offset = 58h) [reset = 0h]

CPSW_STATERAM_TX6_CP is shown in Figure 15-110 and described in Table 15-123.

CPDMA_STATERAM TX CHANNEL 6 COMPLETION POINTER REGISTER *

Figure 15-110. CPSW_STATERAM_TX6_CP Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

TX_CP

R/W-0h

Table 15-123. CPSW_STATERAM_TX6_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CP	R/W	0h	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.

15.5.5.24 CPSW_STATERAM_TX7_CP Register (offset = 5Ch) [reset = 0h]

CPSW_STATERAM_TX7_CP is shown in Figure 15-111 and described in Table 15-124.

CPDMA_STATERAM TX CHANNEL 7 COMPLETION POINTER REGISTER *

Figure 15-111. CPSW_STATERAM_TX7_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															
R/W-0h																															

Table 15-124. CPSW_STATERAM_TX7_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CP	R/W	0h	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.

15.5.5.25 CPSW_STATERAM_RX0_CP Register (offset = 60h) [reset = 0h]

CPSW_STATERAM_RX0_CP is shown in Figure 15-112 and described in Table 15-125.

CPDMA_STATERAM RX CHANNEL 0 COMPLETION POINTER REGISTER *

Figure 15-112. CPSW_STATERAM_RX0_CP Register

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="32">RX_CP</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 15-125. CPSW_STATERAM_RX0_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_CP	R/W	0h	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.

15.5.5.26 CPSW_STATERAM_RX1_CP Register (offset = 64h) [reset = 0h]

CPSW_STATERAM_RX1_CP is shown in Figure 15-113 and described in Table 15-126.

CPDMA_STATERAM RX CHANNEL 1 COMPLETION POINTER REGISTER *

Figure 15-113. CPSW_STATERAM_RX1_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															
R/W-0h																															

Table 15-126. CPSW_STATERAM_RX1_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_CP	R/W	0h	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.

15.5.5.27 CPSW_STATERAM_RX2_CP Register (offset = 68h) [reset = 0h]

CPSW_STATERAM_RX2_CP is shown in Figure 15-114 and described in Table 15-127.

CPDMA_STATERAM RX CHANNEL 2 COMPLETION POINTER REGISTER *

Figure 15-114. CPSW_STATERAM_RX2_CP Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">RX_CP</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 15-127. CPSW_STATERAM_RX2_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_CP	R/W	0h	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.

15.5.5.28 CPSW_STATERAM_RX3_CP Register (offset = 6Ch) [reset = 0h]

CPSW_STATERAM_RX3_CP is shown in Figure 15-115 and described in Table 15-128.

CPDMA_STATERAM RX CHANNEL 3 COMPLETION POINTER REGISTER *

Figure 15-115. CPSW_STATERAM_RX3_CP Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
RX_CP
R/W-0h

Table 15-128. CPSW_STATERAM_RX3_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_CP	R/W	0h	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.

15.5.5.29 CPSW_STATERAM_RX4_CP Register (offset = 70h) [reset = 0h]

CPSW_STATERAM_RX4_CP is shown in Figure 15-116 and described in Table 15-129.

CPDMA_STATERAM RX CHANNEL 4 COMPLETION POINTER REGISTER *

Figure 15-116. CPSW_STATERAM_RX4_CP Register

31-0
RX_CP
R/W-0h

Table 15-129. CPSW_STATERAM_RX4_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_CP	R/W	0h	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.

15.5.5.30 CPSW_STATERAM_RX5_CP Register (offset = 74h) [reset = 0h]

CPSW_STATERAM_RX5_CP is shown in Figure 15-117 and described in Table 15-130.

CPDMA_STATERAM RX CHANNEL 5 COMPLETION POINTER REGISTER *

Figure 15-117. CPSW_STATERAM_RX5_CP Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">RX_CP</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 15-130. CPSW_STATERAM_RX5_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_CP	R/W	0h	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.

15.5.5.31 CPSW_STATERAM_RX6_CP Register (offset = 78h) [reset = 0h]

CPSW_STATERAM_RX6_CP is shown in Figure 15-118 and described in Table 15-131.

CPDMA_STATERAM RX CHANNEL 6 COMPLETION POINTER REGISTER *

Figure 15-118. CPSW_STATERAM_RX6_CP Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">RX_CP</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 15-131. CPSW_STATERAM_RX6_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_CP	R/W	0h	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.

15.5.5.32 CPSW_STATERAM_RX7_CP Register (offset = 7Ch) [reset = 0h]

CPSW_STATERAM_RX7_CP is shown in Figure 15-119 and described in Table 15-132.

CPDMA_STATERAM RX CHANNEL 7 COMPLETION POINTER REGISTER *

Figure 15-119. CPSW_STATERAM_RX7_CP Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">RX_CP</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 15-132. CPSW_STATERAM_RX7_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_CP	R/W	0h	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.

15.5.6 CPSW_PORT Registers

Table 15-133 lists the memory-mapped registers for the CPSW_PORT. All register offset addresses not listed in Table 15-133 should be considered as reserved locations and the register contents should not be modified.

Table 15-133. CPSW_PORT Registers

Offset	Acronym	Register Name	Section
0h	CPSW_PORT_P0_CTRL		Section 15.5.6.1
8h	CPSW_PORT_P0_MAX_BLKS		Section 15.5.6.2
Ch	CPSW_PORT_P0_BLK_CNT		Section 15.5.6.3
10h	CPSW_PORT_P0_TX_IN_CTL		Section 15.5.6.4
14h	CPSW_PORT_P0_VLAN		Section 15.5.6.5
18h	CPSW_PORT_P0_TX_PRI_MAP		Section 15.5.6.6
1Ch	CPSW_PORT_P0_CPDMA_TX_PRI_MAP		Section 15.5.6.7
20h	CPSW_PORT_P0_CPDMA_RX_CH_MAP		Section 15.5.6.8
30h	CPSW_PORT_P0_RX_DSCP_PRI_MAP0		Section 15.5.6.9
34h	CPSW_PORT_P0_RX_DSCP_PRI_MAP1		Section 15.5.6.10
38h	CPSW_PORT_P0_RX_DSCP_PRI_MAP2		Section 15.5.6.11
3Ch	CPSW_PORT_P0_RX_DSCP_PRI_MAP3		Section 15.5.6.12
40h	CPSW_PORT_P0_RX_DSCP_PRI_MAP4		Section 15.5.6.13
44h	CPSW_PORT_P0_RX_DSCP_PRI_MAP5		Section 15.5.6.14
48h	CPSW_PORT_P0_RX_DSCP_PRI_MAP6		Section 15.5.6.15

Table 15-133. CPSW_PORT Registers (continued)

Offset	Acronym	Register Name	Section
4Ch	CPSW_PORT_P0_RX_DSCP_PRI_MAP7		Section 15.5.6.16
100h	CPSW_PORT_P1_CTRL		Section 15.5.6.17
104h	CPSW_PORT_P1_TS_CTL2		Section 15.5.6.18
108h	CPSW_PORT_P1_MAX_BLKS		Section 15.5.6.19
10Ch	CPSW_PORT_P1_BLK_CNT		Section 15.5.6.20
110h	CPSW_PORT_P1_TX_IN_CTL		Section 15.5.6.21
114h	CPSW_PORT_P1_VLAN		Section 15.5.6.22
118h	CPSW_PORT_P1_TX_PRI_MAP		Section 15.5.6.23
11Ch	CPSW_PORT_P1_TS_SEQ_MTYPE		Section 15.5.6.24
120h	CPSW_PORT_P1_SA_LO		Section 15.5.6.25
124h	CPSW_PORT_P1_SA_HI		Section 15.5.6.26
128h	CPSW_PORT_P1_SEND_PERCENT		Section 15.5.6.27
130h	CPSW_PORT_P1_RX_DSCP_PRI_MAP0		Section 15.5.6.28
134h	CPSW_PORT_P1_RX_DSCP_PRI_MAP1		Section 15.5.6.29
138h	CPSW_PORT_P1_RX_DSCP_PRI_MAP2		Section 15.5.6.30
13Ch	CPSW_PORT_P1_RX_DSCP_PRI_MAP3		Section 15.5.6.31
140h	CPSW_PORT_P1_RX_DSCP_PRI_MAP4		Section 15.5.6.32
144h	CPSW_PORT_P1_RX_DSCP_PRI_MAP5		Section 15.5.6.33
148h	CPSW_PORT_P1_RX_DSCP_PRI_MAP6		Section 15.5.6.34
14Ch	CPSW_PORT_P1_RX_DSCP_PRI_MAP7		Section 15.5.6.35
200h	CPSW_PORT_P2_CTRL		Section 15.5.6.36
204h	CPSW_PORT_P2_TS_CTL2		Section 15.5.6.37
208h	CPSW_PORT_P2_MAX_BLKS		Section 15.5.6.38
20Ch	CPSW_PORT_P2_BLK_CNT		Section 15.5.6.39
210h	CPSW_PORT_P2_TX_IN_CTL		Section 15.5.6.40
214h	CPSW_PORT_P2_VLAN		Section 15.5.6.41
218h	CPSW_PORT_P2_TX_PRI_MAP		Section 15.5.6.42
21Ch	CPSW_PORT_P2_TS_SEQ_MTYPE		Section 15.5.6.43
220h	CPSW_PORT_P2_SA_LO		Section 15.5.6.44
224h	CPSW_PORT_P2_SA_HI		Section 15.5.6.45
228h	CPSW_PORT_P2_SEND_PERCENT		Section 15.5.6.46
230h	CPSW_PORT_P2_RX_DSCP_PRI_MAP0		Section 15.5.6.47
234h	CPSW_PORT_P2_RX_DSCP_PRI_MAP1		Section 15.5.6.48
238h	CPSW_PORT_P2_RX_DSCP_PRI_MAP2		Section 15.5.6.49
23Ch	CPSW_PORT_P2_RX_DSCP_PRI_MAP3		Section 15.5.6.50
240h	CPSW_PORT_P2_RX_DSCP_PRI_MAP4		Section 15.5.6.51
244h	CPSW_PORT_P2_RX_DSCP_PRI_MAP5		Section 15.5.6.52

Table 15-133. CPSW_PORT Registers (continued)

Offset	Acronym	Register Name	Section
248h	CPSW_PORT_P2_RX_DSCP_PRI_MAP6		Section 15.5.6.53
24Ch	CPSW_PORT_P2_RX_DSCP_PRI_MAP7		Section 15.5.6.54

15.5.6.1 CPSW_PORT_P0_CTRL Register (offset = 0h) [reset = 0h]

CPSW_PORT_P0_CTRL is shown in Figure 15-120 and described in Table 15-134.

CPSW PORT 0 CONTROL REGISTER

Figure 15-120. CPSW_PORT_P0_CTRL Register

31	30	29	28	27	26	25	24
RESERVED	P0_DLR_CPDMA_CH			RESERVED			P0_PASS_PRI_TAGGED
R-X	R/W-X			R-X			R/W-X
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED		P0_VLAN_LTYPE2_EN	P0_VLAN_LTYPE1_EN	RESERVED			P0_DSCP_PRI_EN
R-X		R/W-X	R/W-X	R-X			R/W-X
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							
R-0h							

Table 15-134. CPSW_PORT_P0_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	P0_DLR_CPDMA_CH	R/W	X	Port 0 DLR CPDMA Channel This field indicates the CPDMA channel that DLR packets will be received on.
27-25	RESERVED	R	X	
24	P0_PASS_PRI_TAGGED	R/W	X	Port 0 Pass Priority Tagged 0h (R/W) = Priority tagged packets have the zero VID replaced with the input port P0_PORT_VLAN[11 to 0] 1h (R/W) = Priority tagged packets are processed unchanged.
23-22	RESERVED	R	X	
21	P0_VLAN_LTYPE2_EN	R/W	X	Port 0 VLAN LTYPE 2 enable 0h (R/W) = Disabled 1h (R/W) = Enabled
20	P0_VLAN_LTYPE1_EN	R/W	X	Port 0 VLAN LTYPE 1 enable 0h (R/W) = Disabled 1h (R/W) = Enabled
19-17	RESERVED	R	X	
16	P0_DSCP_PRI_EN	R/W	X	Port 0 DSCP Priority Enable. All non-tagged IPV4 packets have their received packet priority determined by mapping the 6 TOS bits through the port DSCP priority mapping registers. 0h (R/W) = DSCP priority disabled 1h (R/W) = DSCP priority enabled
15-0	RESERVED	R	0h	

15.5.6.2 CPSW_PORT_P0_MAX_BLKS Register (offset = 8h) [reset = 104h]

CPSW_PORT_P0_MAX_BLKS is shown in Figure 15-121 and described in Table 15-135.

CPSW PORT 0 MAXIMUM FIFO BLOCKS REGISTER

Figure 15-121. CPSW_PORT_P0_MAX_BLKS Register

31	30	29	28	27	26	25	24
RESERVED							
R-X							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-X							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							P0_TX_MAX_BLKS
R-0h							R/W-10h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
P0_TX_MAX_BLKS				P0_RX_MAX_BLKS			
R/W-10h				R/W-4h			

Table 15-135. CPSW_PORT_P0_MAX_BLKS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	X	
15-9	RESERVED	R	0h	
8-4	P0_TX_MAX_BLKS	R/W	10h	Transmit FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical transmit priority queues. 0x10 is the recommended value of p0_tx_max_blks. Port 0 should remain in flow control mode. 0xe is the minimum value tx max blks.
3-0	P0_RX_MAX_BLKS	R/W	4h	Receive FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical receive queue. 0x4 is the recommended value. 0x3 is the minimum value rx max blks and 0x6 is the maximum value.

15.5.6.3 CPSW_PORT_P0_BLK_CNT Register (offset = Ch) [reset = 41h]

CPSW_PORT_P0_BLK_CNT is shown in Figure 15-122 and described in Table 15-136.

CPSW PORT 0 FIFO BLOCK USAGE COUNT (READ ONLY)

Figure 15-122. CPSW_PORT_P0_BLK_CNT Register

31	30	29	28	27	26	25	24
RESERVED							
R-X							
23	22	21	20	19	18	17	16
RESERVED							
R-X							
15	14	13	12	11	10	9	8
RESERVED							P0_TX_BLK_CNT
R-0h							R-4h
7	6	5	4	3	2	1	0
P0_TX_BLK_CNT				P0_RX_BLK_CNT			
R-4h				R-1h			

Table 15-136. CPSW_PORT_P0_BLK_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	X	
15-9	RESERVED	R	0h	
8-4	P0_TX_BLK_CNT	R	4h	Port 0 Transmit Block Count Usage - This value is the number of blocks allocated to the FIFO logical transmit queues.
3-0	P0_RX_BLK_CNT	R	1h	Port 0 Receive Block Count Usage - This value is the number of blocks allocated to the FIFO logical receive queues.

15.5.6.4 CPSW_PORT_P0_TX_IN_CTL Register (offset = 10h) [reset = 40C0h]

CPSW_PORT_P0_TX_IN_CTL is shown in Figure 15-123 and described in Table 15-137.

CPSW PORT 0 TRANSMIT FIFO CONTROL

Figure 15-123. CPSW_PORT_P0_TX_IN_CTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-X							
23	22	21	20	19	18	17	16
TX_RATE_EN				RESERVED		TX_IN_SEL	
R/W-X				R-X		R/W-X	
15	14	13	12	11	10	9	8
TX_BLKS_REM				RESERVED		TX_PRI_WDS	
R/W-4h				R-0h		R/W-C0h	
7	6	5	4	3	2	1	0
TX_PRI_WDS							
R/W-C0h							

Table 15-137. CPSW_PORT_P0_TX_IN_CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	X	
23-20	TX_RATE_EN	R/W	X	Transmit FIFO Input Rate Enable
19-18	RESERVED	R	X	
17-16	TX_IN_SEL	R/W	X	Transmit FIFO Input Queue Type Select. Note that Dual MAC mode is not compatible with escalation or shaping because dual mac mode forces round robin priority on FIFO egress. 0h (R/W) = Normal priority mode 1h (R/W) = Dual MAC mode 2h (R/W) = Rate Limit mode 3h (R/W) = Reserved
15-12	TX_BLKS_REM	R/W	4h	Transmit FIFO Input Blocks to subtract in dual mac mode
11-10	RESERVED	R	0h	
9-0	TX_PRI_WDS	R/W	C0h	Transmit FIFO Words in queue

15.5.6.5 CPSW_PORT_P0_VLAN Register (offset = 14h) [reset = 0h]

CPSW_PORT_P0_VLAN is shown in Figure 15-124 and described in Table 15-138.

CPSW PORT 0 VLAN REGISTER

Figure 15-124. CPSW_PORT_P0_VLAN Register

31	30	29	28	27	26	25	24
RESERVED							
R-X							

23	22	21	20	19	18	17	16
RESERVED							
R-X							

15	14	13	12	11	10	9	8
PORT_PRI			PORT_CFI	PORT_VID			
R/W-0h			R/W-0h	R/W-0h			

7	6	5	4	3	2	1	0
PORT_VID							
R/W-0h							

Table 15-138. CPSW_PORT_P0_VLAN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	X	
15-13	PORT_PRI	R/W	0h	Port VLAN Priority (7 is highest priority)
12	PORT_CFI	R/W	0h	Port CFI bit
11-0	PORT_VID	R/W	0h	Port VLAN ID

15.5.6.6 CPSW_PORT_P0_TX_PRI_MAP Register (offset = 18h) [reset = 33221001h]

CPSW_PORT_P0_TX_PRI_MAP is shown in Figure 15-125 and described in Table 15-139.

CPSW PORT 0 TX HEADER PRI TO SWITCH PRI MAPPING REGISTER

Figure 15-125. CPSW_PORT_P0_TX_PRI_MAP Register

31	30	29	28	27	26	25	24
RESERVED		PRI7		RESERVED		PRI6	
R-X		R/W-X		R-X		R/W-X	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED		PRI5		RESERVED		PRI4	
R-X		R/W-X		R-X		R/W-X	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED		PRI3		RESERVED		PRI2	
R-0h		R/W-1h		R-0h		R/W-0h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED		PRI1		RESERVED		PRI0	
R-0h		R/W-0h		R-0h		R/W-1h	

Table 15-139. CPSW_PORT_P0_TX_PRI_MAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	X	
29-28	PRI7	R/W	X	Priority 7 - A packet header priority of 0x7 is given this switch queue pri.
27-26	RESERVED	R	X	
25-24	PRI6	R/W	X	Priority 6 - A packet header priority of 0x6 is given this switch queue pri.
23-22	RESERVED	R	X	
21-20	PRI5	R/W	X	Priority 5 - A packet header priority of 0x5 is given this switch queue pri.
19-18	RESERVED	R	X	
17-16	PRI4	R/W	X	Priority 4 - A packet header priority of 0x4 is given this switch queue pri.
15-14	RESERVED	R	0h	
13-12	PRI3	R/W	1h	Priority 3 - A packet header priority of 0x3 is given this switch queue pri.
11-10	RESERVED	R	0h	
9-8	PRI2	R/W	0h	Priority 2 - A packet header priority of 0x2 is given this switch queue pri.
7-6	RESERVED	R	0h	
5-4	PRI1	R/W	0h	Priority 1 - A packet header priority of 0x1 is given this switch queue pri.
3-2	RESERVED	R	0h	
1-0	PRI0	R/W	1h	Priority 0 - A packet header priority of 0x0 is given this switch queue pri.

15.5.6.7 CPSW_PORT_P0_CPDMA_TX_PRI_MAP Register (offset = 1Ch) [reset = 76543210h]

CPSW_PORT_P0_CPDMA_TX_PRI_MAP is shown in Figure 15-126 and described in Table 15-140.

CPSW CPDMA TX (PORT 0 RX) PKT PRIORITY TO HEADER PRIORITY

Figure 15-126. CPSW_PORT_P0_CPDMA_TX_PRI_MAP Register

31	30	29	28	27	26	25	24
RESERVED	PRI7			RESERVED	PRI6		
R-X	R/W-X			R-X	R/W-X		

23	22	21	20	19	18	17	16
RESERVED	PRI5			RESERVED	PRI4		
R-X	R/W-X			R-X	R/W-X		

15	14	13	12	11	10	9	8
RESERVED	PRI3			RESERVED	PRI2		
R-0h	R/W-3h			R-0h	R/W-2h		

7	6	5	4	3	2	1	0
RESERVED	PRI1			RESERVED	PRI0		
R-0h	R/W-1h			R-0h	R/W-0h		

Table 15-140. CPSW_PORT_P0_CPDMA_TX_PRI_MAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI7	R/W	X	Priority 7 - A packet pri of 0x7 is mapped (changed) to this header packet pri.
27	RESERVED	R	X	
26-24	PRI6	R/W	X	Priority 6 - A packet pri of 0x6 is mapped (changed) to this header packet pri.
23	RESERVED	R	X	
22-20	PRI5	R/W	X	Priority 5 - A packet pri of 0x5 is mapped (changed) to this header packet pri.
19	RESERVED	R	X	
18-16	PRI4	R/W	X	Priority 4 - A packet pri of 0x4 is mapped (changed) to this header packet pri.
15	RESERVED	R	0h	
14-12	PRI3	R/W	3h	Priority 3 - A packet pri of 0x3 is mapped (changed) to this header packet pri.
11	RESERVED	R	0h	
10-8	PRI2	R/W	2h	Priority 2 - A packet pri of 0x2 is mapped (changed) to this header packet pri.
7	RESERVED	R	0h	
6-4	PRI1	R/W	1h	Priority 1 - A packet pri of 0x1 is mapped (changed) to this header packet pri.
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	Priority 0 - A packet pri of 0x0 is mapped (changed) to this header packet pri.

15.5.6.8 CPSW_PORT_P0_CPDMA_RX_CH_MAP Register (offset = 20h) [reset = 0h]

CPSW_PORT_P0_CPDMA_RX_CH_MAP is shown in Figure 15-127 and described in Table 15-141.

CPSW CPDMA RX (PORT 0 TX) SWITCH PRIORITY TO DMA CHANNEL

Figure 15-127. CPSW_PORT_P0_CPDMA_RX_CH_MAP Register

31	30	29	28	27	26	25	24
RESERVED	P2_PRI3			RESERVED	P2_PRI2		
R-X	R/W-X			R-X	R/W-X		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED	P2_PRI1			RESERVED	P2_PRI0		
R-X	R/W-X			R-X	R/W-X		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED	P1_PRI3			RESERVED	P1_PRI2		
R-0h	R/W-0h			R-0h	R/W-0h		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	P1_PRI1			RESERVED	P1_PRI0		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-141. CPSW_PORT_P0_CPDMA_RX_CH_MAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	P2_PRI3	R/W	X	Port 2 Priority 3 packets go to this CPDMA Rx Channel
27	RESERVED	R	X	
26-24	P2_PRI2	R/W	X	Port 2 Priority 2 packets go to this CPDMA Rx Channel
23	RESERVED	R	X	
22-20	P2_PRI1	R/W	X	Port 2 Priority 1 packets go to this CPDMA Rx Channel
19	RESERVED	R	X	
18-16	P2_PRI0	R/W	X	Port 2 Priority 0 packets go to this CPDMA Rx Channel
15	RESERVED	R	0h	
14-12	P1_PRI3	R/W	0h	Port 1 Priority 3 packets go to this CPDMA Rx Channel
11	RESERVED	R	0h	
10-8	P1_PRI2	R/W	0h	Port 1 Priority 2 packets go to this CPDMA Rx Channel
7	RESERVED	R	0h	
6-4	P1_PRI1	R/W	0h	Port 1 Priority 1 packets go to this CPDMA Rx Channel
3	RESERVED	R	0h	
2-0	P1_PRI0	R/W	0h	Port 1 Priority 0 packets go to this CPDMA Rx Channel

15.5.6.9 CPSW_PORT_P0_RX_DSCP_PRI_MAP0 Register (offset = 30h) [reset = 0h]

CPSW_PORT_P0_RX_DSCP_PRI_MAP0 is shown in Figure 15-128 and described in Table 15-142.

CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 0

Figure 15-128. CPSW_PORT_P0_RX_DSCP_PRI_MAP0 Register

31	30	29	28	27	26	25	24
RESERVED	PRI7			RESERVED	PRI6		
R-X	R/W-X			R-X	R/W-X		
23	22	21	20	19	18	17	16
RESERVED	PRI5			RESERVED	PRI4		
R-X	R/W-X			R-X	R/W-X		
15	14	13	12	11	10	9	8
RESERVED	PRI3			RESERVED	PRI2		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI1			RESERVED	PRI0		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-142. CPSW_PORT_P0_RX_DSCP_PRI_MAP0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI7	R/W	X	Priority 7. A packet TOS of 0d7 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI6	R/W	X	Priority 6. A packet TOS of 0d6 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI5	R/W	X	Priority 5. A packet TOS of 0d5 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI4	R/W	X	Priority 4. A packet TOS of 0d4 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI3	R/W	0h	Priority 3. A packet TOS of 0d3 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI2	R/W	0h	Priority 2. A packet TOS of 0d2 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI1	R/W	0h	Priority 1. A packet TOS of 0d1 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	Priority 0. A packet TOS of 0d0 is mapped to this received packet priority.

15.5.6.10 CPSW_PORT_P0_RX_DSCP_PRI_MAP1 Register (offset = 34h) [reset = 0h]

CPSW_PORT_P0_RX_DSCP_PRI_MAP1 is shown in Figure 15-129 and described in Table 15-143.

CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 1

Figure 15-129. CPSW_PORT_P0_RX_DSCP_PRI_MAP1 Register

31	30	29	28	27	26	25	24
RESERVED	PRI15			RESERVED	PRI14		
R-X	R/W-X			R-X	R/W-X		
23	22	21	20	19	18	17	16
RESERVED	PRI13			RESERVED	PRI12		
R-X	R/W-X			R-X	R/W-X		
15	14	13	12	11	10	9	8
RESERVED	PRI11			RESERVED	PRI10		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI9			RESERVED	PRI8		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-143. CPSW_PORT_P0_RX_DSCP_PRI_MAP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI15	R/W	X	Priority 15. A packet TOS of 0d15 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI14	R/W	X	Priority 14. A packet TOS of 0d14 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI13	R/W	X	Priority 13. A packet TOS of 0d13 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI12	R/W	X	Priority 12. A packet TOS of 0d12 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI11	R/W	0h	Priority 11. A packet TOS of 0d11 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI10	R/W	0h	Priority 10. A packet TOS of 0d10 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI9	R/W	0h	Priority 9. A packet TOS of 0d9 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI8	R/W	0h	Priority 8. A packet TOS of 0d8 is mapped to this received packet priority.

15.5.6.11 CPSW_PORT_P0_RX_DSCP_PRI_MAP2 Register (offset = 38h) [reset = 0h]

CPSW_PORT_P0_RX_DSCP_PRI_MAP2 is shown in Figure 15-130 and described in Table 15-144.

CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 2

Figure 15-130. CPSW_PORT_P0_RX_DSCP_PRI_MAP2 Register

31	30	29	28	27	26	25	24
RESERVED	PRI23			RESERVED	PRI22		
R-X	R/W-X			R-X	R/W-X		
23	22	21	20	19	18	17	16
RESERVED	PRI21			RESERVED	PRI20		
R-X	R/W-X			R-X	R/W-X		
15	14	13	12	11	10	9	8
RESERVED	PRI19			RESERVED	PRI18		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI17			RESERVED	PRI16		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-144. CPSW_PORT_P0_RX_DSCP_PRI_MAP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI23	R/W	X	Priority 23. A packet TOS of 0d23 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI22	R/W	X	Priority 22. A packet TOS of 0d22 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI21	R/W	X	Priority 21. A packet TOS of 0d21 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI20	R/W	X	Priority 20. A packet TOS of 0d20 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI19	R/W	0h	Priority 19. A packet TOS of 0d19 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI18	R/W	0h	Priority 18. A packet TOS of 0d18 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI17	R/W	0h	Priority 17. A packet TOS of 0d17 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI16	R/W	0h	Priority 16. A packet TOS of 0d16 is mapped to this received packet priority.

15.5.6.12 CPSW_PORT_P0_RX_DSCP_PRI_MAP3 Register (offset = 3Ch) [reset = 0h]

CPSW_PORT_P0_RX_DSCP_PRI_MAP3 is shown in Figure 15-131 and described in Table 15-145.

CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 3

Figure 15-131. CPSW_PORT_P0_RX_DSCP_PRI_MAP3 Register

31	30	29	28	27	26	25	24
RESERVED	PRI31			RESERVED	PRI30		
R-0h	R/W-0h			R-0h	R/W-0h		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED	PRI29			RESERVED	PRI28		
R-0h	R/W-0h			R-0h	R/W-0h		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED	PRI27			RESERVED	PRI26		
R-0h	R/W-0h			R-0h	R/W-0h		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	PRI25			RESERVED	PRI24		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-145. CPSW_PORT_P0_RX_DSCP_PRI_MAP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI31	R/W	0h	Priority 31. A packet TOS of 0d31 is mapped to this received packet priority.
27	RESERVED	R	0h	
26-24	PRI30	R/W	0h	Priority 30. A packet TOS of 0d30 is mapped to this received packet priority.
23	RESERVED	R	0h	
22-20	PRI29	R/W	0h	Priority 29. A packet TOS of 0d39 is mapped to this received packet priority.
19	RESERVED	R	0h	
18-16	PRI28	R/W	0h	Priority 28. A packet TOS of 0d28 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI27	R/W	0h	Priority 27. A packet TOS of 0d27 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI26	R/W	0h	Priority 26. A packet TOS of 0d26 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI25	R/W	0h	Priority 25. A packet TOS of 0d25 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI24	R/W	0h	Priority 24. A packet TOS of 0d24 is mapped to this received packet priority.

15.5.6.13 CPSW_PORT_P0_RX_DSCP_PRI_MAP4 Register (offset = 40h) [reset = 0h]

CPSW_PORT_P0_RX_DSCP_PRI_MAP4 is shown in Figure 15-132 and described in Table 15-146.

CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 4

Figure 15-132. CPSW_PORT_P0_RX_DSCP_PRI_MAP4 Register

31	30	29	28	27	26	25	24
RESERVED	PRI39			RESERVED	PRI38		
R-X	R/W-X			R-X	R/W-X		
23	22	21	20	19	18	17	16
RESERVED	PRI37			RESERVED	PRI36		
R-X	R/W-X			R-X	R/W-X		
15	14	13	12	11	10	9	8
RESERVED	PRI35			RESERVED	PRI34		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI33			RESERVED	PRI32		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-146. CPSW_PORT_P0_RX_DSCP_PRI_MAP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI39	R/W	X	Priority 39. A packet TOS of 0d39 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI38	R/W	X	Priority 38. A packet TOS of 0d38 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI37	R/W	X	Priority 37. A packet TOS of 0d37 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI36	R/W	X	Priority 36. A packet TOS of 0d36 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI35	R/W	0h	Priority 35. A packet TOS of 0d35 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI34	R/W	0h	Priority 34. A packet TOS of 0d34 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI33	R/W	0h	Priority 33. A packet TOS of 0d33 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI32	R/W	0h	Priority 32. A packet TOS of 0d32 is mapped to this received packet priority.

15.5.6.14 CPSW_PORT_P0_RX_DSCP_PRI_MAP5 Register (offset = 44h) [reset = 0h]

CPSW_PORT_P0_RX_DSCP_PRI_MAP5 is shown in Figure 15-133 and described in Table 15-147.

CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 5

Figure 15-133. CPSW_PORT_P0_RX_DSCP_PRI_MAP5 Register

31	30	29	28	27	26	25	24
RESERVED	PRI47			RESERVED	PRI46		
R-X	R/W-X			R-X	R/W-X		
23	22	21	20	19	18	17	16
RESERVED	PRI45			RESERVED	PRI44		
R-X	R/W-X			R-X	R/W-X		
15	14	13	12	11	10	9	8
RESERVED	PRI43			RESERVED	PRI42		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI41			RESERVED	PRI40		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-147. CPSW_PORT_P0_RX_DSCP_PRI_MAP5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI47	R/W	X	Priority 47. A packet TOS of 0d47 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI46	R/W	X	Priority 46. A packet TOS of 0d46 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI45	R/W	X	Priority 45. A packet TOS of 0d45 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI44	R/W	X	Priority 44. A packet TOS of 0d44 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI43	R/W	0h	Priority 43. A packet TOS of 0d43 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI42	R/W	0h	Priority 42. A packet TOS of 0d42 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI41	R/W	0h	Priority 41. A packet TOS of 0d41 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI40	R/W	0h	Priority 40. A packet TOS of 0d40 is mapped to this received packet priority.

15.5.6.15 CPSW_PORT_P0_RX_DSCP_PRI_MAP6 Register (offset = 48h) [reset = 0h]

CPSW_PORT_P0_RX_DSCP_PRI_MAP6 is shown in Figure 15-134 and described in Table 15-148.

CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 6

Figure 15-134. CPSW_PORT_P0_RX_DSCP_PRI_MAP6 Register

31	30	29	28	27	26	25	24
RESERVED	PRI55			RESERVED	PRI54		
R-X	R/W-X			R-X	R/W-X		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED	PRI53			RESERVED	PRI52		
R-X	R/W-X			R-X	R/W-X		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED	PRI51			RESERVED	PRI50		
R-0h	R/W-0h			R-0h	R/W-0h		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	PRI49			RESERVED	PRI48		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-148. CPSW_PORT_P0_RX_DSCP_PRI_MAP6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI55	R/W	X	Priority 55. A packet TOS of 0d55 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI54	R/W	X	Priority 54. A packet TOS of 0d54 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI53	R/W	X	Priority 53. A packet TOS of 0d53 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI52	R/W	X	Priority 52. A packet TOS of 0d52 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI51	R/W	0h	Priority 51. A packet TOS of 0d51 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI50	R/W	0h	Priority 50. A packet TOS of 0d50 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI49	R/W	0h	Priority 49. A packet TOS of 0d49 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI48	R/W	0h	Priority 48. A packet TOS of 0d48 is mapped to this received packet priority.

15.5.6.16 CPSW_PORT_P0_RX_DSCP_PRI_MAP7 Register (offset = 4Ch) [reset = 0h]

CPSW_PORT_P0_RX_DSCP_PRI_MAP7 is shown in Figure 15-135 and described in Table 15-149.

CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 7

Figure 15-135. CPSW_PORT_P0_RX_DSCP_PRI_MAP7 Register

31	30	29	28	27	26	25	24
RESERVED	PRI63			RESERVED	PRI62		
R-X	R/W-X			R-X	R/W-X		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED	PRI61			RESERVED	PRI60		
R-X	R/W-X			R-X	R/W-X		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED	PRI59			RESERVED	PRI58		
R-0h	R/W-0h			R-0h	R/W-0h		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	PRI57			RESERVED	PRI56		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-149. CPSW_PORT_P0_RX_DSCP_PRI_MAP7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI63	R/W	X	Priority 63. A packet TOS of 0d63 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI62	R/W	X	Priority 62. A packet TOS of 0d62 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI61	R/W	X	Priority 61. A packet TOS of 0d61 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI60	R/W	X	Priority 60. A packet TOS of 0d60 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI59	R/W	0h	Priority 59. A packet TOS of 0d59 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI58	R/W	0h	Priority 58. A packet TOS of 0d58 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI57	R/W	0h	Priority 57. A packet TOS of 0d57 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI56	R/W	0h	Priority 56. A packet TOS of 0d56 is mapped to this received packet priority.

15.5.6.17 CPSW_PORT_P1_CTRL Register (offset = 100h) [reset = 0h]

CPSW_PORT_P1_CTRL is shown in Figure 15-136 and described in Table 15-150.

CPSW PORT 1 CONTROL REGISTER

Figure 15-136. CPSW_PORT_P1_CTRL Register

31	30	29	28	27	26	25	24
RESERVED						P1_TX_CLKSTOP_EN	P1_PASS_PRI_TAGGED
R-X						R/W-X	R/W-X

23	22	21	20	19	18	17	16
RESERVED		P1_VLAN_LTYPE2_EN	P1_VLAN_LTYPE1_EN	RESERVED			P1_DSCP_PRI_EN
R-X		R/W-X	R/W-X	R-X			R/W-X

15	14	13	12	11	10	9	8
P1_TS_107	P1_TS_320	P1_TS_319	P1_TS_132	P1_TS_131	P1_TS_130	P1_TS_129	P1_TS_TTL_NONZERO
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
P1_TS_UNI_EN	P1_TS_ANNEX_F_EN	P1_TS_ANNEX_E_EN	P1_TS_ANNEX_D_EN	P1_TS_LTYPE2_EN	P1_TS_LTYPE1_EN	P1_TS_TX_EN	P1_TS_RX_EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-150. CPSW_PORT_P1_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	X	
25	P1_TX_CLKSTOP_EN	R/W	X	Port 1 Transmit clockstop enable 0h (R/W) = RGMII transmit clockstop not enabled 1h (R/W) = RGMII transmit clockstop enabled. The transmit clock will be stopped after the LPI state is entered (and indicated to the CPRGMII) and the P1_Idle2LPI time is counted (counter value reused). The P1_Idle2LPI counter value must be greater than 9 transmit clocks (slowest clock).
24	P1_PASS_PRI_TAGGED	R/W	X	Port 1 Pass Priority Tagged 0h (R/W) = Priority tagged packets have the zero VID replaced with the input port P1_PORT_VLAN[11 to 0] 1h (R/W) = Priority tagged packets are processed unchanged.
23-22	RESERVED	R	X	
21	P1_VLAN_LTYPE2_EN	R/W	X	Port 1 VLAN LTYPE 2 enable 0h (R/W) = Disabled 1h (R/W) = VLAN LTYPE2 enabled on transmit and receive
20	P1_VLAN_LTYPE1_EN	R/W	X	Port 1 VLAN LTYPE 1 enable 0h (R/W) = Disabled 1h (R/W) = VLAN LTYPE1 enabled on transmit and receive
19-17	RESERVED	R	X	
16	P1_DSCP_PRI_EN	R/W	X	Port 1 DSCP Priority Enable. All non-tagged IPV4 packets have their received packet priority determined by mapping the 6 TOS bits through the port DSCP priority mapping registers. 0h (R/W) = DSCP priority disabled 1h (R/W) = DSCP priority enabled
15	P1_TS_107	R/W	0h	Port 1 Time Sync Destination IP Address 107 enable 0h (R/W) = Disabled 1h (R/W) = Destination IP address (dec) 224.0.0.107 is enabled.

Table 15-150. CPSW_PORT_P1_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	P1_TS_320	R/W	0h	Port 1 Time Sync Destination Port Number 320 enable 0h (R/W) = Disabled 1h (R/W) = Annex D (UDP/IPv4) time sync packet destination port number 320 (decimal) is enabled.
13	P1_TS_319	R/W	0h	Port 1 Time Sync Destination Port Number 319 enable 0h (R/W) = DSCP priority disabled 1h (R/W) = Annex D (UDP/IPv4) time sync packet destination port number 319 (decimal) is enabled.
12	P1_TS_132	R/W	0h	Port 1 Time Sync Destination IP Address 132 enable 0h (R/W) = DSCP priority disabled 1h (R/W) = Annex D (UDP/IPv4) time sync packet destination IP address number 132 (decimal) is enabled.
11	P1_TS_131	R/W	0h	Port 1 Time Sync Destination IP Address 131 enable 0h (R/W) = DSCP priority disabled 1h (R/W) = Annex D (UDP/IPv4) time sync packet destination IP address number 131 (decimal) is enabled.
10	P1_TS_130	R/W	0h	Port 1 Time Sync Destination IP Address 130 enable 0h (R/W) = DSCP priority disabled 1h (R/W) = Annex D (UDP/IPv4) time sync packet destination IP address number 130 (decimal) is enabled.
9	P1_TS_129	R/W	0h	Port 1 Time Sync Destination IP Address 129 enable 0h (R/W) = DSCP priority disabled 1h (R/W) = Annex D (UDP/IPv4) time sync packet destination IP address number 129 (decimal) is enabled.
8	P1_TS_TTL_NONZERO	R/W	0h	Port 1 Time Sync Time To Live Non-zero enable. 0h (R/W) = TTL must be zero 1h (R/W) = TTL may be any value
7	P1_TS_UNI_EN	R/W	0h	Port 1 Time Sync Unicast Enable 0h (R/W) = Unicast disabled 1h (R/W) = Unicast enabled
6	P1_TS_ANNEX_F_EN	R/W	0h	Port 1 Time Sync Annex F enable 0h (R/W) = Annex F disabled 1h (R/W) = Annex F enabled
5	P1_TS_ANNEX_E_EN	R/W	0h	Port 1 Time Sync Annex E enable 0h (R/W) = Annex E disabled 1h (R/W) = Annex E enabled
4	P1_TS_ANNEX_D_EN	R/W	0h	Port 1 Time Sync Annex D enable 0h (R/W) = Annex D disabled 1h (R/W) = Annex D enabled
3	P1_TS_LTYPE2_EN	R/W	0h	Port 1 Time Sync LTYPE 2 enable 0h (R/W) = Disabled 1h (R/W) = Enabled
2	P1_TS_LTYPE1_EN	R/W	0h	Port 1 Time Sync LTYPE 1 enable 0h (R/W) = Disabled 1h (R/W) = Enabled
1	P1_TS_TX_EN	R/W	0h	Port 1 Time Sync Transmit Enable 0h (R/W) = Disabled 1h (R/W) = Enabled
0	P1_TS_RX_EN	R/W	0h	Port 1 Time Sync Receive Enable 0h (R/W) = Port 1 Receive Time Sync disabled 1h (R/W) = Port 1 Receive Time Sync enabled

15.5.6.18 CPSW_PORT_P1_TS_CTL2 Register (offset = 104h) [reset = 40000h]

CPSW_PORT_P1_TS_CTL2 is shown in Figure 15-137 and described in Table 15-151.

Figure 15-137. CPSW_PORT_P1_TS_CTL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED										P1_DOMAIN_OFFSET					
R-0h										R/W-4h					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
P1_TS_MCAST_TYPE_EN															
R/W-0h															

Table 15-151. CPSW_PORT_P1_TS_CTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-16	P1_DOMAIN_OFFSET	R/W	4h	Domain offset value
15-0	P1_TS_MCAST_TYPE_EN	R/W	0h	Multicast Type Enable

15.5.6.19 CPSW_PORT_P1_MAX_BLKS Register (offset = 108h) [reset = 113h]

CPSW_PORT_P1_MAX_BLKS is shown in Figure 15-138 and described in Table 15-152.

CPSW PORT 1 MAXIMUM FIFO BLOCKS REGISTER

Figure 15-138. CPSW_PORT_P1_MAX_BLKS Register

31	30	29	28	27	26	25	24
RESERVED							
R-X							
23	22	21	20	19	18	17	16
RESERVED							
R-X							
15	14	13	12	11	10	9	8
RESERVED							P1_TX_MAX_BLKS
R-0h							R/W-11h
7	6	5	4	3	2	1	0
P1_TX_MAX_BLKS				P1_RX_MAX_BLKS			
R/W-11h				R/W-3h			

Table 15-152. CPSW_PORT_P1_MAX_BLKS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	X	
15-9	RESERVED	R	0h	
8-4	P1_TX_MAX_BLKS	R/W	11h	Transmit FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical transmit priority queues. 0x11 is the recommended value of p1_tx_max_blks unless the port is in fullduplex flow control mode. In flow control mode, the p1_rx_max_blks will need to increase in order to accept the required run out in fullduplex mode. This value will need to decrease by the amount of increase in p1_rx_max_blks. 0xe is the minimum value tx max blks.
3-0	P1_RX_MAX_BLKS	R/W	3h	Receive FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical receive queue. This value must be greater than or equal to 0x3. It should be increased In fullduplex flow control mode to 0x5 or 0x6 depending on the required runout space. The p1_tx_max_blks value must be decreased by the amount of increase in p1_rx_max_blks. 0x3 is the minimum value rx max blks and 0x6 is the maximum value.

15.5.6.20 CPSW_PORT_P1_BLK_CNT Register (offset = 10Ch) [reset = 41h]

CPSW_PORT_P1_BLK_CNT is shown in Figure 15-139 and described in Table 15-153.

CPSW PORT 1 FIFO BLOCK USAGE COUNT (READ ONLY)

Figure 15-139. CPSW_PORT_P1_BLK_CNT Register

31	30	29	28	27	26	25	24
RESERVED							
R-X							
23	22	21	20	19	18	17	16
RESERVED							
R-X							
15	14	13	12	11	10	9	8
RESERVED							P1_TX_BLK_CNT
R-0h							R-4h
7	6	5	4	3	2	1	0
P1_TX_BLK_CNT				P1_RX_BLK_CNT			
R-4h				R-1h			

Table 15-153. CPSW_PORT_P1_BLK_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	X	
15-9	RESERVED	R	0h	
8-4	P1_TX_BLK_CNT	R	4h	Port 1 Transmit Block Count Usage - This value is the number of blocks allocated to the FIFO logical transmit queues.
3-0	P1_RX_BLK_CNT	R	1h	Port 1 Receive Block Count Usage - This value is the number of blocks allocated to the FIFO logical receive queues.

15.5.6.21 CPSW_PORT_P1_TX_IN_CTL Register (offset = 110h) [reset = 80040C0h]

CPSW_PORT_P1_TX_IN_CTL is shown in Figure 15-140 and described in Table 15-154.

CPSW PORT 1 TRANSMIT FIFO CONTROL

Figure 15-140. CPSW_PORT_P1_TX_IN_CTL Register

31	30	29	28	27	26	25	24
RESERVED				HOST_BLKS_REM			
R-X				R/W-X			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TX_RATE_EN				RESERVED		TX_IN_SEL	
R/W-X				R-X		R/W-X	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TX_BLKS_REM				RESERVED		TX_PRI_WDS	
R/W-4h				R-0h		R/W-C0h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TX_PRI_WDS							
R/W-C0h							

Table 15-154. CPSW_PORT_P1_TX_IN_CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	X	
27-24	HOST_BLKS_REM	R/W	X	Transmit FIFO Blocks that must be free before a non rate-limited CPDMA channel can begin sending a packet to the FIFO.
23-20	TX_RATE_EN	R/W	X	Transmit FIFO Input Rate Enable
19-18	RESERVED	R	X	
17-16	TX_IN_SEL	R/W	X	Transmit FIFO Input Queue Type Select 0h (R/W) = Normal priority mode 1h (R/W) = Reserved 2h (R/W) = Rate Limit mode 3h (R/W) = Reserved
15-12	TX_BLKS_REM	R/W	4h	Transmit FIFO Input Blocks to subtract in dual mac mode and blocks to subtract on non rate-limited traffic in rate-limit mode.
11-10	RESERVED	R	0h	
9-0	TX_PRI_WDS	R/W	C0h	Transmit FIFO Words in queue

15.5.6.22 CPSW_PORT_P1_VLAN Register (offset = 114h) [reset = 0h]

CPSW_PORT_P1_VLAN is shown in Figure 15-141 and described in Table 15-155.

CPSW PORT 1 VLAN REGISTER

Figure 15-141. CPSW_PORT_P1_VLAN Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-X</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-X</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="3">PORT_PRI</td><td>PORT_CFI</td><td colspan="4">PORT_VID</td></tr>
<tr><td colspan="3">R/W-0h</td><td>R/W-0h</td><td colspan="4">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">PORT_VID</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 15-155. CPSW_PORT_P1_VLAN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	X	
15-13	PORT_PRI	R/W	0h	Port VLAN Priority (7 is highest priority)
12	PORT_CFI	R/W	0h	Port CFI bit
11-0	PORT_VID	R/W	0h	Port VLAN ID

15.5.6.23 CPSW_PORT_P1_TX_PRI_MAP Register (offset = 118h) [reset = 33221001h]

CPSW_PORT_P1_TX_PRI_MAP is shown in Figure 15-142 and described in Table 15-156.

CPSW PORT 1 TX HEADER PRIORITY TO SWITCH PRI MAPPING REGISTER

Figure 15-142. CPSW_PORT_P1_TX_PRI_MAP Register

31	30	29	28	27	26	25	24
RESERVED		PRI7		RESERVED		PRI6	
R-X		R/W-X		R-X		R/W-X	
23	22	21	20	19	18	17	16
RESERVED		PRI5		RESERVED		PRI4	
R-X		R/W-X		R-X		R/W-X	
15	14	13	12	11	10	9	8
RESERVED		PRI3		RESERVED		PRI2	
R-0h		R/W-1h		R-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED		PRI1		RESERVED		PRI0	
R-0h		R/W-0h		R-0h		R/W-1h	

Table 15-156. CPSW_PORT_P1_TX_PRI_MAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	X	
29-28	PRI7	R/W	X	Priority 7 - A packet header priority of 0x7 is given this switch queue pri.
27-26	RESERVED	R	X	
25-24	PRI6	R/W	X	Priority 6 - A packet header priority of 0x6 is given this switch queue pri.
23-22	RESERVED	R	X	
21-20	PRI5	R/W	X	Priority 5 - A packet header priority of 0x5 is given this switch queue pri.
19-18	RESERVED	R	X	
17-16	PRI4	R/W	X	Priority 4 - A packet header priority of 0x4 is given this switch queue pri.
15-14	RESERVED	R	0h	
13-12	PRI3	R/W	1h	Priority 3 - A packet header priority of 0x3 is given this switch queue pri.
11-10	RESERVED	R	0h	
9-8	PRI2	R/W	0h	Priority 2 - A packet header priority of 0x2 is given this switch queue pri.
7-6	RESERVED	R	0h	
5-4	PRI1	R/W	0h	Priority 1 - A packet header priority of 0x1 is given this switch queue pri.
3-2	RESERVED	R	0h	
1-0	PRI0	R/W	1h	Priority 0 - A packet header priority of 0x0 is given this switch queue pri.

15.5.6.24 CPSW_PORT_P1_TS_SEQ_MTYPE Register (offset = 11Ch) [reset = 1E0000h]

CPSW_PORT_P1_TS_SEQ_MTYPE is shown in Figure 15-143 and described in Table 15-157.

CPSW PORT 1 TIME SYNC SEQUENCE ID OFFSET AND MSG TYPE.

Figure 15-143. CPSW_PORT_P1_TS_SEQ_MTYPE Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-X</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">P1_TS_SEQ_ID_OFFSET</td></tr>
<tr><td colspan="2">R-X</td><td colspan="6">R/W-X</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">P1_TS_MSG_TYPE_EN</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">P1_TS_MSG_TYPE_EN</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 15-157. CPSW_PORT_P1_TS_SEQ_MTYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	X	
21-16	P1_TS_SEQ_ID_OFFSET	R/W	X	Port 1 Time Sync Sequence ID Offset This is the number of octets that the sequence ID is offset in the tx and rx time sync message header. The minimum value is 6.
15-0	P1_TS_MSG_TYPE_EN	R/W	0h	Port 1 Time Sync Message Type Enable - Each bit in this field enables the corresponding message type in receive and transmit time sync messages (Bit 0 enables message type 0 etc.).

15.5.6.25 CPSW_PORT_P1_SA_LO Register (offset = 120h) [reset = 0h]

CPSW_PORT_P1_SA_LO is shown in Figure 15-144 and described in Table 15-158.

CPSW CPGMAC_SL1 SOURCE ADDRESS LOW REGISTER

Figure 15-144. CPSW_PORT_P1_SA_LO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-X															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACSRCADDR_7_0								MACSRCADDR_15_8							
R/W-0h								R/W-0h							

Table 15-158. CPSW_PORT_P1_SA_LO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	X	
15-8	MACSRCADDR_7_0	R/W	0h	Source Address Lower 8 bits (byte 0)
7-0	MACSRCADDR_15_8	R/W	0h	Source Address bits 15 to 8 (byte 1)

15.5.6.26 CPSW_PORT_P1_SA_HI Register (offset = 124h) [reset = 0h]

CPSW_PORT_P1_SA_HI is shown in Figure 15-145 and described in Table 15-159.

CPSW CPGMAC_SL1 SOURCE ADDRESS HIGH REGISTER

Figure 15-145. CPSW_PORT_P1_SA_HI Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">MACSRCADDR_23_16</td><td colspan="8">MACSRCADDR_31_24</td></tr>
<tr><td colspan="8">R/W-0h</td><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">MACSRCADDR_39_32</td><td colspan="8">MACSRCADDR_47_40</td></tr>
<tr><td colspan="8">R/W-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 15-159. CPSW_PORT_P1_SA_HI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MACSRCADDR_23_16	R/W	0h	Source Address bits 23 to 16 (byte 2)
23-16	MACSRCADDR_31_24	R/W	0h	Source Address bits 31 to 24 (byte 3)
15-8	MACSRCADDR_39_32	R/W	0h	Source Address bits 39 to 32 (byte 4)
7-0	MACSRCADDR_47_40	R/W	0h	Source Address bits 47 to 40 (byte 5)

15.5.6.27 CPSW_PORT_P1_SEND_PERCENT Register (offset = 128h) [reset = 0h]

CPSW_PORT_P1_SEND_PERCENT is shown in Figure 15-146 and described in Table 15-160.

CPSW PORT 1 TRANSMIT QUEUE SEND PERCENTAGES

Figure 15-146. CPSW_PORT_P1_SEND_PERCENT Register

31	30	29	28	27	26	25	24
RESERVED							
R-X							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED	PRI3_SEND_PERCENT						
R-X	R/W-X						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED	PRI2_SEND_PERCENT						
R-0h	R/W-0h						
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	PRI1_SEND_PERCENT						
R-0h	R/W-0h						

Table 15-160. CPSW_PORT_P1_SEND_PERCENT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	X	
22-16	PRI3_SEND_PERCENT	R/W	X	Priority 3 Transmit Percentage - This percentage value is sent from FIFO priority 3 (maximum) when the p1_pri3_shape_en is set (queue shaping enabled). This is the percentage of the wire that packets from priority 3 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).
15	RESERVED	R	0h	
14-8	PRI2_SEND_PERCENT	R/W	0h	Priority 2 Transmit Percentage - This percentage value is sent from FIFO priority 2 (maximum) when the p1_pri2_shape_en is set (queue shaping enabled). This is the percentage of the wire that packets from priority 2 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).
7	RESERVED	R	0h	
6-0	PRI1_SEND_PERCENT	R/W	0h	Priority 1 Transmit Percentage - This percentage value is sent from FIFO priority 1 (maximum) when the p1_pri1_shape_en is set (queue shaping enabled). This is the percentage of the wire that packets from priority 1 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).

15.5.6.28 CPSW_PORT_P1_RX_DSCP_PRI_MAP0 Register (offset = 130h) [reset = 0h]

CPSW_PORT_P1_RX_DSCP_PRI_MAP0 is shown in Figure 15-147 and described in Table 15-161.

CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 0

Figure 15-147. CPSW_PORT_P1_RX_DSCP_PRI_MAP0 Register

31	30	29	28	27	26	25	24
RESERVED	PRI7			RESERVED	PRI6		
R-0h	R/W-0h			R-0h	R/W-0h		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED	PRI5			RESERVED	PRI4		
R-0h	R/W-0h			R-0h	R/W-0h		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED	PRI3			RESERVED	PRI2		
R-0h	R/W-0h			R-0h	R/W-0h		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	PRI1			RESERVED	PRI0		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-161. CPSW_PORT_P1_RX_DSCP_PRI_MAP0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	0h	Priority 7. A packet TOS of 0d7 is mapped to this received packet priority.
27	RESERVED	R	0h	
26-24	PRI6	R/W	0h	Priority 6. A packet TOS of 0d6 is mapped to this received packet priority.
23	RESERVED	R	0h	
22-20	PRI5	R/W	0h	Priority 5. A packet TOS of 0d5 is mapped to this received packet priority.
19	RESERVED	R	0h	
18-16	PRI4	R/W	0h	Priority 4. A packet TOS of 0d4 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI3	R/W	0h	Priority 3. A packet TOS of 0d3 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI2	R/W	0h	Priority 2. A packet TOS of 0d2 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI1	R/W	0h	Priority 1. A packet TOS of 0d1 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	Priority 0. A packet TOS of 0d0 is mapped to this received packet priority.

15.5.6.29 CPSW_PORT_P1_RX_DSCP_PRI_MAP1 Register (offset = 134h) [reset = 0h]

CPSW_PORT_P1_RX_DSCP_PRI_MAP1 is shown in Figure 15-148 and described in Table 15-162.

CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 1

Figure 15-148. CPSW_PORT_P1_RX_DSCP_PRI_MAP1 Register

31	30	29	28	27	26	25	24
RESERVED	PRI15			RESERVED	PRI14		
R-X	R/W-X			R-X	R/W-X		
23	22	21	20	19	18	17	16
RESERVED	PRI13			RESERVED	PRI12		
R-X	R/W-X			R-X	R/W-X		
15	14	13	12	11	10	9	8
RESERVED	PRI11			RESERVED	PRI10		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI9			RESERVED	PRI8		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-162. CPSW_PORT_P1_RX_DSCP_PRI_MAP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI15	R/W	X	Priority 15. A packet TOS of 0d15 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI14	R/W	X	Priority 14. A packet TOS of 0d14 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI13	R/W	X	Priority 13. A packet TOS of 0d13 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI12	R/W	X	Priority 12. A packet TOS of 0d12 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI11	R/W	0h	Priority 11. A packet TOS of 0d11 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI10	R/W	0h	Priority 10. A packet TOS of 0d10 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI9	R/W	0h	Priority 9. A packet TOS of 0d9 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI8	R/W	0h	Priority 8. A packet TOS of 0d8 is mapped to this received packet priority.

15.5.6.30 CPSW_PORT_P1_RX_DSCP_PRI_MAP2 Register (offset = 138h) [reset = 0h]

CPSW_PORT_P1_RX_DSCP_PRI_MAP2 is shown in Figure 15-149 and described in Table 15-163.

CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 2

Figure 15-149. CPSW_PORT_P1_RX_DSCP_PRI_MAP2 Register

31	30	29	28	27	26	25	24
RESERVED	PRI23			RESERVED	PRI22		
R-X	R/W-X			R-X	R/W-X		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED	PRI21			RESERVED	PRI20		
R-X	R/W-X			R-X	R/W-X		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED	PRI19			RESERVED	PRI18		
R-0h	R/W-0h			R-0h	R/W-0h		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	PRI17			RESERVED	PRI16		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-163. CPSW_PORT_P1_RX_DSCP_PRI_MAP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI23	R/W	X	Priority 23. A packet TOS of 0d23 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI22	R/W	X	Priority 22. A packet TOS of 0d22 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI21	R/W	X	Priority 21. A packet TOS of 0d21 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI20	R/W	X	Priority 20. A packet TOS of 0d20 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI19	R/W	0h	Priority 19. A packet TOS of 0d19 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI18	R/W	0h	Priority 18. A packet TOS of 0d18 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI17	R/W	0h	Priority 17. A packet TOS of 0d17 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI16	R/W	0h	Priority 16. A packet TOS of 0d16 is mapped to this received packet priority.

15.5.6.31 CPSW_PORT_P1_RX_DSCP_PRI_MAP3 Register (offset = 13Ch) [reset = 0h]

CPSW_PORT_P1_RX_DSCP_PRI_MAP3 is shown in Figure 15-150 and described in Table 15-164.

CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 3

Figure 15-150. CPSW_PORT_P1_RX_DSCP_PRI_MAP3 Register

31	30	29	28	27	26	25	24
RESERVED	PRI31			RESERVED	PRI30		
R-X	R/W-X			R-X	R/W-X		
23	22	21	20	19	18	17	16
RESERVED	PRI29			RESERVED	PRI28		
R-X	R/W-X			R-X	R/W-X		
15	14	13	12	11	10	9	8
RESERVED	PRI27			RESERVED	PRI26		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI25			RESERVED	PRI24		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-164. CPSW_PORT_P1_RX_DSCP_PRI_MAP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI31	R/W	X	Priority 31. A packet TOS of 0d31 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI30	R/W	X	Priority 30. A packet TOS of 0d30 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI29	R/W	X	Priority 29. A packet TOS of 0d39 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI28	R/W	X	Priority 28. A packet TOS of 0d28 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI27	R/W	0h	Priority 27. A packet TOS of 0d27 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI26	R/W	0h	Priority 26. A packet TOS of 0d26 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI25	R/W	0h	Priority 25. A packet TOS of 0d25 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI24	R/W	0h	Priority 24. A packet TOS of 0d24 is mapped to this received packet priority.

15.5.6.32 CPSW_PORT_P1_RX_DSCP_PRI_MAP4 Register (offset = 140h) [reset = 0h]

CPSW_PORT_P1_RX_DSCP_PRI_MAP4 is shown in Figure 15-151 and described in Table 15-165.

CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 4

Figure 15-151. CPSW_PORT_P1_RX_DSCP_PRI_MAP4 Register

31	30	29	28	27	26	25	24
RESERVED	PRI39			RESERVED	PRI38		
R-X	R/W-X			R-X	R/W-X		
23	22	21	20	19	18	17	16
RESERVED	PRI37			RESERVED	PRI36		
R-X	R/W-X			R-X	R/W-X		
15	14	13	12	11	10	9	8
RESERVED	PRI35			RESERVED	PRI34		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI33			RESERVED	PRI32		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-165. CPSW_PORT_P1_RX_DSCP_PRI_MAP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI39	R/W	X	Priority 39. A packet TOS of 0d39 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI38	R/W	X	Priority 38. A packet TOS of 0d38 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI37	R/W	X	Priority 37. A packet TOS of 0d37 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI36	R/W	X	Priority 36. A packet TOS of 0d36 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI35	R/W	0h	Priority 35. A packet TOS of 0d35 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI34	R/W	0h	Priority 34. A packet TOS of 0d34 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI33	R/W	0h	Priority 33. A packet TOS of 0d33 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI32	R/W	0h	Priority 32. A packet TOS of 0d32 is mapped to this received packet priority.

15.5.6.33 CPSW_PORT_P1_RX_DSCP_PRI_MAP5 Register (offset = 144h) [reset = 0h]

CPSW_PORT_P1_RX_DSCP_PRI_MAP5 is shown in Figure 15-152 and described in Table 15-166.

CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 5

Figure 15-152. CPSW_PORT_P1_RX_DSCP_PRI_MAP5 Register

31	30	29	28	27	26	25	24
RESERVED	PRI47			RESERVED	PRI46		
R-X	R/W-X			R-X	R/W-X		
23	22	21	20	19	18	17	16
RESERVED	PRI45			RESERVED	PRI44		
R-X	R/W-X			R-X	R/W-X		
15	14	13	12	11	10	9	8
RESERVED	PRI43			RESERVED	PRI42		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI41			RESERVED	PRI40		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-166. CPSW_PORT_P1_RX_DSCP_PRI_MAP5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI47	R/W	X	Priority 47. A packet TOS of 0d47 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI46	R/W	X	Priority 46. A packet TOS of 0d46 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI45	R/W	X	Priority 45. A packet TOS of 0d45 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI44	R/W	X	Priority 44. A packet TOS of 0d44 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI43	R/W	0h	Priority 43. A packet TOS of 0d43 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI42	R/W	0h	Priority 42. A packet TOS of 0d42 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI41	R/W	0h	Priority 41. A packet TOS of 0d41 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI40	R/W	0h	Priority 40. A packet TOS of 0d40 is mapped to this received packet priority.

15.5.6.34 CPSW_PORT_P1_RX_DSCP_PRI_MAP6 Register (offset = 148h) [reset = 0h]

CPSW_PORT_P1_RX_DSCP_PRI_MAP6 is shown in Figure 15-153 and described in Table 15-167.

CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 6

Figure 15-153. CPSW_PORT_P1_RX_DSCP_PRI_MAP6 Register

31	30	29	28	27	26	25	24
RESERVED	PRI55			RESERVED	PRI54		
R-X	R/W-X			R-X	R/W-X		
23	22	21	20	19	18	17	16
RESERVED	PRI53			RESERVED	PRI52		
R-X	R/W-X			R-X	R/W-X		
15	14	13	12	11	10	9	8
RESERVED	PRI51			RESERVED	PRI50		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI49			RESERVED	PRI48		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-167. CPSW_PORT_P1_RX_DSCP_PRI_MAP6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI55	R/W	X	Priority 55. A packet TOS of 0d55 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI54	R/W	X	Priority 54. A packet TOS of 0d54 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI53	R/W	X	Priority 53. A packet TOS of 0d53 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI52	R/W	X	Priority 52. A packet TOS of 0d52 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI51	R/W	0h	Priority 51. A packet TOS of 0d51 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI50	R/W	0h	Priority 50. A packet TOS of 0d50 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI49	R/W	0h	Priority 49. A packet TOS of 0d49 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI48	R/W	0h	Priority 48. A packet TOS of 0d48 is mapped to this received packet priority.

15.5.6.35 CPSW_PORT_P1_RX_DSCP_PRI_MAP7 Register (offset = 14Ch) [reset = 0h]

CPSW_PORT_P1_RX_DSCP_PRI_MAP7 is shown in Figure 15-154 and described in Table 15-168.

CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 7

Figure 15-154. CPSW_PORT_P1_RX_DSCP_PRI_MAP7 Register

31	30	29	28	27	26	25	24
RESERVED	PRI63			RESERVED	PRI62		
R-X	R/W-X			R-X	R/W-X		
23	22	21	20	19	18	17	16
RESERVED	PRI61			RESERVED	PRI60		
R-X	R/W-X			R-X	R/W-X		
15	14	13	12	11	10	9	8
RESERVED	PRI59			RESERVED	PRI58		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI57			RESERVED	PRI56		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-168. CPSW_PORT_P1_RX_DSCP_PRI_MAP7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI63	R/W	X	Priority 63. A packet TOS of 0d63 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI62	R/W	X	Priority 62. A packet TOS of 0d62 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI61	R/W	X	Priority 61. A packet TOS of 0d61 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI60	R/W	X	Priority 60. A packet TOS of 0d60 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI59	R/W	0h	Priority 59. A packet TOS of 0d59 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI58	R/W	0h	Priority 58. A packet TOS of 0d58 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI57	R/W	0h	Priority 57. A packet TOS of 0d57 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI56	R/W	0h	Priority 56. A packet TOS of 0d56 is mapped to this received packet priority.

15.5.6.36 CPSW_PORT_P2_CTRL Register (offset = 200h) [reset = 0h]

CPSW_PORT_P2_CTRL is shown in Figure 15-155 and described in Table 15-169.

CPSW_3GF PORT 2 CONTROL REGISTER

Figure 15-155. CPSW_PORT_P2_CTRL Register

31	30	29	28	27	26	25	24
RESERVED						P2_TX_CLKSTOP_EN	P2_PASS_PRI_TAGGED
R-X						R/W-X	R/W-X

23	22	21	20	19	18	17	16
RESERVED		P2_VLAN_LTYPE2_EN	P2_VLAN_LTYPE1_EN	RESERVED			P2_DSCP_PRI_EN
R-X		R/W-X	R/W-X	R-X			R/W-X

15	14	13	12	11	10	9	8
P2_TS_107	P2_TS_320	P2_TS_319	P2_TS_132	P2_TS_131	P2_TS_130	P2_TS_129	P2_TS_TTL_NONZERO
0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
P2_TS_UNI_EN	P2_TS_ANNEX_F_EN	P2_TS_ANNEX_E_EN	P2_TS_ANNEX_D_EN	P2_TS_LTYPE2_EN	P2_TS_LTYPE1_EN	P2_TS_TX_EN	P2_TS_RX_EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-169. CPSW_PORT_P2_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	X	
25	P2_TX_CLKSTOP_EN	R/W	X	Port 2 Transmit clockstop enable 0h (R/W) = RGMII transmit clockstop not enabled 1h (R/W) = RGMII transmit clockstop enabled. The transmit clock will be stopped after the LPI state is entered (and indicated to the CPRGMII) and the P2_Idle2LPI time is counted (counter value reused). The P2_Idle2LPI counter value must be greater than 9 transmit clocks (slowest clock).
24	P2_PASS_PRI_TAGGED	R/W	X	Port 2 Pass Priority Tagged 0h (R/W) = Priority tagged packets have the zero VID replaced with the input port P2_PORT_VLAN[11 to 0] 1h (R/W) = Priority tagged packets are processed unchanged.
23-22	RESERVED	R	X	
21	P2_VLAN_LTYPE2_EN	R/W	X	Port 2 VLAN LTYPE 2 enable 0h (R/W) = Disabled 1h (R/W) = VLAN LTYPE2 enabled on transmit and receive
20	P2_VLAN_LTYPE1_EN	R/W	X	Port 2 VLAN LTYPE 1 enable 0h (R/W) = Disabled 1h (R/W) = VLAN LTYPE1 enabled on transmit and receive
19-17	RESERVED	R	X	
16	P2_DSCP_PRI_EN	R/W	X	Port 0 DSCP Priority Enable. All non-tagged IPV4 packets have their received packet priority determined by mapping the 6 TOS bits through the port DSCP priority mapping registers. 0h (R/W) = DSCP priority disabled 1h (R/W) = DSCP priority enabled
15	P2_TS_107		0h	Port 2 Time Sync Destination IP Address 107 enable 0h (R/W) = Disabled 1h (R/W) = Destination IP address (dec) 224.0.0.107 is enabled.

Table 15-169. CPSW_PORT_P2_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	P2_TS_320	R/W	0h	Port 2 Time Sync Destination Port Number 320 enable 0h (R/W) = Disabled 1h (R/W) = Annex D (UDP/IPv4) time sync packet destination port number 320 (decimal) is enabled.
13	P2_TS_319	R/W	0h	Port 2 Time Sync Destination Port Number 319 enable 0h (R/W) = DSCP priority disabled 1h (R/W) = Annex D (UDP/IPv4) time sync packet destination port number 319 (decimal) is enabled.
12	P2_TS_132	R/W	0h	Port 2 Time Sync Destination IP Address 132 enable 0h (R/W) = DSCP priority disabled 1h (R/W) = Annex D (UDP/IPv4) time sync packet destination IP address number 132 (decimal) is enabled.
11	P2_TS_131	R/W	0h	Port 2 Time Sync Destination IP Address 131 enable 0h (R/W) = DSCP priority disabled 1h (R/W) = Annex D (UDP/IPv4) time sync packet destination IP address number 131 (decimal) is enabled.
10	P2_TS_130	R/W	0h	Port 2 Time Sync Destination IP Address 130 enable 0h (R/W) = DSCP priority disabled 1h (R/W) = Annex D (UDP/IPv4) time sync packet destination IP address number 130 (decimal) is enabled.
9	P2_TS_129	R/W	0h	Port 2 Time Sync Destination IP Address 129 enable 0h (R/W) = DSCP priority disabled 1h (R/W) = Annex D (UDP/IPv4) time sync packet destination IP address number 129 (decimal) is enabled.
8	P2_TS_TTL_NONZERO	R/W	0h	Port 2 Time Sync Time To Live Non-zero enable. 0h (R/W) = TTL must be zero 1h (R/W) = TTL may be any value
7	P2_TS_UNI_EN	R/W	0h	Port 2 Time Sync Unicast Enable 0h (R/W) = Unicast disabled 1h (R/W) = Unicast enabled
6	P2_TS_ANNEX_F_EN	R/W	0h	Port 2 Time Sync Annex F enable 0h (R/W) = Annex F disabled 1h (R/W) = Annex F enabled
5	P2_TS_ANNEX_E_EN	R/W	0h	Port 2 Time Sync Annex E enable 0h (R/W) = Annex E disabled 1h (R/W) = Annex E enabled
4	P2_TS_ANNEX_D_EN	R/W	0h	Port 2 Time Sync Annex D enable 0h (R/W) = Annex D disabled 1h (R/W) = Annex D enabled
3	P2_TS_LTYPE2_EN	R	0h	Port 2 Time Sync LTYPE 2 enable 0h (R/W) = Disabled 1h (R/W) = Enabled
2	P2_TS_LTYPE1_EN	R/W	0h	Port 2 Time Sync LTYPE 1 enable 0h (R/W) = Disabled 1h (R/W) = Enabled
1	P2_TS_TX_EN	R/W	0h	Port 2 Time Sync Transmit Enable 0h (R/W) = Disabled 1h (R/W) = Enabled
0	P2_TS_RX_EN	R/W	0h	Port 2 Time Sync Receive Enable 0h (R/W) = Port 1 Receive Time Sync disabled 1h (R/W) = Port 1 Receive Time Sync enabled

15.5.6.37 CPSW_PORT_P2_TS_CTL2 Register (offset = 204h) [reset = 40000h]

CPSW_PORT_P2_TS_CTL2 is shown in Figure 15-156 and described in Table 15-170.

Figure 15-156. CPSW_PORT_P2_TS_CTL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED										P2_DOMAIN_OFFSET					
R-0h										R/W-4h					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
P2_TS_MCAST_TYPE_EN															
R/W-0h															

Table 15-170. CPSW_PORT_P2_TS_CTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-16	P2_DOMAIN_OFFSET	R/W	4h	Domain offset value
15-0	P2_TS_MCAST_TYPE_EN	R/W	0h	Multicast Type Enable

15.5.6.38 CPSW_PORT_P2_MAX_BLKS Register (offset = 208h) [reset = 113h]

CPSW_PORT_P2_MAX_BLKS is shown in Figure 15-157 and described in Table 15-171.

CPSW PORT 2 MAXIMUM FIFO BLOCKS REGISTER

Figure 15-157. CPSW_PORT_P2_MAX_BLKS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							P2_TX_MAX_BLKS
R-0h							R/W-11h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
P2_TX_MAX_BLKS				P2_RX_MAX_BLKS			
R/W-11h				R/W-3h			

Table 15-171. CPSW_PORT_P2_MAX_BLKS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	P2_TX_MAX_BLKS	R/W	11h	Transmit FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical transmit priority queues. 0x11 is the recommended value of p2_tx_max_blks unless the port is in fullduplex flow control mode. In flow control mode, the p2_rx_max_blks will need to increase in order to accept the required run out in fullduplex mode. This value will need to decrease by the amount of increase in p2_rx_max_blks. 0xe is the minimum value tx max blks.
3-0	P2_RX_MAX_BLKS	R/W	3h	Receive FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical receive queue. This value must be greater than or equal to 0x3. It should be increased In fullduplex flow control mode to 0x5 or 0x6 depending on the required runout space. The p2_tx_max_blks value must be decreased by the amount of increase in p2_rx_max_blks. 0x3 is the minimum value rx max blks and 0x6 is the maximum value.

15.5.6.39 CPSW_PORT_P2_BLK_CNT Register (offset = 20Ch) [reset = 41h]

CPSW_PORT_P2_BLK_CNT is shown in Figure 15-158 and described in Table 15-172.

CPSW PORT 2 FIFO BLOCK USAGE COUNT (READ ONLY)

Figure 15-158. CPSW_PORT_P2_BLK_CNT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>P2_TX_BLK_CNT</td></tr>
<tr><td colspan="7">R-0h</td><td>R-4h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">P2_TX_BLK_CNT</td><td colspan="4">P2_RX_BLK_CNT</td></tr>
<tr><td colspan="4">R-4h</td><td colspan="4">R-1h</td></tr>
</table>

Table 15-172. CPSW_PORT_P2_BLK_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	P2_TX_BLK_CNT	R	4h	Port 2 Transmit Block Count Usage - This value is the number of blocks allocated to the FIFO logical transmit queues.
3-0	P2_RX_BLK_CNT	R	1h	Port 2 Receive Block Count Usage - This value is the number of blocks allocated to the FIFO logical receive queues.

15.5.6.40 CPSW_PORT_P2_TX_IN_CTL Register (offset = 210h) [reset = 80040C0h]

CPSW_PORT_P2_TX_IN_CTL is shown in Figure 15-159 and described in Table 15-173.

CPSW PORT 2 TRANSMIT FIFO CONTROL

Figure 15-159. CPSW_PORT_P2_TX_IN_CTL Register

31	30	29	28	27	26	25	24
RESERVED				HOST_BLKS_REM			
R-0h				R/W-8h			
23	22	21	20	19	18	17	16
TX_RATE_EN				RESERVED		TX_IN_SEL	
R/W-0h				R-0h		R/W-0h	
15	14	13	12	11	10	9	8
TX_BLKS_REM				RESERVED		TX_PRI_WDS	
R/W-4h				R-0h		R/W-C0h	
7	6	5	4	3	2	1	0
TX_PRI_WDS							
R/W-C0h							

Table 15-173. CPSW_PORT_P2_TX_IN_CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-24	HOST_BLKS_REM	R/W	8h	Transmit FIFO Blocks that must be free before a non rate-limited CPDMA channel can begin sending a packet to the FIFO.
23-20	TX_RATE_EN	R/W	0h	Transmit FIFO Input Rate Enable
19-18	RESERVED	R	0h	
17-16	TX_IN_SEL	R/W	0h	Transmit FIFO Input Queue Type Select 0h (R/W) = Normal priority mode 1h (R/W) = Reserved 2h (R/W) = Rate Limit mode 3h (R/W) = Reserved
15-12	TX_BLKS_REM	R/W	4h	Transmit FIFO Input Blocks to subtract in dual mac mode and blocks to subtract on non rate-limited traffic in rate-limit mode.
11-10	RESERVED	R	0h	
9-0	TX_PRI_WDS	R/W	C0h	Transmit FIFO Words in queue

15.5.6.41 CPSW_PORT_P2_VLAN Register (offset = 214h) [reset = 0h]

CPSW_PORT_P2_VLAN is shown in Figure 15-160 and described in Table 15-174.

CPSW PORT 2 VLAN REGISTER

Figure 15-160. CPSW_PORT_P2_VLAN Register

15	14	13	12	11	10	9	8
PORT_PRI			PORT_CFI	PORT_VID			
R/W-0h			R/W-0h	R/W-0h			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PORT_VID							
R/W-0h							

Table 15-174. CPSW_PORT_P2_VLAN Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	PORT_PRI	R/W	0h	Port VLAN Priority (7 is highest priority)
12	PORT_CFI	R/W	0h	Port CFI bit
11-0	PORT_VID	R/W	0h	Port VLAN ID

15.5.6.42 CPSW_PORT_P2_TX_PRI_MAP Register (offset = 218h) [reset = 33221001h]

CPSW_PORT_P2_TX_PRI_MAP is shown in Figure 15-161 and described in Table 15-175.

CPSW PORT 2 TX HEADER PRIORITY TO SWITCH PRI MAPPING REGISTER

Figure 15-161. CPSW_PORT_P2_TX_PRI_MAP Register

31	30	29	28	27	26	25	24
RESERVED		PRI7		RESERVED		PRI6	
R-X		R/W-X		R-X		R/W-X	
23	22	21	20	19	18	17	16
RESERVED		PRI5		RESERVED		PRI4	
R-X		R/W-X		R-X		R/W-X	
15	14	13	12	11	10	9	8
RESERVED		PRI3		RESERVED		PRI2	
R-0h		R/W-1h		R-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED		PRI1		RESERVED		PRI0	
R-0h		R/W-0h		R-0h		R/W-1h	

Table 15-175. CPSW_PORT_P2_TX_PRI_MAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	X	
29-28	PRI7	R/W	X	Priority 7 - A packet header priority of 0x7 is given this switch queue pri.
27-26	RESERVED	R	X	
25-24	PRI6	R/W	X	Priority 6 - A packet header priority of 0x6 is given this switch queue pri.
23-22	RESERVED	R	X	
21-20	PRI5	R/W	X	Priority 5 - A packet header priority of 0x5 is given this switch queue pri.
19-18	RESERVED	R	X	
17-16	PRI4	R/W	X	Priority 4 - A packet header priority of 0x4 is given this switch queue pri.
15-14	RESERVED	R	0h	
13-12	PRI3	R/W	1h	Priority 3 - A packet header priority of 0x3 is given this switch queue pri.
11-10	RESERVED	R	0h	
9-8	PRI2	R/W	0h	Priority 2 - A packet header priority of 0x2 is given this switch queue pri.
7-6	RESERVED	R	0h	
5-4	PRI1	R/W	0h	Priority 1 - A packet header priority of 0x1 is given this switch queue pri.
3-2	RESERVED	R	0h	
1-0	PRI0	R/W	1h	Priority 0 - A packet header priority of 0x0 is given this switch queue pri.

15.5.6.43 CPSW_PORT_P2_TS_SEQ_MTYPE Register (offset = 21Ch) [reset = 1E0000h]

CPSW_PORT_P2_TS_SEQ_MTYPE is shown in Figure 15-162 and described in Table 15-176.

CPSW_3GF PORT 2 TIME SYNC SEQUENCE ID OFFSET AND MSG TYPE.

Figure 15-162. CPSW_PORT_P2_TS_SEQ_MTYPE Register

31	30	29	28	27	26	25	24
RESERVED							
R-X							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED		P2_TS_SEQ_ID_OFFSET					
R-X		R/W-X					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
P2_TS_MSG_TYPE_EN							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
P2_TS_MSG_TYPE_EN							
R/W-0h							

Table 15-176. CPSW_PORT_P2_TS_SEQ_MTYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	X	
21-16	P2_TS_SEQ_ID_OFFSET	R/W	X	Port 2 Time Sync Sequence ID Offset This is the number of octets that the sequence ID is offset in the tx and rx time sync message header. The minimum value is 6.
15-0	P2_TS_MSG_TYPE_EN	R/W	0h	Port 2 Time Sync Message Type Enable - Each bit in this field enables the corresponding message type in receive and transmit time sync messages (Bit 0 enables message type 0 etc.).

15.5.6.44 CPSW_PORT_P2_SA_LO Register (offset = 220h) [reset = 0h]

CPSW_PORT_P2_SA_LO is shown in Figure 15-163 and described in Table 15-177.

CPSW CPGMAC_SL2 SOURCE ADDRESS LOW REGISTER

Figure 15-163. CPSW_PORT_P2_SA_LO Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-X</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">MACSRCADDR_7_0</td><td colspan="8">MACSRCADDR_15_8</td></tr>
<tr><td colspan="8">R/W-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 15-177. CPSW_PORT_P2_SA_LO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	X	
15-8	MACSRCADDR_7_0	R/W	0h	Source Address Lower 8 bits (byte 0)
7-0	MACSRCADDR_15_8	R/W	0h	Source Address bits 15 to 8 (byte 1)

15.5.6.45 CPSW_PORT_P2_SA_HI Register (offset = 224h) [reset = 0h]

CPSW_PORT_P2_SA_HI is shown in Figure 15-164 and described in Table 15-178.

CPSW CPGMAC_SL2 SOURCE ADDRESS HIGH REGISTER

Figure 15-164. CPSW_PORT_P2_SA_HI Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MACSRCADDR_23_16								MACSRCADDR_31_23							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACSRCADDR_39_32								MACSRCADDR_47_40							
R/W-0h								R/W-0h							

Table 15-178. CPSW_PORT_P2_SA_HI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MACSRCADDR_23_16	R/W	0h	Source Address bits 23 to 16 (byte 2)
23-16	MACSRCADDR_31_23	R/W	0h	Source Address bits 31 to 23 (byte 3)
15-8	MACSRCADDR_39_32	R/W	0h	Source Address bits 39 to 32 (byte 4)
7-0	MACSRCADDR_47_40	R/W	0h	Source Address bits 47 to 40 (byte 5)

15.5.6.46 CPSW_PORT_P2_SEND_PERCENT Register (offset = 228h) [reset = 0h]

CPSW_PORT_P2_SEND_PERCENT is shown in Figure 15-165 and described in Table 15-179.

CPSW PORT 2 TRANSMIT QUEUE SEND PERCENTAGES

Figure 15-165. CPSW_PORT_P2_SEND_PERCENT Register

31	30	29	28	27	26	25	24
RESERVED							
R-X							
23	22	21	20	19	18	17	16
RESERVED	PRI3_SEND_PERCENT						
R-X	R/W-X						
15	14	13	12	11	10	9	8
RESERVED	PRI2_SEND_PERCENT						
R-0h	R/W-0h						
7	6	5	4	3	2	1	0
RESERVED	PRI1_SEND_PERCENT						
R-0h	R/W-0h						

Table 15-179. CPSW_PORT_P2_SEND_PERCENT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	X	
22-16	PRI3_SEND_PERCENT	R/W	X	Priority 3 Transmit Percentage - This percentage value is sent from FIFO priority 3 (maximum) when the p1_pri3_shape_en is set (queue shaping enabled). This is the percentage of the wire that packets from priority 3 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).
15	RESERVED	R	0h	
14-8	PRI2_SEND_PERCENT	R/W	0h	Priority 2 Transmit Percentage - This percentage value is sent from FIFO priority 2 (maximum) when the p1_pri2_shape_en is set (queue shaping enabled). This is the percentage of the wire that packets from priority 2 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).
7	RESERVED	R	0h	
6-0	PRI1_SEND_PERCENT	R/W	0h	Priority 1 Transmit Percentage - This percentage value is sent from FIFO priority 1 (maximum) when the p1_pri1_shape_en is set (queue shaping enabled). This is the percentage of the wire that packets from priority 1 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).

15.5.6.47 CPSW_PORT_P2_RX_DSCP_PRI_MAP0 Register (offset = 230h) [reset = 0h]

CPSW_PORT_P2_RX_DSCP_PRI_MAP0 is shown in Figure 15-166 and described in Table 15-180.

CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 0

Figure 15-166. CPSW_PORT_P2_RX_DSCP_PRI_MAP0 Register

31	30	29	28	27	26	25	24
RESERVED	PRI7			RESERVED	PRI6		
R-X	R/W-X			R-X	R/W-X		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED	PRI5			RESERVED	PRI4		
R-X	R/W-X			R-X	R/W-X		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED	PRI3			RESERVED	PRI2		
R-0h	R/W-0h			R-0h	R/W-0h		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	PRI1			RESERVED	PRI0		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-180. CPSW_PORT_P2_RX_DSCP_PRI_MAP0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI7	R/W	X	Priority 7. A packet TOS of 0d7 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI6	R/W	X	Priority 6. A packet TOS of 0d6 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI5	R/W	X	Priority 5. A packet TOS of 0d5 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI4	R/W	X	Priority 4. A packet TOS of 0d4 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI3	R/W	0h	Priority 3. A packet TOS of 0d3 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI2	R/W	0h	Priority 2. A packet TOS of 0d2 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI1	R/W	0h	Priority 1. A packet TOS of 0d1 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	Priority 0. A packet TOS of 0d0 is mapped to this received packet priority.

15.5.6.48 CPSW_PORT_P2_RX_DSCP_PRI_MAP1 Register (offset = 234h) [reset = 0h]

CPSW_PORT_P2_RX_DSCP_PRI_MAP1 is shown in Figure 15-167 and described in Table 15-181.

CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 1

Figure 15-167. CPSW_PORT_P2_RX_DSCP_PRI_MAP1 Register

31	30	29	28	27	26	25	24
RESERVED	PRI15			RESERVED	PRI14		
R-X	R/W-X			R-X	R/W-X		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED	PRI13			RESERVED	PRI12		
R-X	R/W-X			R-X	R/W-X		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED	PRI11			RESERVED	PRI10		
R-0h	R/W-0h			R-0h	R/W-0h		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	PRI9			RESERVED	PRI8		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-181. CPSW_PORT_P2_RX_DSCP_PRI_MAP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI15	R/W	X	Priority 15. A packet TOS of 0d15 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI14	R/W	X	Priority 14. A packet TOS of 0d14 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI13	R/W	X	Priority 13. A packet TOS of 0d13 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI12	R/W	X	Priority 12. A packet TOS of 0d12 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI11	R/W	0h	Priority 11. A packet TOS of 0d11 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI10	R/W	0h	Priority 10. A packet TOS of 0d10 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI9	R/W	0h	Priority 9. A packet TOS of 0d9 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI8	R/W	0h	Priority 8. A packet TOS of 0d8 is mapped to this received packet priority.

15.5.6.49 CPSW_PORT_P2_RX_DSCP_PRI_MAP2 Register (offset = 238h) [reset = 0h]

CPSW_PORT_P2_RX_DSCP_PRI_MAP2 is shown in Figure 15-168 and described in Table 15-182.

CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 2

Figure 15-168. CPSW_PORT_P2_RX_DSCP_PRI_MAP2 Register

31	30	29	28	27	26	25	24
RESERVED	PRI23			RESERVED	PRI22		
R-X	R/W-X			R-X	R/W-X		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED	PRI21			RESERVED	PRI20		
R-X	R/W-X			R-X	R/W-X		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED	PRI19			RESERVED	PRI18		
R-0h	R/W-0h			R-0h	R/W-0h		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	PRI17			RESERVED	PRI16		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-182. CPSW_PORT_P2_RX_DSCP_PRI_MAP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI23	R/W	X	Priority 23. A packet TOS of 0d23 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI22	R/W	X	Priority 22. A packet TOS of 0d22 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI21	R/W	X	Priority 21. A packet TOS of 0d21 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI20	R/W	X	Priority 20. A packet TOS of 0d20 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI19	R/W	0h	Priority 19. A packet TOS of 0d19 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI18	R/W	0h	Priority 18. A packet TOS of 0d18 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI17	R/W	0h	Priority 17. A packet TOS of 0d17 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI16	R/W	0h	Priority 16. A packet TOS of 0d16 is mapped to this received packet priority.

15.5.6.50 CPSW_PORT_P2_RX_DSCP_PRI_MAP3 Register (offset = 23Ch) [reset = 0h]

CPSW_PORT_P2_RX_DSCP_PRI_MAP3 is shown in Figure 15-169 and described in Table 15-183.

CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 3

Figure 15-169. CPSW_PORT_P2_RX_DSCP_PRI_MAP3 Register

31	30	29	28	27	26	25	24
RESERVED	PRI31			RESERVED	PRI30		
R-X	R/W-X			R-X	R/W-X		
23	22	21	20	19	18	17	16
RESERVED	PRI29			RESERVED	PRI28		
R-X	R/W-X			R-X	R/W-X		
15	14	13	12	11	10	9	8
RESERVED	PRI27			RESERVED	PRI26		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI25			RESERVED	PRI24		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-183. CPSW_PORT_P2_RX_DSCP_PRI_MAP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI31	R/W	X	Priority 31. A packet TOS of 0d31 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI30	R/W	X	Priority 30. A packet TOS of 0d30 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI29	R/W	X	Priority 29. A packet TOS of 0d39 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI28	R/W	X	Priority 28. A packet TOS of 0d28 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI27	R/W	0h	Priority 27. A packet TOS of 0d27 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI26	R/W	0h	Priority 26. A packet TOS of 0d26 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI25	R/W	0h	Priority 25. A packet TOS of 0d25 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI24	R/W	0h	Priority 24. A packet TOS of 0d24 is mapped to this received packet priority.

15.5.6.51 CPSW_PORT_P2_RX_DSCP_PRI_MAP4 Register (offset = 240h) [reset = 0h]

CPSW_PORT_P2_RX_DSCP_PRI_MAP4 is shown in Figure 15-170 and described in Table 15-184.

CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 4

Figure 15-170. CPSW_PORT_P2_RX_DSCP_PRI_MAP4 Register

31	30	29	28	27	26	25	24
RESERVED	PRI39			RESERVED	PRI38		
R-X	R/W-X			R-X	R/W-X		
23	22	21	20	19	18	17	16
RESERVED	PRI37			RESERVED	PRI36		
R-X	R/W-X			R-X	R/W-X		
15	14	13	12	11	10	9	8
RESERVED	PRI35			RESERVED	PRI34		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI33			RESERVED	PRI32		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-184. CPSW_PORT_P2_RX_DSCP_PRI_MAP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI39	R/W	X	Priority 39. A packet TOS of 0d39 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI38	R/W	X	Priority 38. A packet TOS of 0d38 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI37	R/W	X	Priority 37. A packet TOS of 0d37 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI36	R/W	X	Priority 36. A packet TOS of 0d36 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI35	R/W	0h	Priority 35. A packet TOS of 0d35 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI34	R/W	0h	Priority 34. A packet TOS of 0d34 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI33	R/W	0h	Priority 33. A packet TOS of 0d33 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI32	R/W	0h	Priority 32. A packet TOS of 0d32 is mapped to this received packet priority.

15.5.6.52 CPSW_PORT_P2_RX_DSCP_PRI_MAP5 Register (offset = 244h) [reset = 0h]

CPSW_PORT_P2_RX_DSCP_PRI_MAP5 is shown in Figure 15-171 and described in Table 15-185.

CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 5

Figure 15-171. CPSW_PORT_P2_RX_DSCP_PRI_MAP5 Register

31	30	29	28	27	26	25	24
RESERVED	PRI47			RESERVED	PRI46		
R-X	R/W-X			R-X	R/W-X		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED	PRI45			RESERVED	PRI44		
R-X	R/W-X			R-X	R/W-X		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED	PRI43			RESERVED	PRI42		
R-0h	R/W-0h			R-0h	R/W-0h		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	PRI41			RESERVED	PRI40		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-185. CPSW_PORT_P2_RX_DSCP_PRI_MAP5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	X	
30-28	PRI47	R/W	X	Priority 47. A packet TOS of 0d47 is mapped to this received packet priority.
27	RESERVED	R	X	
26-24	PRI46	R/W	X	Priority 46. A packet TOS of 0d46 is mapped to this received packet priority.
23	RESERVED	R	X	
22-20	PRI45	R/W	X	Priority 45. A packet TOS of 0d45 is mapped to this received packet priority.
19	RESERVED	R	X	
18-16	PRI44	R/W	X	Priority 44. A packet TOS of 0d44 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI43	R/W	0h	Priority 43. A packet TOS of 0d43 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI42	R/W	0h	Priority 42. A packet TOS of 0d42 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI41	R/W	0h	Priority 41. A packet TOS of 0d41 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI40	R/W	0h	Priority 40. A packet TOS of 0d40 is mapped to this received packet priority.

15.5.6.53 CPSW_PORT_P2_RX_DSCP_PRI_MAP6 Register (offset = 248h) [reset = 0h]

CPSW_PORT_P2_RX_DSCP_PRI_MAP6 is shown in Figure 15-172 and described in Table 15-186.

CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 6

Figure 15-172. CPSW_PORT_P2_RX_DSCP_PRI_MAP6 Register

31	30	29	28	27	26	25	24
RESERVED	PRI55			RESERVED	PRI54		
R-0h	R/W-0h			R-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	PRI53			RESERVED	PRI52		
R-0h	R/W-0h			R-0h	R/W-0h		
15	14	13	12	11	10	9	8
RESERVED	PRI51			RESERVED	PRI50		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI49			RESERVED	PRI48		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-186. CPSW_PORT_P2_RX_DSCP_PRI_MAP6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI55	R/W	0h	Priority 55. A packet TOS of 0d55 is mapped to this received packet priority.
27	RESERVED	R	0h	
26-24	PRI54	R/W	0h	Priority 54. A packet TOS of 0d54 is mapped to this received packet priority.
23	RESERVED	R	0h	
22-20	PRI53	R/W	0h	Priority 53. A packet TOS of 0d53 is mapped to this received packet priority.
19	RESERVED	R	0h	
18-16	PRI52	R/W	0h	Priority 52. A packet TOS of 0d52 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI51	R/W	0h	Priority 51. A packet TOS of 0d51 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI50	R/W	0h	Priority 50. A packet TOS of 0d50 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI49	R/W	0h	Priority 49. A packet TOS of 0d49 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI48	R/W	0h	Priority 48. A packet TOS of 0d48 is mapped to this received packet priority.

15.5.6.54 CPSW_PORT_P2_RX_DSCP_PRI_MAP7 Register (offset = 24Ch) [reset = 0h]

CPSW_PORT_P2_RX_DSCP_PRI_MAP7 is shown in Figure 15-173 and described in Table 15-187.

CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 7

Figure 15-173. CPSW_PORT_P2_RX_DSCP_PRI_MAP7 Register

31	30	29	28	27	26	25	24
RESERVED	PRI63			RESERVED	PRI62		
R-0h	R/W-0h			R-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	PRI61			RESERVED	PRI60		
R-0h	R/W-0h			R-0h	R/W-0h		
15	14	13	12	11	10	9	8
RESERVED	PRI59			RESERVED	PRI58		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI57			RESERVED	PRI56		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 15-187. CPSW_PORT_P2_RX_DSCP_PRI_MAP7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI63	R/W	0h	Priority 63. A packet TOS of 0d63 is mapped to this received packet priority.
27	RESERVED	R	0h	
26-24	PRI62	R/W	0h	Priority 62. A packet TOS of 0d62 is mapped to this received packet priority.
23	RESERVED	R	0h	
22-20	PRI61	R/W	0h	Priority 61. A packet TOS of 0d61 is mapped to this received packet priority.
19	RESERVED	R	0h	
18-16	PRI60	R/W	0h	Priority 60. A packet TOS of 0d60 is mapped to this received packet priority.
15	RESERVED	R	0h	
14-12	PRI59	R/W	0h	Priority 59. A packet TOS of 0d59 is mapped to this received packet priority.
11	RESERVED	R	0h	
10-8	PRI58	R/W	0h	Priority 58. A packet TOS of 0d58 is mapped to this received packet priority.
7	RESERVED	R	0h	
6-4	PRI57	R/W	0h	Priority 57. A packet TOS of 0d57 is mapped to this received packet priority.
3	RESERVED	R	0h	
2-0	PRI56	R/W	0h	Priority 56. A packet TOS of 0d56 is mapped to this received packet priority.

15.5.7 CPSW_SL Registers

Table 15-188 lists the memory-mapped registers for the CPSW_SL. All register offset addresses not listed in Table 15-188 should be considered as reserved locations and the register contents should not be modified.

Table 15-188. CPSW_SL Registers

Offset	Acronym	Register Name	Section
0h	CPSW_SL_IDVER	CPGMAC_SL ID/VERSION REGISTER	Section 15.5.7.1
4h	CPSW_SL_MACCTRL	CPGMAC_SL MAC CONTROL REGISTER	Section 15.5.7.2
8h	CPSW_SL_MACSTS	CPGMAC_SL MAC STATUS REGISTER	Section 15.5.7.3
Ch	CPSW_SL_SOFT_RESET	CPGMAC_SL SOFT RESET REGISTER	Section 15.5.7.4
10h	CPSW_SL_RX_MAXLEN	CPGMAC_SL RX MAXIMUM LENGTH REGISTER	Section 15.5.7.5
14h	CPSW_SL_BOFFTEST	CPGMAC_SL BACKOFF TEST REGISTER	Section 15.5.7.6
18h	CPSW_SL_RX_PAUSE	CPGMAC_SL RECEIVE PAUSE TIMER REGISTER	Section 15.5.7.7
1Ch	CPSW_SL_TX_PAUSE	CPGMAC_SL TRANSMIT PAUSE TIMER REGISTER	Section 15.5.7.8
20h	CPSW_SL_EMCTRL	CPGMAC_SL EMULATION CONTROL REGISTER	Section 15.5.7.9
24h	CPSW_SL_RX_PRI_MAP	CPGMAC_SL RX PKT PRIORITY TO HEADER PRIORITY MAPPING REGISTER	Section 15.5.7.10
28h	CPSW_SL_TX_GAP	TRANSMIT INTER-PACKET GAP REGISTER	Section 15.5.7.11

15.5.7.1 CPSW_SL_IDVER Register (Offset = 0h) [reset = 00170112h]

CPSW_SL_IDVER is shown in Figure 15-174 and described in Table 15-189.

Return to Summary Table.

CPGMAC_SL ID/VERSION REGISTER

Figure 15-174. CPSW_SL_IDVER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">IDENT</td><td colspan="5">Z</td><td colspan="3">X</td><td colspan="8">Y</td></tr>
<tr><td colspan="16">R-17h</td><td colspan="5">R-0h</td><td colspan="3">R-1h</td><td colspan="8">R-12h</td></tr>
</table>

Table 15-189. CPSW_SL_IDVER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	IDENT	R	17h	Rx Identification Value
15-11	Z	R	0h	Rx Z value (X.Y.Z)
10-8	X	R	1h	Rx X value (major)
7-0	Y	R	12h	Rx Y value (minor)

15.5.7.2 CPSW_SL_MACCTRL Register (Offset = 4h) [reset = 0h]

CPSW_SL_MACCTRL is shown in Figure 15-175 and described in Table 15-190.

Return to Summary Table.

CPGMAC_SL MAC CONTROL REGISTER

Figure 15-175. CPSW_SL_MACCTRL Register

31	30	29	28	27	26	25	24
RESERVED							RX_CMF_EN
R-0h							R/W-0h
23	22	21	20	19	18	17	16
RX_CSF_EN	RX_CEF_EN	TX_SHORT_GAP_LIM_EN	RESERVED		EXT_EN	GIG_FORCE	IFCTL_B
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
IFCTL_A	RESERVED		CRS_FLOW_EN	CMD_IDLE	TX_SHORT_GAP_EN	RESERVED	
R/W-0h	R-0h		R/W-0h	R/W-0h	R/W-0h	R-0h	
7	6	5	4	3	2	1	0
GIG	TX_PACE	GMII_EN	TX_FLOW_EN	RX_FLOW_EN	MTEST	LOOPBACK	FULLDUPLEX
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-190. CPSW_SL_MACCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	
24	RX_CMF_EN	R/W	0h	RX Copy MAC Control Frames Enable - Enables MAC control frames to be transferred to memory. MAC control frames are normally acted upon (if enabled), but not copied to memory. MAC control frames that are pause frames will be acted upon if enabled in the MacControl register, regardless of the value of rx_cmf_en. Frames transferred to memory due to rx_cmf_en will have the control bit set in their EOP buffer descriptor. 0h (R/W) = MAC control frames are filtered (but acted upon if enabled). 1h (R/W) = MAC control frames are transferred to memory.
23	RX_CSF_EN	R/W	0h	RX Copy Short Frames Enable - Enables frames or fragments shorter than 64 bytes to be copied to memory. Frames transferred to memory due to rx_csf_en will have the fragment or undersized bit set in their receive footer. Fragments are short frames that contain CRC/align/code errors and undersized are short frames without errors. 0h (R/W) = Short frames are filtered 1h (R/W) = Short frames are transferred to memory.
22	RX_CEF_EN	R/W	0h	RX Copy Error Frames Enable - Enables frames containing errors to be transferred to memory. The appropriate error bit will be set in the frame receive footer. Frames containing errors will be filtered when rx_cef _en is not set. 0h (R/W) = Frames containing errors are filtered. 1h (R/W) = Frames containing errors are transferred to memory.
21	TX_SHORT_GAP_LIM_EN	R/W	0h	Transmit Short Gap Limit Enable When set this bit limits the number of short gap packets transmitted to 100ppm. Each time a short gap packet is sent, a counter is loaded with 10,000 and decremented on each wireside clock. Another short gap packet will not be sent out until the counter decrements to zero. This mode is included to preclude the host from filling up the FIFO and sending every packet out with short gap which would violate the maximum number of packets per second allowed.

Table 15-190. CPSW_SL_MACCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20-19	RESERVED	R	0h	
18	EXT_EN	R/W	0h	Control Enable - Enables the fullduplex and gigabit mode to be selected from the FULLDUPLEX_IN and GIG_IN input signals and not from the fullduplex and gig bits in this register. The FULLDUPLEX_MODE bit reflects the actual fullduplex mode selected.
17	GIG_FORCE	R/W	0h	Gigabit Mode Force - This bit is used to force the CPGMAC_SL into gigabit mode if the input GMII_MTCLK has been stopped by the PHY.
16	IFCTL_B	R/W	0h	Connects to the speed_in input of the respective RMII gasket. When using RMII mode: 0h (R/W) = 10Mbps operation 1h (R/W) = 100Mbps operation
15	IFCTL_A	R/W	0h	Connects to the speed_in input of the respective RMII gasket. When using RMII mode: 0h (R/W) = 10Mbps operation 1h (R/W) = 100Mbps operation
14-13	RESERVED	R	0h	
12	CRS_FLOW_EN	R/W	0h	Carrier Sense Flow Control Enable When set this bit enables the GMII_MCRS (carrier sense) to be used as a hardware flow control in fullduplex mode.
11	CMD_IDLE	R/W	0h	Command Idle 0h (R/W) = Idle not commanded 1h (R/W) = Idle Commanded (read idle in MacStatus)
10	TX_SHORT_GAP_EN	R/W	0h	Transmit Short Gap Enable 0h (R/W) = Transmit with a short IPG is disabled 1h (R/W) = Transmit with a short IPG (when TX_SHORT_GAP input is asserted) is enabled.
9-8	RESERVED	R	0h	
7	GIG	R/W	0h	Gigabit Mode 0h (R/W) = 10/100 mode 1h (R/W) = Gigabit mode (full duplex only) The GIG_OUT output is the value of this bit.
6	TX_PACE	R/W	0h	Transmit Pacing Enable 0h (R/W) = Transmit Pacing Disabled 1h (R/W) = Transmit Pacing Enabled
5	GMII_EN	R/W	0h	GMII Enable. This bit must be set before the MAC will transmit or receive data in any of the supported interface modes. (ex. MII, RMII). This bit does not select the interface mode but rather holds or releases the MAC TX and RX state machines from reset. 0h (R/W) = The MAC RX and TX state machines are held in reset. 1h (R/W) = The MAC RX and TX state machines are released from reset and transmit/receive are enabled.
4	TX_FLOW_EN	R/W	0h	Transmit Flow Control Enable - Determines if incoming pause frames are acted upon in full-duplex mode. Incoming pause frames are not acted upon in half-duplex mode regardless of this bit setting. The RX_MBP_Enable bits determine whether or not received pause frames are transferred to memory. 0h (R/W) = Transmit Flow Control Disabled. Full-duplex mode - Incoming pause frames are not acted upon. 1h (R/W) = Transmit Flow Control Enabled. Full-duplex mode - Incoming pause frames are acted upon.

Table 15-190. CPSW_SL_MACCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	RX_FLOW_EN	R/W	0h	Receive Flow Control Enable 0h (R/W) = Receive Flow Control Disabled Half-duplex mode - No flow control generated collisions are sent. Full-duplex mode - No outgoing pause frames are sent. 1h (R/W) = Receive Flow Control Enabled Half-duplex mode - Collisions are initiated when receive flow control is triggered. Full-duplex mode - Outgoing pause frames are sent when receive flow control is triggered.
2	MTEST	R/W	0h	Manufacturing Test mode - This bit must be set to allow writes to the Backoff_Test and PauseTimer registers.
1	LOOPBACK	R/W	0h	Loop Back Mode - Loopback mode forces internal fullduplex mode regardless of whether the fullduplex bit is set or not. The loopback bit should be changed only when GMII_en is deasserted. 0h (R/W) = Not looped back 1h (R/W) = Loop Back Mode enabled
0	FULLDUPLEX	R/W	0h	Full Duplex mode - Gigabit mode forces fullduplex mode regardless of whether the fullduplex bit is set or not. The FULLDUPLEX_OUT output is the value of this register bit 0h (R/W) = Half duplex mode 1h (R/W) = Full duplex mode

15.5.7.3 CPSW_SL_MACSTS Register (Offset = 8h) [reset = 80000000h]

CPSW_SL_MACSTS is shown in Figure 15-176 and described in Table 15-191.

Return to Summary Table.

CPGMAC_SL MAC STATUS REGISTER

Figure 15-176. CPSW_SL_MACSTS Register

31	30	29	28	27	26	25	24
IDLE	RESERVED						
R-1h	R-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED			EXT_GIG	EXT_FULLDUPLEX	RESERVED	RX_FLOW_ACT	TX_FLOW_ACT
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h

Table 15-191. CPSW_SL_MACSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IDLE	R	1h	CPGMAC_SL IDLE - The CPGMAC_SL is in the idle state (valid after an idle command) 0h (R/W) = The CPGMAC_SL is not in the idle state. 1h (R/W) = The CPGMAC_SL is in the idle state.
30-5	RESERVED	R	0h	
4	EXT_GIG	R	0h	External GIG - This is the value of the EXT_GIG input bit.
3	EXT_FULLDUPLEX	R	0h	External Fullduplex - This is the value of the EXT_FULLDUPLEX input bit.
2	RESERVED	R	0h	
1	RX_FLOW_ACT	R	0h	Receive Flow Control Active - When asserted, indicates that receive flow control is enabled and triggered.
0	TX_FLOW_ACT	R	0h	Transmit Flow Control Active - When asserted, this bit indicates that the pause time period is being observed for a received pause frame. No new transmissions will begin while this bit is asserted except for the transmission of pause frames. Any transmission in progress when this bit is asserted will complete.

15.5.7.4 CPSW_SL_SOFT_RESET Register (Offset = Ch) [reset = 0h]

CPSW_SL_SOFT_RESET is shown in Figure 15-177 and described in Table 15-192.

Return to Summary Table.

CPGMAC_SL SOFT RESET REGISTER

Figure 15-177. CPSW_SL_SOFT_RESET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							SOFT_RESET
R-0h							R/W-0h

Table 15-192. CPSW_SL_SOFT_RESET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	SOFT_RESET	R/W	0h	Software reset - Writing a one to this bit causes the CPGMAC_SL logic to be reset. After writing a one to this bit, it may be polled to determine if the reset has occurred. If a one is read, the reset has not yet occurred. If a zero is read then reset has occurred.

15.5.7.5 CPSW_SL_RX_MAXLEN Register (Offset = 10h) [reset = 5EEh]

CPSW_SL_RX_MAXLEN is shown in Figure 15-178 and described in Table 15-193.

Return to Summary Table.

CPGMAC_SL RX MAXIMUM LENGTH REGISTER

Figure 15-178. CPSW_SL_RX_MAXLEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		RX_MAXLEN															
R-0h																		R/W-5EEh															

Table 15-193. CPSW_SL_RX_MAXLEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	
13-0	RX_MAXLEN	R/W	5EEh	RX Maximum Frame Length - This field determines the maximum length of a received frame. The reset value is 1518 (dec). Frames with byte counts greater than rx_maxlen are long frames. Long frames with no errors are oversized frames. Long frames with CRC, code, or alignment error are jabber frames. The maximum value is 16,383.

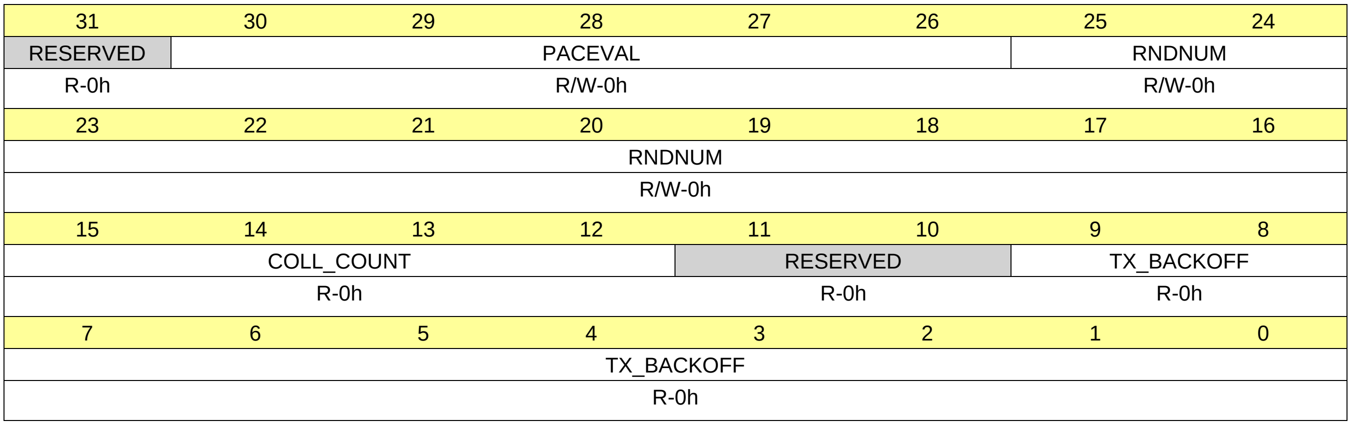
15.5.7.6 CPSW_SL_BOFFTEST Register (Offset = 14h) [reset = 0h]

CPSW_SL_BOFFTEST is shown in Figure 15-179 and described in Table 15-194.

Return to Summary Table.

CPGMAC_SL BACKOFF TEST REGISTER

Figure 15-179. CPSW_SL_BOFFTEST Register



31	30	29	28	27	26	25	24
RESERVED	PACEVAL					RNDNUM	
R-0h	R/W-0h					R/W-0h	
23	22	21	20	19	18	17	16
RNDNUM							
R/W-0h							
15	14	13	12	11	10	9	8
COLL_COUNT				RESERVED		TX_BACKOFF	
R-0h				R-0h		R-0h	
7	6	5	4	3	2	1	0
TX_BACKOFF							
R-0h							

Table 15-194. CPSW_SL_BOFFTEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-26	PACEVAL	R/W	0h	Pacing Register Current Value. A non-zero value in this field indicates that transmit pacing is active. A transmit frame collision or deferral causes paceval to loaded with decimal 31, good frame transmissions (with no collisions or deferrals) cause paceval to be decremented down to zero. When paceval is nonzero, the transmitter delays 4 IPGs between new frame transmissions after each successfully transmitted frame that had no deferrals or collisions. Transmit pacing helps reduce "capture" effects improving overall network bandwidth.
25-16	RNDNUM	R/W	0h	Backoff Random Number Generator - This field allows the Backoff Random Number Generator to be read (or written in test mode only). This field can be written only when mtest has previously been set. Reading this field returns the generator's current value. The value is reset to zero and begins counting on the clock after the deassertion of reset.
15-12	COLL_COUNT	R	0h	Collision Count - The number of collisions the current frame has experienced.
11-10	RESERVED	R	0h	
9-0	TX_BACKOFF	R	0h	Backoff Count - This field allows the current value of the backoff counter to be observed for test purposes. This field is loaded automatically according to the backoff algorithm, and is decremented by one for each slot time after the collision.

15.5.7.7 CPSW_SL_RX_PAUSE Register (Offset = 18h) [reset = 0h]

CPSW_SL_RX_PAUSE is shown in Figure 15-180 and described in Table 15-195.

Return to Summary Table.

CPGMAC_SL RECEIVE PAUSE TIMER REGISTER

Figure 15-180. CPSW_SL_RX_PAUSE Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">RX_PAUSETIMER</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">R-0h</td></tr>
</table>

Table 15-195. CPSW_SL_RX_PAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	RX_PAUSETIMER	R	0h	RX Pause Timer Value - This field allows the contents of the receive pause timer to be observed (and written in test mode). The receive pause timer is loaded with 0xFF00 when the CPGMAC_SL sends an outgoing pause frame (with pause time of 0xFFFF). The receive pause timer is decremented at slot time intervals. If the receive pause timer decrements to zero, then another outgoing pause frame will be sent and the load/decrement process will be repeated.

15.5.7.8 CPSW_SL_TX_PAUSE Register (Offset = 1Ch) [reset = 0h]

CPSW_SL_TX_PAUSE is shown in Figure 15-181 and described in Table 15-196.

Return to Summary Table.

CPGMAC_SL TRANSMIT PAUSE TIMER REGISTER

Figure 15-181. CPSW_SL_TX_PAUSE Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">TX_PAUSETIMER</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">R-0h</td></tr>
</table>

Table 15-196. CPSW_SL_TX_PAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	TX_PAUSETIMER	R	0h	TX Pause Timer Value - This field allows the contents of the transmit pause timer to be observed (and written in test mode). The transmit pause timer is loaded by a received (incoming) pause frame, and then decremented, at slottime intervals, down to zero at which time CPGMAC_SL transmit frames are again enabled.

15.5.7.9 CPSW_SL_EMCTRL Register (Offset = 20h) [reset = 0h]

CPSW_SL_EMCTRL is shown in Figure 15-182 and described in Table 15-197.

Return to Summary Table.

CPGMAC_SL EMULATION CONTROL REGISTER

Figure 15-182. CPSW_SL_EMCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						SOFT	FREE
R-0h						R/W-0h	R/W-0h

Table 15-197. CPSW_SL_EMCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	SOFT	R/W	0h	Emulation Soft Bit
0	FREE	R/W	0h	Emulation Free Bit

15.5.7.10 CPSW_SL_RX_PRI_MAP Register (Offset = 24h) [reset = 76543210h]

CPSW_SL_RX_PRI_MAP is shown in Figure 15-183 and described in Table 15-198.

Return to Summary Table.

CPGMAC_SL RX PKT PRIORITY TO HEADER PRIORITY MAPPING REGISTER

Figure 15-183. CPSW_SL_RX_PRI_MAP Register

31	30	29	28	27	26	25	24
RESERVED	PRI7			RESERVED	PRI6		
R-0h	R/W-7h			R-0h	R/W-6h		
23	22	21	20	19	18	17	16
RESERVED	PRI5			RESERVED	PRI4		
R-0h	R/W-5h			R-0h	R/W-4h		
15	14	13	12	11	10	9	8
RESERVED	PRI3			RESERVED	PRI2		
R-0h	R/W-3h			R-0h	R/W-2h		
7	6	5	4	3	2	1	0
RESERVED	PRI1			RESERVED	PRI0		
R-0h	R/W-1h			R-0h	R/W-0h		

Table 15-198. CPSW_SL_RX_PRI_MAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	7h	Priority 7 - A packet priority of 0x7 is mapped (changed) to this value.
27	RESERVED	R	0h	
26-24	PRI6	R/W	6h	Priority 6 - A packet priority of 0x6 is mapped (changed) to this value.
23	RESERVED	R	0h	
22-20	PRI5	R/W	5h	Priority 5 - A packet priority of 0x5 is mapped (changed) to this value.
19	RESERVED	R	0h	
18-16	PRI4	R/W	4h	Priority 4 - A packet priority of 0x4 is mapped (changed) to this value.
15	RESERVED	R	0h	
14-12	PRI3	R/W	3h	Priority 3 - A packet priority of 0x3 is mapped (changed) to this value.
11	RESERVED	R	0h	
10-8	PRI2	R/W	2h	Priority 2 - A packet priority of 0x2 is mapped (changed) to this value.
7	RESERVED	R	0h	
6-4	PRI1	R/W	1h	Priority 1 - A packet priority of 0x1 is mapped (changed) to this value.
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	Priority 0 - A packet priority of 0x0 is mapped (changed) to this value.

15.5.7.11 CPSW_SL_TX_GAP Register (Offset = 28h) [reset = Ch]

CPSW_SL_TX_GAP is shown in Figure 15-184 and described in Table 15-199.

Return to Summary Table.

TRANSMIT INTER-PACKET GAP REGISTER

Figure 15-184. CPSW_SL_TX_GAP Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="23">RESERVED</td><td colspan="9">TX_GAP</td></tr>
<tr><td colspan="23">R-0h</td><td colspan="9">R/W-Ch</td></tr>
</table>

Table 15-199. CPSW_SL_TX_GAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-0	TX_GAP	R/W	Ch	Transmit Inter-Packet Gap

15.5.8 CPSW_SS Registers

Table 15-200 lists the memory-mapped registers for the CPSW_SS. All register offset addresses not listed in Table 15-200 should be considered as reserved locations and the register contents should not be modified.

Table 15-200. CPSW_SS Registers

Offset	Acronym	Register Name	Section
0h	CPSW_SS_ID_VER	ID VERSION REGISTER	Section 15.5.8.1
4h	CPSW_SS_CTRL	SWITCH CONTROL REGISTER	Section 15.5.8.2
8h	CPSW_SS_SOFT_RESET	SOFT RESET REGISTER	Section 15.5.8.3
Ch	CPSW_SS_STAT_PORT_EN	STATISTICS PORT ENABLE REGISTER	Section 15.5.8.4
10h	CPSW_SS_PTYPE	TRANSMIT PRIORITY TYPE REGISTER	Section 15.5.8.5
14h	CPSW_SS_SOFT_IDLE	SOFTWARE IDLE	Section 15.5.8.6
18h	CPSW_SS_THRU_RATE	THROUGHPUT RATE	Section 15.5.8.7
1Ch	CPSW_SS_GAP_THRESH	CPGMAC_SL SHORT GAP THRESHOLD	Section 15.5.8.8
20h	CPSW_SS_TX_START_WDS	TRANSMIT START WORDS	Section 15.5.8.9
24h	CPSW_SS_FLOW_CTRL	FLOW CONTROL	Section 15.5.8.10
28h	CPSW_SS_VLAN_LTYPE	LTYPE1 AND LTYPE 2 REGISTER	Section 15.5.8.11
2Ch	CPSW_SS_TS_LTYPE	VLAN_LTYPE1 AND VLAN_LTYPE2 REGISTER	Section 15.5.8.12
30h	CPSW_SS_DLR_LTYPE	DLR LTYPE REGISTER	Section 15.5.8.13
34h	CPSW_SS_STS		Section 15.5.8.14

15.5.8.1 CPSW_SS_ID_VER Register (offset = 0h) [reset = 190112h]

CPSW_SS_ID_VER is shown in Figure 15-185 and described in Table 15-201.

ID VERSION REGISTER

Figure 15-185. CPSW_SS_ID_VER Register

31	30	29	28	27	26	25	24
CPSW_3G_IDENT							
R-19h							

23	22	21	20	19	18	17	16
CPSW_3G_IDENT							
R-19h							

15	14	13	12	11	10	9	8
CPSW_3G_RTL_VER					CPSW_3G_MAJ_VER		
R-0h					R-1h		

7	6	5	4	3	2	1	0
CPSW_3G_MINOR_VER							
R-12h							

Table 15-201. CPSW_SS_ID_VER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	CPSW_3G_IDENT	R	19h	3G Identification Value
15-11	CPSW_3G_RTL_VER	R	0h	3G RTL Version Value
10-8	CPSW_3G_MAJ_VER	R	1h	3G Major Version Value
7-0	CPSW_3G_MINOR_VER	R	12h	3G Minor Version Value

15.5.8.2 CPSW_SS_CTRL Register (offset = 4h) [reset = 0h]

CPSW_SS_CTRL is shown in Figure 15-186 and described in Table 15-202.

SWITCH CONTROL REGISTER

Figure 15-186. CPSW_SS_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				DLR_EN	RX_VLAN_ENCAP	VLAN_AWARE	FIFO_LOOPBACK
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-202. CPSW_SS_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	DLR_EN	R/W	0h	DLR enable 0h (R/W) = DLR is disabled. DLR packets will not be moved to queue priority 3 and will not be separated out onto dlr_cpdma_ch. 1h (R/W) = DLR is enabled. DLR packets be moved to destination port transmit queue priority 3 and will be separated out onto dlr_cpdma_ch when packet is to egress on port 0.
2	RX_VLAN_ENCAP	R/W	0h	Port 0 VLAN Encapsulation (egress): 0h (R/W) = Port 2 receive packets (from 3G) are not VLAN encapsulated. 1h (R/W) = Port 2 receive packets (from 3G) are VLAN encapsulated.
1	VLAN_AWARE	R/W	0h	VLAN Aware Mode: 0h (R/W) = 3G is in the VLAN unaware mode. 1h (R/W) = 3G is in the VLAN aware mode.
0	FIFO_LOOPBACK	R/W	0h	FIFO Loopback Mode 0h (R/W) = Loopback is disabled 1h (R/W) = FIFO Loopback mode enabled. Each packet received is turned around and sent out on the same port's transmit path. Port 2 receive is fixed on channel zero. The RXSOFOVERRUN statistic will increment for every packet sent in FIFO loopback mode.

15.5.8.3 CPSW_SS_SOFT_RESET Register (offset = 8h) [reset = 0h]

CPSW_SS_SOFT_RESET is shown in Figure 15-187 and described in Table 15-203.

SOFT RESET REGISTER

Figure 15-187. CPSW_SS_SOFT_RESET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							SOFT_RESET
R-0h							R/W-0h

Table 15-203. CPSW_SS_SOFT_RESET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	SOFT_RESET	R/W	0h	Software reset - Writing a one to this bit causes the 3G logic to be reset. After writing a one to this bit, it may be polled to determine if the reset has occurred. If a one is read, the reset has not yet occurred. If a zero is read then reset has occurred.

15.5.8.4 CPSW_SS_STAT_PORT_EN Register (offset = Ch) [reset = 0h]

CPSW_SS_STAT_PORT_EN is shown in Figure 15-188 and described in Table 15-204.

STATISTICS PORT ENABLE REGISTER

Figure 15-188. CPSW_SS_STAT_PORT_EN Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">RESERVED</td><td>P2_STAT_EN</td><td>P1_STAT_EN</td><td>P0_STAT_EN</td></tr>
<tr><td colspan="5">R-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 15-204. CPSW_SS_STAT_PORT_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	P2_STAT_EN	R/W	0h	Port 2 (GMII2 and Port 2 FIFO) Statistics Enable 0h (R/W) = Port 2 statistics are not enabled. 1h (R/W) = Port 2 statistics are enabled.
1	P1_STAT_EN	R/W	0h	Port 1 (GMII1 and Port 1 FIFO) Statistics Enable 0h (R/W) = Port 1 statistics are not enabled. 1h (R/W) = Port 1 statistics are enabled.
0	P0_STAT_EN	R/W	0h	Port 0 Statistics Enable. FIFO overruns (SOFOVERRUNS) are the only port 0 statistics that are enabled to be kept. 0h (R/W) = Port 0 statistics are not enabled. 1h (R/W) = Port 0 statistics are enabled.

15.5.8.5 CPSW_SS_PTYPE Register (offset = 10h) [reset = 0h]

CPSW_SS_PTYPE is shown in Figure 15-189 and described in Table 15-205.

TRANSMIT PRIORITY TYPE REGISTER

Figure 15-189. CPSW_SS_PTYPE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED		P2_PRI3_SHAPE_EN	P2_PRI2_SHAPE_EN	P2_PRI1_SHAPE_EN	P1_PRI3_SHAPE_EN	P1_PRI2_SHAPE_EN	P1_PRI1_SHAPE_EN
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED					P2_PTYPE_ESC	P1_PTYPE_ESC	P0_PTYPE_ESC
R-0h					R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED			ESC_PRI_LD_VAL				
R-0h			R/W-0h				

Table 15-205. CPSW_SS_PTYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21	P2_PRI3_SHAPE_EN	R/W	0h	Port 2 Queue Priority 3 Transmit Shape Enable - If there is only one shaping queue then it must be priority 3.
20	P2_PRI2_SHAPE_EN	R/W	0h	Port 2 Queue Priority 2 Transmit Shape Enable - If there are two shaping queues then they must be priorities 3 and 2.
19	P2_PRI1_SHAPE_EN	R/W	0h	Port 2 Queue Priority 1 Transmit Shape Enable - If there are three shaping queues all three bits should be set.
18	P1_PRI3_SHAPE_EN	R/W	0h	Port 1 Queue Priority 3 Transmit Shape Enable - If there is only one shaping queue then it must be priority 3.
17	P1_PRI2_SHAPE_EN	R/W	0h	Port 1 Queue Priority 2 Transmit Shape Enable- If there are two shaping queues then they must be priorities 3 and 2.
16	P1_PRI1_SHAPE_EN	R/W	0h	Port 1 Queue Priority 1 Transmit Shape Enable- If there are three shaping queues all three bits should be set.
15-11	RESERVED	R	0h	
10	P2_PTYPE_ESC	R/W	0h	Port 2 Priority Type Escalate 0h (R/W) = Port 2 priority type fixed 1h (R/W) = Port 2 priority type escalate Escalate should not be used with queue shaping.
9	P1_PTYPE_ESC	R/W	0h	Port 1 Priority Type Escalate 0h (R/W) = Port 1 priority type fixed 1h (R/W) = Port 1 priority type escalate Escalate should not be used with queue shaping.
8	P0_PTYPE_ESC	R/W	0h	Port 0 Priority Type Escalate 0h (R/W) = Port 0 priority type fixed 1h (R/W) = Port 0 priority type escalate Escalate should not be used with queue shaping.
7-5	RESERVED	R	0h	
4-0	ESC_PRI_LD_VAL	R/W	0h	Escalate Priority Load Value When a port is in escalate priority, this is the number of higher priority packets sent before the next lower priority is allowed to send a packet. Escalate priority allows lower priority packets to be sent at a fixed rate relative to the next higher priority.

15.5.8.6 CPSW_SS_SOFT_IDLE Register (offset = 14h) [reset = 0h]

CPSW_SS_SOFT_IDLE is shown in Figure 15-190 and described in Table 15-206.

SOFTWARE IDLE

Figure 15-190. CPSW_SS_SOFT_IDLE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							SOFT_IDLE
R-0h							R/W-0h

Table 15-206. CPSW_SS_SOFT_IDLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	SOFT_IDLE	R/W	0h	Software Idle - Setting this bit causes the switch fabric to stop forwarding packets at the next start of packet.

15.5.8.7 CPSW_SS_THRU_RATE Register (offset = 18h) [reset = 3003h]

CPSW_SS_THRU_RATE is shown in Figure 15-191 and described in Table 15-207.

THROUGHPUT RATE

Figure 15-191. CPSW_SS_THRU_RATE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
SL_RX_THRU_RATE				RESERVED			
R/W-3h				R-0h			
7	6	5	4	3	2	1	0
RESERVED				CPDMA_THRU_RATE			
R-0h				R/W-3h			

Table 15-207. CPSW_SS_THRU_RATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-12	SL_RX_THRU_RATE	R/W	3h	CPGMAC_SL Switch FIFO receive through rate. This register value is the maximum throughput of the ethernet ports to the crossbar SCR. The default is one 8-byte word for every 3 CLK periods maximum.
11-8	RESERVED	R	0h	
7-4	RESERVED	R	0h	
3-0	CPDMA_THRU_RATE	R/W	3h	CPDMA Switch FIFO receive through rate. This register value is the maximum throughput of the CPDMA host port to the crossbar SCR. The default is one 8-byte word for every 3 CLK periods maximum.

15.5.8.8 CPSW_SS_GAP_THRESH Register (offset = 1Ch) [reset = Bh]

CPSW_SS_GAP_THRESH is shown in Figure 15-192 and described in Table 15-208.

CPGMAC_SL SHORT GAP THRESHOLD

Figure 15-192. CPSW_SS_GAP_THRESH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED											GAP_THRESH				
R-0h											R/W-Bh				

Table 15-208. CPSW_SS_GAP_THRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4-0	GAP_THRESH	R/W	Bh	CPGMAC_SL Short Gap Threshold - This is the CPGMAC_SL associated FIFO transmit block usage value for triggering TX_SHORT_GAP.

15.5.8.9 CPSW_SS_TX_START_WDS Register (offset = 20h) [reset = 20h]

CPSW_SS_TX_START_WDS is shown in Figure 15-193 and described in Table 15-209.

TRANSMIT START WORDS

Figure 15-193. CPSW_SS_TX_START_WDS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					TX_START_WDS										
R-0h					R/W-20h										

Table 15-209. CPSW_SS_TX_START_WDS Register Field Descriptions

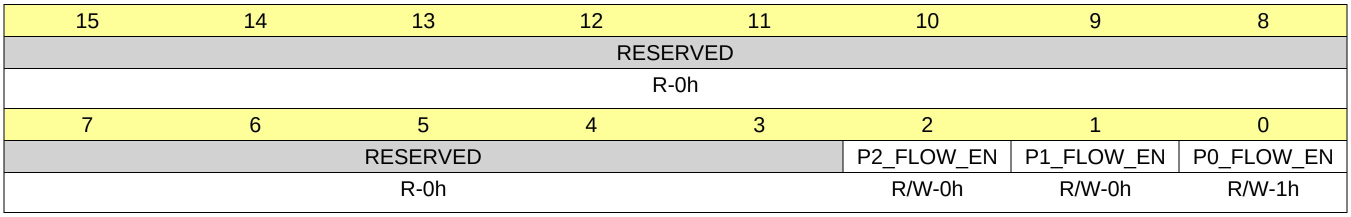
Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10-0	TX_START_WDS	R/W	20h	FIFO Packet Transmit (egress) Start Words. This value is the number of required packet words in the transmit FIFO before the packet egress will begin. This value is non-zero to preclude underrun. Decimal 32 is the recommended value. It should not be increased unnecessairly to prevent adding to the switch latency.

15.5.8.10 CPSW_SS_FLOW_CTRL Register (offset = 24h) [reset = 1h]

CPSW_SS_FLOW_CTRL is shown in Figure 15-194 and described in Table 15-210.

FLOW CONTROL

Figure 15-194. CPSW_SS_FLOW_CTRL Register



15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					P2_FLOW_EN	P1_FLOW_EN	P0_FLOW_EN
R-0h					R/W-0h	R/W-0h	R/W-1h

Table 15-210. CPSW_SS_FLOW_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-3	RESERVED	R	0h	
2	P2_FLOW_EN	R/W	0h	Port 2 Receive flow control enable
1	P1_FLOW_EN	R/W	0h	Port 1 Receive flow control enable
0	P0_FLOW_EN	R/W	1h	Port 0 Receive flow control enable

15.5.8.11 CPSW_SS_VLAN_LTYPE Register (offset = 28h) [reset = 81008100h]

CPSW_SS_VLAN_LTYPE is shown in Figure 15-195 and described in Table 15-211.

LTYPE1 AND LTYPE 2 REGISTER

Figure 15-195. CPSW_SS_VLAN_LTYPE Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">VLAN_LTYPE2</td><td colspan="16">VLAN_LTYPE1</td></tr>
<tr><td colspan="16">R/W-8100h</td><td colspan="16">R/W-8100h</td></tr>
</table>

Table 15-211. CPSW_SS_VLAN_LTYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	VLAN_LTYPE2	R/W	8100h	Time Sync VLAN LTYPE2 This VLAN LTYPE value is used for tx and rx. This is the inner VLAN if both are present.
15-0	VLAN_LTYPE1	R/W	8100h	Time Sync VLAN LTYPE1 This VLAN LTYPE value is used for tx and rx. This is the outer VLAN if both are present.

15.5.8.12 CPSW_SS_TS_LTYPE Register (offset = 2Ch) [reset = 0h]

CPSW_SS_TS_LTYPE is shown in Figure 15-196 and described in Table 15-212.

VLAN_LTYPE1 AND VLAN_LTYPE2 REGISTER

Figure 15-196. CPSW_SS_TS_LTYPE Register

31-16	15-0
TS_LTYPE2	TS_LTYPE1
R/W-0h	R/W-0h

Table 15-212. CPSW_SS_TS_LTYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TS_LTYPE2	R/W	0h	Time Sync LTYPE2 This is an Ethertype value to match for tx and rx time sync packets.
15-0	TS_LTYPE1	R/W	0h	Time Sync LTYPE1 This is an ethertype value to match for tx and rx time sync packets.

15.5.8.13 CPSW_SS_DLR_LTYPE Register (offset = 30h) [reset = 80E1h]

CPSW_SS_DLR_LTYPE is shown in Figure 15-197 and described in Table 15-213.

DLR LTYPE REGISTER

Figure 15-197. CPSW_SS_DLR_LTYPE Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DLR_LTYPE															
R/W-80E1h															

Table 15-213. CPSW_SS_DLR_LTYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	DLR_LTYPE	R/W	80E1h	DLR LTYPE

15.5.8.14 CPSW_SS_STS Register (offset = 34h) [reset = 400000h]

CPSW_SS_STS is shown in Figure 15-198 and described in Table 15-214.

Figure 15-198. CPSW_SS_STS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED	P2_FIFO_EMPTY	P1_FIFO_EMPTY	P0_FIFO_EMPTY	RESERVED			
R-0h	R-1h	R-0h	R-0h	R-0h			
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							
R-0h							

Table 15-214. CPSW_SS_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	0h	
22	P2_FIFO_EMPTY	R	1h	Port 2 Transmit FIFO Empty 0h = There are packet(s) in the port 2 transmit FIFO. 1h = The port 2 transmit FIFO is empty. The packet can still be in transmission, but when set this bit indicates that there are no full packets in the transmit FIFO.
21	P1_FIFO_EMPTY	R	0h	Port 1 Transmit FIFO Empty 0h = There are packet(s) in the port 1 transmit FIFO. 1h = The port 1 transmit FIFO is empty. The packet can still be in transmission, but when set this bit indicates that there are no full packets in the transmit FIFO.
20	P0_FIFO_EMPTY	R	0h	Port 0 Transmit FIFO Empty 0h = There are packet(s) in the port 2 transmit FIFO. 1h = The port 2 transmit FIFO is empty. The packet can still be in transmission, but when set this bit indicates that there are no full packets in the transmit FIFO.
19-0	RESERVED	R	0h	

15.5.9 CPSW_WR Registers

Table 15-215 lists the memory-mapped registers for the CPSW_WR. All register offset addresses not listed in Table 15-215 should be considered as reserved locations and the register contents should not be modified.

Table 15-215. CPSW_WR Registers

Offset	Acronym	Register Name	Section
0h	CPSW_WR_IDVER		Section 20.1.3.1
4h	CPSW_WR_SOFT_RESET		Section 15.5.9.2
8h	CPSW_WR_CTRL		Section 15.5.9.3
Ch	CPSW_WR_INT_CTRL		Section 15.5.9.4
10h	CPSW_WR_C0_RX_THRESH_EN		Section 15.5.9.5
14h	CPSW_WR_C0_RX_EN		Section 15.5.9.6
18h	CPSW_WR_C0_TX_EN		Section 15.5.9.7

Table 15-215. CPSW_WR Registers (continued)

Offset	Acronym	Register Name	Section
1Ch	CPSW_WR_C0_MISC_EN		Section 15.5.9.8
20h	CPSW_WR_C1_RX_THRESH_EN		Section 15.5.9.9
24h	CPSW_WR_C1_RX_EN		Section 15.5.9.10
28h	CPSW_WR_C1_TX_EN		Section 15.5.9.11
2Ch	CPSW_WR_C1_MISC_EN		Section 15.5.9.12
30h	CPSW_WR_C2_RX_THRESH_EN		Section 15.5.9.13
34h	CPSW_WR_C2_RX_EN		Section 15.5.9.14
38h	CPSW_WR_C2_TX_EN		Section 15.5.9.15
3Ch	CPSW_WR_C2_MISC_EN		Section 15.5.9.16
40h	CPSW_WR_C0_RX_THRESH_STAT		Section 15.5.9.17
44h	CPSW_WR_C0_RX_STAT		Section 15.5.9.18
48h	CPSW_WR_C0_TX_STAT		Section 15.5.9.19
4Ch	CPSW_WR_C0_MISC_STAT		Section 15.5.9.20
50h	CPSW_WR_C1_RX_THRESH_STAT		Section 15.5.9.21
54h	CPSW_WR_C1_RX_STAT		Section 15.5.9.22
58h	CPSW_WR_C1_TX_STAT		Section 15.5.9.23
5Ch	CPSW_WR_C1_MISC_STAT		Section 15.5.9.24
60h	CPSW_WR_C2_RX_THRESH_STAT		Section 15.5.9.25
64h	CPSW_WR_C2_RX_STAT		Section 15.5.9.26
68h	CPSW_WR_C2_TX_STAT		Section 15.5.9.27
6Ch	CPSW_WR_C2_MISC_STAT		Section 15.5.9.28
70h	CPSW_WR_C0_RX_IMAX		Section 15.5.9.29
74h	CPSW_WR_C0_TX_IMAX		Section 15.5.9.30
78h	CPSW_WR_C1_RX_IMAX		Section 15.5.9.31
7Ch	CPSW_WR_C1_TX_IMAX		Section 15.5.9.32
80h	CPSW_WR_C2_RX_IMAX		Section 15.5.9.33
84h	CPSW_WR_C2_TX_IMAX		Section 15.5.9.34
88h	CPSW_WR_RGMII_CTL		Section 15.5.9.35

15.5.9.1 CPSW_WR_IDVER Register (offset = 0h) [reset = 4EDB0100h]

CPSW_WR_IDVER is shown in Figure 20-3 and described in Table 20-6.

SUBSYSTEM ID VERSION REGISTER

Figure 15-199. CPSW_WR_IDVER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="2">SCHEME</td><td colspan="2">RESERVED</td><td colspan="4">FUNCTION</td></tr>
<tr><td colspan="2">R-1h</td><td colspan="2">R-0h</td><td colspan="4">R-EDBh</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">FUNCTION</td></tr>
<tr><td colspan="8">R-EDBh</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="5">RTL</td><td colspan="3">MAJOR</td></tr>
<tr><td colspan="5">R-0h</td><td colspan="3">R-1h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CUSTOM</td><td colspan="6">MINOR</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="6">R-0h</td></tr>
</table>

Table 15-216. CPSW_WR_IDVER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	Scheme value
29-28	RESERVED	R	0h	
27-16	FUNCTION	R	EDBh	function value
15-11	RTL	R	0h	rtl version
10-8	MAJOR	R	1h	major version
7-6	CUSTOM	R	0h	custom version
5-0	MINOR	R	0h	minor version

15.5.9.2 CPSW_WR_SOFT_RESET Register (offset = 4h) [reset = 0h]

CPSW_WR_SOFT_RESET is shown in Figure 15-200 and described in Table 15-217.

SUBSYSTEM SOFT RESET REGISTER

Figure 15-200. CPSW_WR_SOFT_RESET Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="7">RESERVED</td><td>SOFT_RESET</td></tr>
<tr><td colspan="7">R-0h</td><td>R/W-0h</td></tr>
</table>

Table 15-217. CPSW_WR_SOFT_RESET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	SOFT_RESET	R/W	0h	Software reset - Writing a one to this bit causes the CPGMACSS_R logic to be reset (INT, REGS, CPPI). Software reset occurs on the clock following the register bit write.

15.5.9.3 CPSW_WR_CTRL Register (offset = 8h) [reset = 0h]

CPSW_WR_CTRL is shown in Figure 15-201 and described in Table 15-218.

SUBSYSTEM CONTROL REGISTER

Figure 15-201. CPSW_WR_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				MMR_STDBYMODE		MMR_IDLEMODE	
R-0h				R/W-0h		R/W-0h	

Table 15-218. CPSW_WR_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-2	MMR_STDBYMODE	R/W	0h	Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of STANDBY state. 0h (R/W) = Force-standby mode : Local initiator is unconditionally placed in standbystate. 1h (R/W) = No-standby mode : Local initiator is unconditionally placed out of standby state. 2h (R/W) = Reserved : Reserved. 3h (R/W) = Reserved : Reserved.
1-0	MMR_IDLEMODE	R/W	0h	Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of IDLE state. 0h (R/W) = Force-idle mode : Local initiator is unconditionally placed in idle state. 1h (R/W) = No-idle mode : Local initiator is unconditionally placed out of idle state. 2h (R/W) = Reserved : Reserved. 3h (R/W) = Reserved : Reserved.

15.5.9.4 CPSW_WR_INT_CTRL Register (offset = Ch) [reset = 0h]

CPSW_WR_INT_CTRL is shown in Figure 15-202 and described in Table 15-219.

SUBSYSTEM INTERRUPT CONTROL

Figure 15-202. CPSW_WR_INT_CTRL Register

31	30	29	28	27	26	25	24
INT_TEST	RESERVED						
R/W-0h	R-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED		INT_PACE_EN					
R-0h		R/W-0h					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED				INT_PRESCALE			
R-0h				R-0h			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INT_PRESCALE							
R-0h							

Table 15-219. CPSW_WR_INT_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INT_TEST	R/W	0h	Interrupt Test - Test bit to the interrupt pacing blocks
30-22	RESERVED	R	0h	
21-16	INT_PACE_EN	R/W	0h	ARRAY(0x1b8eec0)
15-12	RESERVED	R	0h	
11-0	INT_PRESCALE	R	0h	Interrupt Counter Prescaler - The number of MAIN_CLK periods in 4us.

15.5.9.5 CPSW_WR_C0_RX_THRESH_EN Register (offset = 10h) [reset = 0h]

CPSW_WR_C0_RX_THRESH_EN is shown in Figure 15-203 and described in Table 15-220.

SUBSYSTEM CORE 0 RECEIVE THRESHOLD INT ENABLE REGISTER

Figure 15-203. CPSW_WR_C0_RX_THRESH_EN Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">C0_RX_THRESH_EN</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 15-220. CPSW_WR_C0_RX_THRESH_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	C0_RX_THRESH_EN	R/W	0h	Core 0 Receive Threshold Enable - Each bit in this register corresponds to the bit in the receive threshold interrupt that is enabled to generate an interrupt on C0_RX_THRESH_PULSE.

15.5.9.6 CPSW_WR_C0_RX_EN Register (offset = 14h) [reset = 0h]

CPSW_WR_C0_RX_EN is shown in Figure 15-204 and described in Table 15-221.

SUBSYSTEM CORE 0 RECEIVE INTERRUPT ENABLE REGISTER

Figure 15-204. CPSW_WR_C0_RX_EN Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">RESERVED</td><td colspan="8">C0_RX_EN</td></tr>
<tr><td colspan="24">R-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 15-221. CPSW_WR_C0_RX_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	C0_RX_EN	R/W	0h	Core 0 Receive Enable - Each bit in this register corresponds to the bit in the rx interrupt that is enabled to generate an interrupt on C0_RX_PULSE.

15.5.9.7 CPSW_WR_C0_TX_EN Register (offset = 18h) [reset = 0h]

CPSW_WR_C0_TX_EN is shown in Figure 15-205 and described in Table 15-222.

SUBSYSTEM CORE 0 TRANSMIT INTERRUPT ENABLE REGISTER

Figure 15-205. CPSW_WR_C0_TX_EN Register

31–8	7–0
RESERVED	C0_TX_EN
R-0h	R/W-0h

Table 15-222. CPSW_WR_C0_TX_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	C0_TX_EN	R/W	0h	Core 0 Transmit Enable - Each bit in this register corresponds to the bit in the tx interrupt that is enabled to generate an interrupt on C0_TX_PULSE.

15.5.9.8 CPSW_WR_C0_MISC_EN Register (offset = 1Ch) [reset = 0h]

CPSW_WR_C0_MISC_EN is shown in Figure 15-206 and described in Table 15-223.

SUBSYSTEM CORE 0 MISC INTERRUPT ENABLE REGISTER

Figure 15-206. CPSW_WR_C0_MISC_EN Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="11">RESERVED</td><td colspan="5">C0_MISC_EN</td></tr>
<tr><td colspan="11">R-0h</td><td colspan="5">R/W-0h</td></tr>
</table>

Table 15-223. CPSW_WR_C0_MISC_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4-0	C0_MISC_EN	R/W	0h	Core 0 Misc Enable - Each bit in this register corresponds to the miscellaneous interrupt (evnt_pend, stat_pend, host_pend, mdio_linkint, mdio_userint) that is enabled to generate an interrupt on C0_Misc_PULSE.

15.5.9.9 CPSW_WR_C1_RX_THRESH_EN Register (offset = 20h) [reset = 0h]

CPSW_WR_C1_RX_THRESH_EN is shown in Figure 15-207 and described in Table 15-224.

SUBSYSTEM CORE 1 RECEIVE THRESHOLD INT ENABLE REGISTER

Figure 15-207. CPSW_WR_C1_RX_THRESH_EN Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">C1_RX_THRESH_EN</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 15-224. CPSW_WR_C1_RX_THRESH_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	C1_RX_THRESH_EN	R/W	0h	Core 1 Receive Threshold Enable - Each bit in this register corresponds to the bit in the receive threshold interrupt that is enabled to generate an interrupt on C1_RX_THRESH_PULSE.

15.5.9.10 CPSW_WR_C1_RX_EN Register (offset = 24h) [reset = 0h]

CPSW_WR_C1_RX_EN is shown in Figure 15-208 and described in Table 15-225.

SUBSYSTEM CORE 1 RECEIVE INTERRUPT ENABLE REGISTER

Figure 15-208. CPSW_WR_C1_RX_EN Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">RESERVED</td><td colspan="8">C1_RX_EN</td></tr>
<tr><td colspan="24">R-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 15-225. CPSW_WR_C1_RX_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	C1_RX_EN	R/W	0h	Core 1 Receive Enable - Each bit in this register corresponds to the bit in the rx interrupt that is enabled to generate an interrupt on C1_RX_PULSE.

15.5.9.11 CPSW_WR_C1_TX_EN Register (offset = 28h) [reset = 0h]

CPSW_WR_C1_TX_EN is shown in Figure 15-209 and described in Table 15-226.

SUBSYSTEM CORE 1 TRANSMIT INTERRUPT ENABLE REGISTER

Figure 15-209. CPSW_WR_C1_TX_EN Register

31-8	7-0
RESERVED	C1_TX_EN
R-0h	R/W-0h

Table 15-226. CPSW_WR_C1_TX_EN Register Field Descriptions

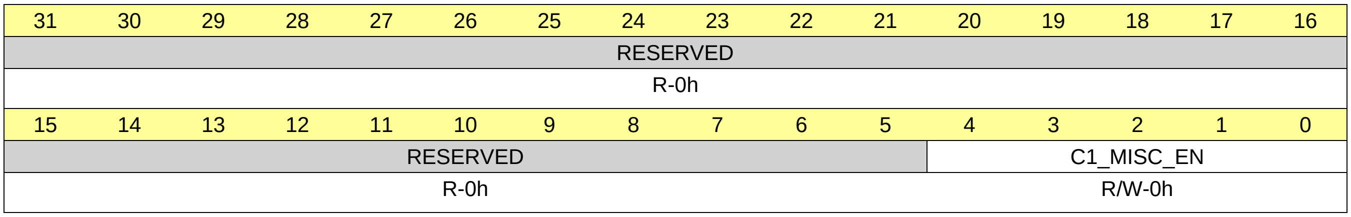
Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	C1_TX_EN	R/W	0h	Core 1 Transmit Enable - Each bit in this register corresponds to the bit in the tx interrupt that is enabled to generate an interrupt on C1_TX_PULSE.

15.5.9.12 CPSW_WR_C1_MISC_EN Register (offset = 2Ch) [reset = 0h]

CPSW_WR_C1_MISC_EN is shown in Figure 15-210 and described in Table 15-227.

SUBSYSTEM CORE 1 MISC INTERRUPT ENABLE REGISTER

Figure 15-210. CPSW_WR_C1_MISC_EN Register



<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="11">RESERVED</td><td colspan="5">C1_MISC_EN</td></tr>
<tr><td colspan="11">R-0h</td><td colspan="5">R/W-0h</td></tr>
</table>

Table 15-227. CPSW_WR_C1_MISC_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4-0	C1_MISC_EN	R/W	0h	Core 1 Misc Enable - Each bit in this register corresponds to the miscellaneous interrupt (evnt_pend, stat_pend, host_pend, mdio_linkint, mdio_userint) that is enabled to generate an interrupt on C1_Misc_PULSE.

15.5.9.13 CPSW_WR_C2_RX_THRESH_EN Register (offset = 30h) [reset = 0h]

CPSW_WR_C2_RX_THRESH_EN is shown in Figure 15-211 and described in Table 15-228.

SUBSYSTEM CORE 2 RECEIVE THRESHOLD INT ENABLE REGISTER

Figure 15-211. CPSW_WR_C2_RX_THRESH_EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								C2_RX_THRESH_EN							
R-0h								R/W-0h							

Table 15-228. CPSW_WR_C2_RX_THRESH_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	C2_RX_THRESH_EN	R/W	0h	Core 2 Receive Threshold Enable - Each bit in this register corresponds to the bit in the receive threshold interrupt that is enabled to generate an interrupt on C2_RX_THRESH_PULSE.

15.5.9.14 CPSW_WR_C2_RX_EN Register (offset = 34h) [reset = 0h]

CPSW_WR_C2_RX_EN is shown in Figure 15-212 and described in Table 15-229.

SUBSYSTEM CORE 2 RECEIVE INTERRUPT ENABLE REGISTER

Figure 15-212. CPSW_WR_C2_RX_EN Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">RESERVED</td><td colspan="8">C2_RX_EN</td></tr>
<tr><td colspan="24">R-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 15-229. CPSW_WR_C2_RX_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	C2_RX_EN	R/W	0h	Core 2 Receive Enable - Each bit in this register corresponds to the bit in the rx interrupt that is enabled to generate an interrupt on C2_RX_PULSE.

15.5.9.15 CPSW_WR_C2_TX_EN Register (offset = 38h) [reset = 0h]

CPSW_WR_C2_TX_EN is shown in Figure 15-213 and described in Table 15-230.

SUBSYSTEM CORE 2 TRANSMIT INTERRUPT ENABLE REGISTER

Figure 15-213. CPSW_WR_C2_TX_EN Register

31-8	7-0
RESERVED	C2_TX_EN
R-0h	R/W-0h

Table 15-230. CPSW_WR_C2_TX_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	C2_TX_EN	R/W	0h	Core 2 Transmit Enable - Each bit in this register corresponds to the bit in the tx interrupt that is enabled to generate an interrupt on C2_TX_PULSE.

15.5.9.16 CPSW_WR_C2_MISC_EN Register (offset = 3Ch) [reset = 0h]

CPSW_WR_C2_MISC_EN is shown in Figure 15-214 and described in Table 15-231.

SUBSYSTEM CORE 2 MISC INTERRUPT ENABLE REGISTER

Figure 15-214. CPSW_WR_C2_MISC_EN Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="11">RESERVED</td><td colspan="5">C2_MISC_EN</td></tr>
<tr><td colspan="11">R-0h</td><td colspan="5">R/W-0h</td></tr>
</table>

Table 15-231. CPSW_WR_C2_MISC_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4-0	C2_MISC_EN	R/W	0h	Core 2 Misc Enable - Each bit in this register corresponds to the miscellaneous interrupt (evnt_pend, stat_pend, host_pend, mdio_linkint, mdio_userint) that is enabled to generate an interrupt on C2_Misc_PULSE.

15.5.9.17 CPSW_WR_C0_RX_THRESH_STAT Register (offset = 40h) [reset = 0h]

CPSW_WR_C0_RX_THRESH_STAT is shown in Figure 15-215 and described in Table 15-232.

SUBSYSTEM CORE 0 RX THRESHOLD MASKED INT STATUS REGISTER

Figure 15-215. CPSW_WR_C0_RX_THRESH_STAT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">C0_RX_THRESH_STAT</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="8">R-0h</td></tr>
</table>

Table 15-232. CPSW_WR_C0_RX_THRESH_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	C0_RX_THRESH_STAT	R	0h	Core 0 Receive Threshold Masked Interrupt Status - Each bit in this read only register corresponds to the bit in the receive threshold interrupt that is enabled and generating an interrupt on C0_RX_THRESH_PULSE.

15.5.9.18 CPSW_WR_C0_RX_STAT Register (offset = 44h) [reset = 0h]

CPSW_WR_C0_RX_STAT is shown in Figure 15-216 and described in Table 15-233.

SUBSYSTEM CORE 0 RX INTERRUPT MASKED INT STATUS REGISTER

Figure 15-216. CPSW_WR_C0_RX_STAT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">C0_RX_STAT</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="8">R-0h</td></tr>
</table>

Table 15-233. CPSW_WR_C0_RX_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	C0_RX_STAT	R	0h	Core 0 Receive Masked Interrupt Status - Each bit in this read only register corresponds to the bit in the Rx interrupt that is enabled and generating an interrupt on C0_RX_PULSE.

15.5.9.19 CPSW_WR_C0_TX_STAT Register (offset = 48h) [reset = 0h]

CPSW_WR_C0_TX_STAT is shown in Figure 15-217 and described in Table 15-234.

SUBSYSTEM CORE 0 TX INTERRUPT MASKED INT STATUS REGISTER

Figure 15-217. CPSW_WR_C0_TX_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED								C0_TX_STAT							
R-0h								R-0h							

Table 15-234. CPSW_WR_C0_TX_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	C0_TX_STAT	R	0h	Core 0 Transmit Masked Interrupt Status - Each bit in this read only register corresponds to the bit in the Tx interrupt that is enabled and generating an interrupt on C0_TX_PULSE .

15.5.9.20 CPSW_WR_C0_MISC_STAT Register (offset = 4Ch) [reset = 0h]

CPSW_WR_C0_MISC_STAT is shown in Figure 15-218 and described in Table 15-235.

SUBSYSTEM CORE 0 MISC INTERRUPT MASKED INT STATUS REGISTER

Figure 15-218. CPSW_WR_C0_MISC_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED									C0_MISC_STAT						
R-0h									R-0h						

Table 15-235. CPSW_WR_C0_MISC_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6-0	C0_MISC_STAT	R	0h	Core 0 Misc Masked Interrupt Status - Each bit in this register corresponds to the miscellaneous interrupt (evnt_pend, stat_pend, host_pend, mdio_linkint, mdio_userint) that is enabled and generating an interrupt on C0_MISC_PULSE .

15.5.9.21 CPSW_WR_C1_RX_THRESH_STAT Register (offset = 50h) [reset = 0h]

CPSW_WR_C1_RX_THRESH_STAT is shown in Figure 15-219 and described in Table 15-236.

SUBSYSTEM CORE 1 RX THRESHOLD MASKED INT STATUS REGISTER

Figure 15-219. CPSW_WR_C1_RX_THRESH_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED								C1_RX_THRESH_STAT							
R-0h								R-0h							

Table 15-236. CPSW_WR_C1_RX_THRESH_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	C1_RX_THRESH_STAT	R	0h	Core 1 Receive Threshold Masked Interrupt Status - Each bit in this register corresponds to the bit in the receive threshold interrupt that is enabled and generating an interrupt on C1_RX_THRESH_PULSE.

15.5.9.22 CPSW_WR_C1_RX_STAT Register (offset = 54h) [reset = 0h]

CPSW_WR_C1_RX_STAT is shown in Figure 15-220 and described in Table 15-237.

SUBSYSTEM CORE 1 RECEIVE MASKED INTERRUPT STATUS REGISTER

Figure 15-220. CPSW_WR_C1_RX_STAT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">C1_RX_STAT</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="8">R-0h</td></tr>
</table>

Table 15-237. CPSW_WR_C1_RX_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	C1_RX_STAT	R	0h	Core 1 Receive Masked Interrupt Status - Each bit in this register corresponds to the bit in the Rx interrupt that is enabled and generating an interrupt on C1_RX_PULSE.

15.5.9.23 CPSW_WR_C1_TX_STAT Register (offset = 58h) [reset = 0h]

CPSW_WR_C1_TX_STAT is shown in Figure 15-221 and described in Table 15-238.

SUBSYSTEM CORE 1 TRANSMIT MASKED INTERRUPT STATUS REGISTER

Figure 15-221. CPSW_WR_C1_TX_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								C1_TX_STAT							
R-0h								R-0h							

Table 15-238. CPSW_WR_C1_TX_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	C1_TX_STAT	R	0h	Core 1 Transmit Masked Interrupt Status - Each bit in this register corresponds to the bit in the Tx interrupt that is enabled and generating an interrupt on C1_TX_PULSE.

15.5.9.24 CPSW_WR_C1_MISC_STAT Register (offset = 5Ch) [reset = 0h]

CPSW_WR_C1_MISC_STAT is shown in Figure 15-222 and described in Table 15-239.

SUBSYSTEM CORE 1 MISC MASKED INTERRUPT STATUS REGISTER

Figure 15-222. CPSW_WR_C1_MISC_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED									C1_MISC_STAT						
R-0h									R-0h						

Table 15-239. CPSW_WR_C1_MISC_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6-0	C1_MISC_STAT	R	0h	Core 1 Misc Masked Interrupt Status - Each bit in this register corresponds to the miscellaneous interrupt (evnt_pend, stat_pend, host_pend, mdio_linkint, mdio_userint) that is enabled and generating an interrupt on C1_MISC_PULSE .

15.5.9.25 CPSW_WR_C2_RX_THRESH_STAT Register (offset = 60h) [reset = 0h]

CPSW_WR_C2_RX_THRESH_STAT is shown in Figure 15-223 and described in Table 15-240.

SUBSYSTEM CORE 2 RX THRESHOLD MASKED INT STATUS REGISTER

Figure 15-223. CPSW_WR_C2_RX_THRESH_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								C2_RX_THRESH_STAT							
R-0h								R-0h							

Table 15-240. CPSW_WR_C2_RX_THRESH_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	C2_RX_THRESH_STAT	R	0h	Core 2 Receive Threshold Masked Interrupt Status - Each bit in this register corresponds to the bit in the receive threshold interrupt that is enabled and generating an interrupt on C2_RX_THRESH_PULSE.

15.5.9.26 CPSW_WR_C2_RX_STAT Register (offset = 64h) [reset = 0h]

CPSW_WR_C2_RX_STAT is shown in Figure 15-224 and described in Table 15-241.

SUBSYSTEM CORE 2 RECEIVE MASKED INTERRUPT STATUS REGISTER

Figure 15-224. CPSW_WR_C2_RX_STAT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">C2_RX_STAT</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="8">R-0h</td></tr>
</table>

Table 15-241. CPSW_WR_C2_RX_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	C2_RX_STAT	R	0h	Core 2 Receive Masked Interrupt Status - Each bit in this register corresponds to the bit in the Rx interrupt that is enabled and generating an interrupt on C2_RX_PULSE.

15.5.9.27 CPSW_WR_C2_TX_STAT Register (offset = 68h) [reset = 0h]

CPSW_WR_C2_TX_STAT is shown in Figure 15-225 and described in Table 15-242.

SUBSYSTEM CORE 2 TRANSMIT MASKED INTERRUPT STATUS REGISTER

Figure 15-225. CPSW_WR_C2_TX_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								C2_TX_STAT							
R-0h								R-0h							

Table 15-242. CPSW_WR_C2_TX_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	C2_TX_STAT	R	0h	Core 2 Transmit Masked Interrupt Status - Each bit in this register corresponds to the bit in the Tx interrupt that is enabled and generating an interrupt on C2_TX_PULSE.

15.5.9.28 CPSW_WR_C2_MISC_STAT Register (offset = 6Ch) [reset = 0h]

CPSW_WR_C2_MISC_STAT is shown in Figure 15-226 and described in Table 15-243.

SUBSYSTEM CORE 2 MISC MASKED INTERRUPT STATUS REGISTER

Figure 15-226. CPSW_WR_C2_MISC_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									C2_MISC_STAT						
R-0h									R-0h						

Table 15-243. CPSW_WR_C2_MISC_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6-0	C2_MISC_STAT	R	0h	Core 2 Misc Masked Interrupt Status - Each bit in this register corresponds to the miscellaneous interrupt (evnt_pend, stat_pend, host_pend, mdio_linkint, mdio_userint) that is enabled and generating an interrupt on C2_MISC_PULSE .

15.5.9.29 CPSW_WR_C0_RX_IMAX Register (offset = 70h) [reset = 0h]

CPSW_WR_C0_RX_IMAX is shown in Figure 15-227 and described in Table 15-244.

SUBSYSTEM CORE 0 RECEIVE INTERRUPTS PER MILLISECOND

Figure 15-227. CPSW_WR_C0_RX_IMAX Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="10">RESERVED</td><td colspan="6">C0_RX_IMAX</td></tr>
<tr><td colspan="10">R-0h</td><td colspan="6">R/W-0h</td></tr>
</table>

Table 15-244. CPSW_WR_C0_RX_IMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5-0	C0_RX_IMAX	R/W	0h	Core 0 Receive Interrupts per Millisecond - The maximum number of interrupts per millisecond generated on C0_RX_PULSE if pacing is enabled for this interrupt.

15.5.9.30 CPSW_WR_C0_TX_IMAX Register (offset = 74h) [reset = 0h]

CPSW_WR_C0_TX_IMAX is shown in Figure 15-228 and described in Table 15-245.

SUBSYSTEM CORE 0 TRANSMIT INTERRUPTS PER MILLISECOND

Figure 15-228. CPSW_WR_C0_TX_IMAX Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="10">RESERVED</td><td colspan="6">C0_TX_IMAX</td></tr>
<tr><td colspan="10">R-0h</td><td colspan="6">R/W-0h</td></tr>
</table>

Table 15-245. CPSW_WR_C0_TX_IMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5-0	C0_TX_IMAX	R/W	0h	Core 0 Transmit Interrupts per Millisecond - The maximum number of interrupts per millisecond generated on C0_TX_PULSE if pacing is enabled for this interrupt.

15.5.9.31 CPSW_WR_C1_RX_IMAX Register (offset = 78h) [reset = 0h]

CPSW_WR_C1_RX_IMAX is shown in Figure 15-229 and described in Table 15-246.

SUBSYSTEM CORE 1 RECEIVE INTERRUPTS PER MILLISECOND

Figure 15-229. CPSW_WR_C1_RX_IMAX Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="10">RESERVED</td><td colspan="6">C1_RX_IMAX</td></tr>
<tr><td colspan="10">R-0h</td><td colspan="6">R/W-0h</td></tr>
</table>

Table 15-246. CPSW_WR_C1_RX_IMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5-0	C1_RX_IMAX	R/W	0h	Core 1 Receive Interrupts per Millisecond - The maximum number of interrupts per millisecond generated on C1_RX_PULSE if pacing is enabled for this interrupt.

15.5.9.32 CPSW_WR_C1_TX_IMAX Register (offset = 7Ch) [reset = 0h]

CPSW_WR_C1_TX_IMAX is shown in Figure 15-230 and described in Table 15-247.

SUBSYSTEM CORE 1 TRANSMIT INTERRUPTS PER MILLISECOND

Figure 15-230. CPSW_WR_C1_TX_IMAX Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED										C1_TX_IMAX					
R-0h										R/W-0h					

Table 15-247. CPSW_WR_C1_TX_IMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5-0	C1_TX_IMAX	R/W	0h	Core 1 Transmit Interrupts per Millisecond - The maximum number of interrupts per millisecond generated on C1_TX_PULSE if pacing is enabled for this interrupt.

15.5.9.33 CPSW_WR_C2_RX_IMAX Register (offset = 80h) [reset = 0h]

CPSW_WR_C2_RX_IMAX is shown in Figure 15-231 and described in Table 15-248.

SUBSYSTEM CORE 2 RECEIVE INTERRUPTS PER MILLISECOND

Figure 15-231. CPSW_WR_C2_RX_IMAX Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="10">RESERVED</td><td colspan="6">C2_RX_IMAX</td></tr>
<tr><td colspan="10">R-0h</td><td colspan="6">R/W-0h</td></tr>
</table>

Table 15-248. CPSW_WR_C2_RX_IMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5-0	C2_RX_IMAX	R/W	0h	Core 2 Receive Interrupts per Millisecond - The maximum number of interrupts per millisecond generated on C2_RX_PULSE if pacing is enabled for this interrupt.

15.5.9.34 CPSW_WR_C2_TX_IMAX Register (offset = 84h) [reset = 0h]

CPSW_WR_C2_TX_IMAX is shown in Figure 15-232 and described in Table 15-249.

SUBSYSTEM CORE 2 TRANSMIT INTERRUPTS PER MILLISECOND

Figure 15-232. CPSW_WR_C2_TX_IMAX Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										C2_TX_IMAX					
R-0h										R/W-0h					

Table 15-249. CPSW_WR_C2_TX_IMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5-0	C2_TX_IMAX	R/W	0h	Core 2 Transmit Interrupts per Millisecond - The maximum number of interrupts per millisecond generated on C2_TX_PULSE if pacing is enabled for this interrupt.

15.5.9.35 CPSW_WR_RGMII_CTL Register (offset = 88h) [reset = 0h]

CPSW_WR_RGMII_CTL is shown in Figure 15-233 and described in Table 15-250.

RGMII CONTROL SIGNAL REGISTER

Figure 15-233. CPSW_WR_RGMII_CTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RGMII2_FULL DUPLEX	RGMII2_SPEED		RGMII2_LINK	RGMII1_FULL DUPLEX	RGMII1_SPEED		RGMII1_LINK
R-0h	R-0h		R-0h	R-0h	R-0h		R-0h

Table 15-250. CPSW_WR_RGMII_CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	RGMII2_FULLDUPLEX	R	0h	RGMII 2 Fullduplex - This is the CPRGMII fullduplex output signal. 0h (R/W) = Half-duplex mode 1h (R/W) = Full-duplex mode
6-5	RGMII2_SPEED	R	0h	RGMII2 Speed - This is the CPRGMI speed output signal 0h (R/W) = 10Mbps mode 1h (R/W) = 100Mbps mode 2h (R/W) = 1000Mbps (gig) mode 3h (R/W) = Reserved
4	RGMII2_LINK	R	0h	RGMII2 Link Indicator - This is the CPRGMII link output signal 0h (R/W) = RGMII2 link is down 1h (R/W) = RGMII2 link is up
3	RGMII1_FULLDUPLEX	R	0h	RGMII1 Fullduplex - This is the CPRGMII fullduplex output signal. 0h (R/W) = Half-duplex mode 1h (R/W) = Full-duplex mode
2-1	RGMII1_SPEED	R	0h	RGMII1 Speed - This is the CPRGMII speed output signal 0h (R/W) = 10Mbps mode 1h (R/W) = 100Mbps mode 2h (R/W) = 1000Mbps (gig) mode 3h (R/W) = Reserved
0	RGMII1_LINK	R	0h	RGMII1 Link Indicator - This is the CPRGMII link output signal 0h (R/W) = RGMII1 link is down 1h (R/W) = RGMII1 link is up

15.5.10 MDIO Registers

Table 15-251 lists the memory-mapped registers for the MDIO. All register offset addresses not listed in Table 15-251 should be considered as reserved locations and the register contents should not be modified.

Table 15-251. MDIO REGISTERS

Offset	Acronym	Register Name	Section
0h	MDIO_VER	MDIO Version Register	Section 15.5.10.1
4h	MDIO_CTRL	MDIO Control Register	Section 15.5.10.2
8h	MDIO_ALIVE	PHY Alive Status Register	Section 15.5.10.3
Ch	MDIO_LINK	PHY Link Status Register	Section 15.5.10.4
10h	MDIO_LINKINTRAW	MDIO Link Status Change Interrupt Register	Section 15.5.10.5
14h	MDIO_LINKINTMASKED	MDIO Link Status Change Interrupt Register (Masked Value)	Section 15.5.10.6
20h	MDIO_USERINTRAW	MDIO User Command Complete Interrupt Register (Raw Value)	Section 15.5.10.7
24h	MDIO_USERINTMASKED	MDIO User Command Complete Interrupt Register (Masked Value)	Section 15.5.10.8
28h	MDIO_USERINTMASKSET	MDIO User Command Complete Interrupt Mask Set Register	Section 15.5.10.9
2Ch	MDIO_USERINTMASKCLR	MDIO User Interrupt Mask Clear Register	Section 15.5.10.10
80h	MDIO_USERACCESS0	MDIO User Access Register 0	Section 15.5.10.11
84h	MDIO_USERPHYSEL0	MDIO User PHY Select Register 0	Section 15.5.10.12
88h	MDIO_USERACCESS1	MDIO User Access Register 1	Section 15.5.10.13
8Ch	MDIO_USERPHYSEL1	MDIO User PHY Select Register 1	Section 15.5.10.14

15.5.10.1 MDIO_VER Register (offset = 0h) [reset = 70104h]

MDIO_VER is shown in Figure 15-234 and described in Table 15-252.

Figure 15-234. MDIO_VER Register

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="16">MODID</td><td colspan="8">REVMAJ</td><td colspan="8">REVMIN</td></tr>
<tr><td colspan="16">R-7h</td><td colspan="8">R-1h</td><td colspan="8">R-4h</td></tr>
</table>

Table 15-252. MDIO_VER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MODID	R	7h	Identifies type of peripheral.
15-8	REVMAJ	R	1h	Management interface module major revision value.
7-0	REVMIN	R	4h	Management interface module minor revision value.

15.5.10.2 MDIO_CTRL Register (offset = 4h) [reset = 810000FFh]

MDIO_CTRL is shown in Figure 15-235 and described in Table 15-253.

Figure 15-235. MDIO_CTRL Register

31	30	29	28	27	26	25	24
IDLE	EN	RESERVED1	HIGHEST_USER_CHANNEL				
R-1h	R/W-0h	R-0h	R-1h				

23	22	21	20	19	18	17	16
RESERVED2			PREAMBLE	FAULT	FAULTENB	INTTESTENB	RESERVED3
R-0h			R/W-0h	R/WC-0h	R/W-0h	R/W-0h	R-0h

15	14	13	12	11	10	9	8
CLKDIV							
R/W-FFh							

7	6	5	4	3	2	1	0
CLKDIV							
R/W-FFh							

Table 15-253. MDIO_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IDLE	R	1h	MDIO state machine IDLE. Set to 1 when the state machine is in the idle state. 0h = State machine is not in idle state. 1h = State machine is in idle state.
30	EN	R/W	0h	Enable control. If the MDIO state machine is active at the time it is disabled, it will complete the current operation before halting and setting the idle bit. If using byte access, the enable bit has to be the last bit written in this register. 0h = Disables the MDIO state machine. 1h = Enable the MDIO state machine.
29	RESERVED1	R	0h	
28-24	HIGHEST_USER_CHANNEL	R	1h	Highest user channel. This field specifies the highest user access channel that is available in the module and is currently set to 1. This implies that the MDIOUSERACCESS1 register is the highest available user access channel.
23-21	RESERVED2	R	0h	
20	PREAMBLE	R/W	0h	Preamble disable. 0h = Standard MDIO preamble is used. 1h = Disables this device from sending MDIO frame preambles.
19	FAULT	R/WC	0h	Fault indicator. This bit is set to 1 if the MDIO pins fail to read back what the device is driving onto them. This indicates a physical layer fault and the module state machine is reset. Writing a 1 to it clears this bit. 0h = No failure. 1h = Physical layer fault; the MDIO state machine is reset.
18	FAULTENB	R/W	0h	Fault detect enable. This bit has to be set to 1 to enable the physical layer fault detection. 0h = Disables the physical layer fault detection. 1h = Enables the physical layer fault detection.

Table 15-253. MDIO_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	INTTESTENB	R/W	0h	Interrupt test enable. This bit can be set to 1 to enable the host to set the USERINT and LINKINT bits for test purposes. 0h = Interrupt bits are not set. 1h = Enables the host to set the USERINT and LINKINT bits for test purposes.
16	RESERVED3	R	0h	
15-0	CLKDIV	R/W	FFh	Clock divider. This field specifies the division ratio between CLK and the frequency of MDIO_CLK. MDIO_CLK is disabled when clkdiv is set to 0. MDIO_CLK frequency = clk frequency/(clkdiv+1).

15.5.10.3 MDIO_ALIVE Register (offset = 8h) [reset = 0h]

MDIO_ALIVE is shown in Figure 15-236 and described in Table 15-254.

Figure 15-236. MDIO_ALIVE Register

31-0
ALIVE
R/WC-0h

Table 15-254. MDIO_ALIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ALIVE	R/WC	0h	MDIO alive. Each of the 32 bits of this register is set if the most recent access to the PHY with address corresponding to the register bit number was acknowledged by the PHY, the bit is reset if the PHY fails to acknowledge the access. Both the user and polling accesses to a PHY will cause the corresponding alive bit to be updated. The alive bits are only meant to be used to give an indication of the presence or not of a PHY with the corresponding address. Writing a 1 to any bit will clear it, writing a 0 has no effect.

15.5.10.4 MDIO_LINK Register (offset = Ch) [reset = 0h]

MDIO_LINK is shown in Figure 15-237 and described in Table 15-255.

Figure 15-237. MDIO_LINK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINK																															
R-0h																															

Table 15-255. MDIO_LINK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LINK	R	0h	MDIO link state. This register is updated after a read of the Generic Status Register of a PHY. The bit is set if the PHY with the corresponding address has link and the PHY acknowledges the read transaction. The bit is reset if the PHY indicates it does not have link or fails to acknowledge the read transaction. Writes to the register have no effect. In addition, the status of the two PHYs specified in the MDIOUSERPHYSEL registers can be determined using the MLINK input pins. This is determined by the LINKSEL bit in the MDIOUSERPHYSEL register.

15.5.10.5 MDIO_LINKINTRAW Register (offset = 10h) [reset = 0h]

MDIO_LINKINTRAW is shown in Figure 15-238 and described in Table 15-256.

Figure 15-238. MDIO_LINKINTRAW Register

31	30	29	28	27	26	25	24
RESERVED1							
R-0h							
23	22	21	20	19	18	17	16
RESERVED1							
R-0h							
15	14	13	12	11	10	9	8
RESERVED1							
R-0h							
7	6	5	4	3	2	1	0
RESERVED1						LINKINTRAW	
R-0h						R/WC-0h	

Table 15-256. MDIO_LINKINTRAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED1	R	0h	
1-0	LINKINTRAW	R/WC	0h	MDIO link change event, raw value. When asserted 1, a bit indicates that there was an MDIO link change event (that is, change in the MDIOLINK register) corresponding to the PHY address in the MDIOUSERPHYSEL register. LINKINTRAW[0] and LINKINTRAW[1] correspond to MDIOUSERPHYSEL0 and MDIOUSERPHYSEL1, respectively. Writing a 1 will clear the event and writing 0 has no effect. If the INTTESTENB bit in the MDIOCONTROL register is set, the host may set the LINKINTRAW bits to a 1. This mode may be used for test purposes.

15.5.10.6 MDIO_LINKINTMASKED Register (offset = 14h) [reset = 0h]

MDIO_LINKINTMASKED is shown in Figure 15-239 and described in Table 15-257.

Figure 15-239. MDIO_LINKINTMASKED Register

31	30	29	28	27	26	25	24
RESERVED1							
R-0h							

23	22	21	20	19	18	17	16
RESERVED1							
R-0h							

15	14	13	12	11	10	9	8
RESERVED1							
R-0h							

7	6	5	4	3	2	1	0
RESERVED1						LINKINTMASKED	
R-0h						R/WC-0h	

Table 15-257. MDIO_LINKINTMASKED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED1	R	0h	
1-0	LINKINTMASKED	R/WC	0h	MDIO link change interrupt, masked value. When asserted 1, a bit indicates that there was an MDIO link change event (that is, change in the MDIO Link register) corresponding to the PHY address in the MDIOUSERPHYSEL register and the corresponding LINKINTENB bit was set. LINKINTMASKED[0] and LINKINTMASKED[1] correspond to MDIOUSERPHYSEL0 and MDIOUSERPHYSEL1, respectively. Writing a 1 will clear the interrupt and writing 0 has no effect. If the INTTESTENB bit in the MDIOCONTROL register is set, the host may set the LINKINT bits to a 1. This mode may be used for test purposes.

15.5.10.7 MDIO_USERINTRAW Register (offset = 20h) [reset = 0h]

MDIO_USERINTRAW is shown in Figure 15-240 and described in Table 15-258.

Figure 15-240. MDIO_USERINTRAW Register

31	30	29	28	27	26	25	24
RESERVED1							
R-0h							

23	22	21	20	19	18	17	16
RESERVED1							
R-0h							

15	14	13	12	11	10	9	8
RESERVED1							
R-0h							

7	6	5	4	3	2	1	0
RESERVED1						USERINTRAW	
R-0h						R/WC-0h	

Table 15-258. MDIO_USERINTRAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED1	R	0h	
1-0	USERINTRAW	R/WC	0h	Raw value of MDIO user command complete event for the MDIOUSERACCESS1 register through the MDIOUSERACCESS0 register, respectively. When asserted 1, a bit indicates that the previously scheduled PHY read or write command using that particular MDIOUSERACCESSn register has completed. Writing a 1 will clear the event and writing 0 has no effect. If the INTTESTENB bit in the MDIOCONTROL register is set, the host may set the USERINTRAW bits to a 1. This mode may be used for test purposes.

15.5.10.8 MDIO_USERINTMASKED Register (offset = 24h) [reset = 0h]

MDIO_USERINTMASKED is shown in Figure 15-241 and described in Table 15-259.

Figure 15-241. MDIO_USERINTMASKED Register

31	30	29	28	27	26	25	24
RESERVED1							
R-0h							

23	22	21	20	19	18	17	16
RESERVED1							
R-0h							

15	14	13	12	11	10	9	8
RESERVED1							
R-0h							

7	6	5	4	3	2	1	0
RESERVED1						USERINTMASKED	
R-0h						R/WC-0h	

Table 15-259. MDIO_USERINTMASKED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED1	R	0h	
1-0	USERINTMASKED	R/WC	0h	Masked value of MDIO user command complete interrupt for the MDIOUSERACCESS1 register through the MDIOUSERACCESS0 register, respectively. When asserted 1, a bit indicates that the previously scheduled PHY read or write command using that particular MDIOUSERACCESSn register has completed and the corresponding USERINTMASKSET bit is set to 1. Writing a 1 will clear the interrupt and writing 0 has no effect. If the INTTESTENB bit in the MDIOCONTROL register is set, the host may set the USERINTMASKED bits to a 1. This mode may be used for test purposes.

15.5.10.9 MDIO_USERINTMASKSET Register (offset = 28h) [reset = 0h]

MDIO_USERINTMASKSET is shown in Figure 15-242 and described in Table 15-260.

Figure 15-242. MDIO_USERINTMASKSET Register

31	30	29	28	27	26	25	24
RESERVED1							
R-0h							

23	22	21	20	19	18	17	16
RESERVED1							
R-0h							

15	14	13	12	11	10	9	8
RESERVED1							
R-0h							

7	6	5	4	3	2	1	0
RESERVED1						USERINTMASKSET	
R-0h						R/WC-0h	

Table 15-260. MDIO_USERINTMASKSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED1	R	0h	
1-0	USERINTMASKSET	R/WC	0h	MDIO user interrupt mask set for USERINTMASKED, respectively. Writing a bit to 1 will enable MDIO user command complete interrupts for that particular MDIOUSERACCESSn register. MDIO user interrupt for a particular MDIOUSERACCESSn register is disabled if the corresponding bit is 0. Writing a 0 to this register has no effect.

15.5.10.10 MDIO_USERINTMASKCLR Register (offset = 2Ch) [reset = 0h]

MDIO_USERINTMASKCLR is shown in Figure 15-243 and described in Table 15-261.

Figure 15-243. MDIO_USERINTMASKCLR Register

31	30	29	28	27	26	25	24
RESERVED1							
R-0h							

23	22	21	20	19	18	17	16
RESERVED1							
R-0h							

15	14	13	12	11	10	9	8
RESERVED1							
R-0h							

7	6	5	4	3	2	1	0
RESERVED1						USERINTMASKCLR	
R-0h						R/WC-0h	

Table 15-261. MDIO_USERINTMASKCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED1	R	0h	
1-0	USERINTMASKCLR	R/WC	0h	MDIO user command complete interrupt mask clear for USERINTMASKED, respectively. Writing a bit to 1 will disable further user command complete interrupts for that particular MDIOUSERACCESSn register. Writing a 0 to this register has no effect.

15.5.10.11 MDIO_USERACCESS0 Register (offset = 80h) [reset = 0h]

MDIO_USERACCESS0 is shown in Figure 15-244 and described in Table 15-262.

Figure 15-244. MDIO_USERACCESS0 Register

31	30	29	28	27	26	25	24
GO	WRITE	ACK	RESERVED1			REGADR	
R/W/S-0h	R/W-0h	R/W-0h	R-0h			R/W-0h	

23	22	21	20	19	18	17	16
REGADR			PHYADR				
R/W-0h			R/W-0h				

15	14	13	12	11	10	9	8
DATA							
R/W-0h							

7	6	5	4	3	2	1	0
DATA							
R/W-0h							

Table 15-262. MDIO_USERACCESS0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GO	R/W/S	0h	Go. Writing a 1 to this bit causes the MDIO state machine to perform an MDIO access when it is convenient for it to do so, this is not an instantaneous process. Writing a 0 to this bit has no effect. This bit is write able only if the MDIO state machine is enabled. This bit will self clear when the requested access has been completed. Any writes to the MDIOUSERACCESS0 register are blocked when the GO bit is 1. If byte access is being used, the GO bit should be written last.
30	WRITE	R/W	0h	Write enable. Setting this bit to a 1 causes the MDIO transaction to be a register write, otherwise it is a register read.
29	ACK	R/W	0h	Acknowledge. This bit is set if the PHY acknowledged the read transaction.
28-26	RESERVED1	R	0h	
25-21	REGADR	R/W	0h	Register address. Specifies the PHY register to be accessed for this transaction.
20-16	PHYADR	R/W	0h	PHY address. Specifies the PHY to be accesses for this transaction.
15-0	DATA	R/W	0h	User data. The data value read from or to be written to the specified PHY register.

15.5.10.12 MDIO_USERPHYSEL0 Register (offset = 84h) [reset = 0h]

MDIO_USERPHYSEL0 is shown in Figure 15-245 and described in Table 15-263.

Figure 15-245. MDIO_USERPHYSEL0 Register

31	30	29	28	27	26	25	24
RESERVED1							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED1							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED1							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LINKSEL	LINKINTENB	RESERVED2	PHYADDRMON				
R/W-0h	R/W-0h	R-0h	R/W-0h				

Table 15-263. MDIO_USERPHYSEL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED1	R	0h	
7	LINKSEL	R/W	0h	Link status determination select. Set to 1 to determine link status using the MLINK pin. Default value is 0 which implies that the link status is determined by the MDIO state machine.
6	LINKINTENB	R/W	0h	Link change interrupt enable. Set to 1 to enable link change status interrupts for PHY address specified in PHYADDRMON. Link change interrupts are disabled if this bit is set to 0. 0h = Link change interrupts are disabled. 1h = Link change status interrupts for PHY address specified in PHYADDRMON bits are enabled.
5	RESERVED2	R	0h	
4-0	PHYADDRMON	R/W	0h	PHY address whose link status is to be monitored.

15.5.10.13 MDIO_USERACCESS1 Register (offset = 88h) [reset = 0h]

MDIO_USERACCESS1 is shown in Figure 15-246 and described in Table 15-264.

Figure 15-246. MDIO_USERACCESS1 Register

31	30	29	28	27	26	25	24
GO	WRITE	ACK	RESERVED1			REGADR	
R/W/S-0h	R/W-0h	R/W-0h	R-0h			R/W-0h	

23	22	21	20	19	18	17	16
REGADR			PHYADR				
R/W-0h			R/W-0h				

15	14	13	12	11	10	9	8
DATA							
R/W-0h							

7	6	5	4	3	2	1	0
DATA							
R/W-0h							

Table 15-264. MDIO_USERACCESS1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GO	R/W/S	0h	Writing a 1 to this bit causes the MDIO state machine to perform an MDIO access when it is convenient for it to do so, this is not an instantaneous process. Writing a 0 to this bit has no effect. This bit is write able only if the MDIO state machine is enabled. This bit will self clear when the requested access has been completed. Any writes to the MDIOUSERACCESS0 register are blocked when the GO bit is 1. If byte access is being used, the GO bit should be written last.
30	WRITE	R/W	0h	Write enable. Setting this bit to a 1 causes the MDIO transaction to be a register write, otherwise it is a register read.
29	ACK	R/W	0h	Acknowledge. This bit is set if the PHY acknowledged the read transaction.
28-26	RESERVED1	R	0h	
25-21	REGADR	R/W	0h	Register address specifies the PHY register to be accessed for this transaction.
20-16	PHYADR	R/W	0h	PHY address specifies the PHY to be accesses for this transaction.
15-0	DATA	R/W	0h	User data. The data value read from or to be written to the specified PHY register.

15.5.10.14 MDIO_USERPHYSEL1 Register (offset = 8Ch) [reset = 0h]

MDIO_USERPHYSEL1 is shown in Figure 15-247 and described in Table 15-265.

Figure 15-247. MDIO_USERPHYSEL1 Register

31	30	29	28	27	26	25	24
RESERVED1							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED1							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED1							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LINKSEL	LINKINTENB	RESERVED2	PHYADDRMON				
R/W-0h	R/W-0h	R-0h	R/W-0h				

Table 15-265. MDIO_USERPHYSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED1	R	0h	
7	LINKSEL	R/W	0h	Link status determination select. Set to 1 to determine link status using the MLINK pin. Default value is 0 which implies that the link status is determined by the MDIO state machine.
6	LINKINTENB	R/W	0h	Link change interrupt enable. Set to 1 to enable link change status interrupts for PHY address specified in PHYADDRMON. Link change interrupts are disabled if this bit is cleared to 0. 0h = Link change interrupts are disabled 1h = Link change status interrupts for PHY address specified in PHYADDRMON bits are enabled.
5	RESERVED2	R	0h	
4-0	PHYADDRMON	R/W	0h	PHY address whose link status is to be monitored.

Chapter 16

Universal Serial Bus (USB)

This chapter describes the universal serial bus (USB) of the device.

Topic **Page**

16.1 Introduction

USB (Universal Serial Bus) provides a low-cost connectivity solution for numerous consumer portable devices by implementing a mechanism for data transfer between USB devices.

The AM437x instantiates two independent instances of a third-party USB subsystem (USB2SS) operating at USB2.0 speeds (480Mb/s), either of which can be independently configured to act as a USB Host or a USB Device. SuperSpeed (5.0 Gb/s) operation is not supported in either operational mode.

This document serves to describe the integration of this third-party USB subsystem and should not be considered sufficient for those wishing to modify the existing Linux USB driver(s) or create a new driver to support this controller implementation. For those who do wish to substantially modify the existing Linux USB driver(s), or create new drivers, contact your TI sales representative for more information on how to obtain the third-party IP databook under NDA.

16.1.1 Features

The USB 2.0 subsystem, supports the following USB features:

- Operational modes:
  - Supports USB 2.0 Host mode at High-Speed (HS, 480 Mbps), Full-Speed (FS, 12 Mbps), and Low-Speed (LS, 1.5 Mbps)
  - Supports USB 2.0 Device mode at High-Speed (HS, 480 Mbps), and Full-Speed (FS, 12 Mbps). Low-Speed is not supported in Device mode.
  - Supports all modes of transfers - Control, Bulk, Interrupt, and Isochronous.
- A DRD (Dual-Role-Device - Host or Device) USB controller with the following features:
  - Compatible to the xHCI 1.0 specification in Host mode
  - Compatible with the USB 2.0 specification in Device mode
  - Supports 15 IN (Receive), 15 OUT (Transmit) endpoints (EPs), and one EP0 endpoint which is bidirectional
  - Internal DMA controller
  - Descriptor caching and data pre-fetching ensures high performance
  - Dynamic FIFO memory allocation for all endpoints
- Operation flexibility
  - Same programming model for HS, FS, and LS operation
  - Each controller instance can provide either USB Host or USB Device functionality
  - Multiple interrupt lines:
    - Four programmable interrupts
    - A MISC interrupt line for all miscellaneous events
- External requirements:
  - Needs an external Charge Pump for VBUS 5 V generation in Host mode. (Device mode does not require VBUS generation).

16.1.2 Unsupported Features

The following are USB features which are not supported:

- Battery Charger Support
- Accessory Charger Adaptor Support
- OTG functionality
- No Virtualization support
- SuperSpeed (5Gb/s) operation
- USB 2.0 ECN: Link Power Management (LPM)

16.2 Integration

Figure 16-1 shows the functional block diagram of the USB 2.0 subsystem (USB2SS), which includes a wrapper module, a USB controller module, a PHY module, external interfaces, and internal interfaces.

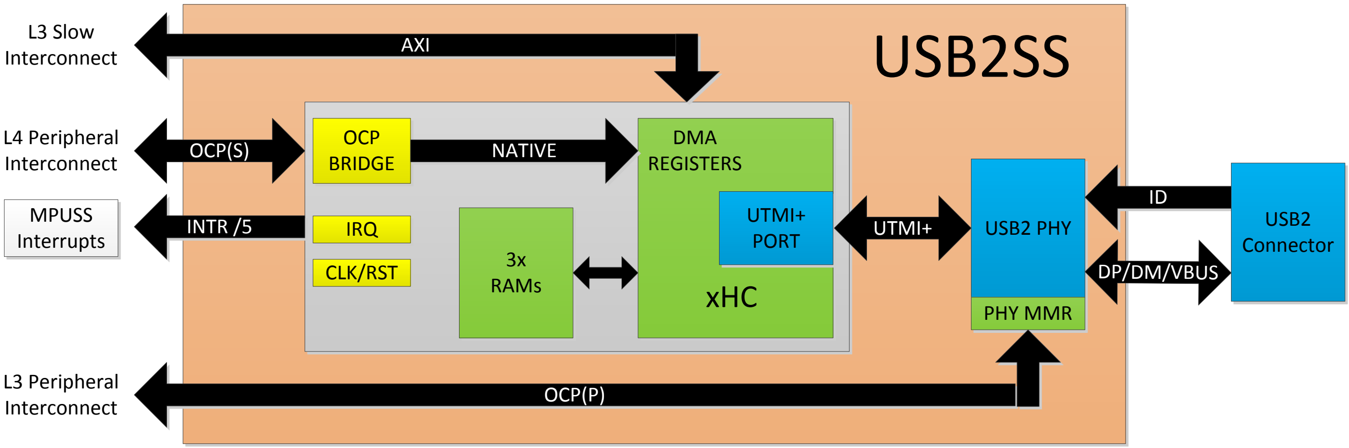


Figure 16-1. USB 2.0 Subsystem (USB2SS) Functional Block Diagram

- The wrapper module, which provides:
  - A bridge to the system bus
  - Local MMR to configure and control the USB controller
  - Interrupt management
- The USB Core with a Dual-Role-Device Controller module and internal RAMs:
  - Compatible to the xHCI 1.0 standard in Host mode
  - Support for either USB Host or USB Device modes
  - Internal DMA controller for high-bandwidth, low overhead data transfer
  - Three local RAMs supporting:
    - Dynamic Tx/Rx FIFO memory allocation for all EPs
    - Descriptor caching and data pre-fetching for high performance
  - Supports all transfer types - Control, Bulk, Interrupt, and Isochronous
  - Supports high-bandwidth ISO mode
  - Supports 15 IN, 15 OUT endpoints (EPs), and one EP0 endpoint which is bidirectional
- The integrated USB PHY module:
  - A USB2 PHY connected to the core via internal UTMI+ port
- External interfaces (towards device boundary):
  - To USB Connector:
    - DP/DM - a bidirectional signal pair for HS, FS, LS mode operation
    - VBUS - an analog pin for monitoring the voltage on VBUS.
    - ID - an input signal for operating mode (Host/Device) determination.
  - To an external Charge Pump for VBUS 5V generation:
    - DRVVBUS - an active high output that controls an external 5V charge pump. The core uses this output to automatically ensure that VBUS has 5V supplied by the charge pump during Host mode operation.

- Internal interface (towards internal system):
  - An OCP slave interface for MMR transactions to/from the controller.
  - An OCP slave interface for MMR transactions to/from the PHY.
  - An AXI master interface for high-bandwidth DMA transactions.
  - Interrupt interface:
    - USB_MAINn_INT[3:0] - 4 programmable interrupts associated with DMA traffic
    - USB_MISCINT - single interrupt line for misc events

16.2.1 USB Clock and Reset Management

The USB2SS contains several functional clock domains: BUS, PHYMMR, and UTMI+ Interface.

Table 16-1 provides the details of the clock drivers and the control of the clocks.

Table 16-1. USB2SS Clock Sources and Clock Control

Clock Signal	Max Frequency	Reference / Source	Comments
ocp_clk XHCImaster/slave interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l3s_gclk From PRCM
phy_clk Phy slave interface clock	100 MHz	CORE_CLKOUTM4 / 4	pd_per_l4s_gclk From PRCM
suspend_clk Phy low power functional clock	125 MHz	CLK_M_OSC	pd_usb_otg_ssrefclk From PRCM
phy_other_refclk960m UTMI source clock	960 MHz	USB_PHYCLK	DCOCLKLDO from Per PLL
phy_other_trefclk Phy reference clock	50 MHz	CLK_M_OSC	
phy_other_sparein[0] Phy auto-resume logic wakeup clock	32 KHz	CLK32K_RTC	pd_wakeup_usbphy_32khz_gclk From PRCM

16.2.2 USB Pin List

The names of the pins used by the USB2SS with descriptions are shown in Table 16-2 and Table 16-3.

Table 16-2. USB Signal Pins Description

Pin Name	IOZP	IPD/IPU	Nominal Voltage	Description
USBn_DP	IOZ		3.3V	USB 2.0 and 1.1 specification-compliant signal pins. They are HS/FS/LS bidirectional differential data pins.
USBn_DM	IOZ		3.3V	

Table 16-3. USB Control, Configuration, and Monitor Signal Pins

Pin Name	IOZP	Nominal Voltage	Description
USBn_DRVVBUS	O		An active-high digital output signal for VBUS power supply. Used to enable an external charge pump to supply +5V power to the VBUS port of the device as well as the VBUS port of the USB receptacle, when appropriate per the USB standard. Output is 0 at reset.
USBn_VBUS	A	5.0V	An analog input for monitoring the voltage on VBUS.
USBn_ID	A		Used to determine the operational mode (Host/Peripheral) of the controller. When grounded, the controller will operate as a USB Host. When left floating, the controller will operate as a USB Peripheral. NOTE: This pin should never be connected to a voltage source.
USBn_CE	O	3.3V	An active high digital output for PHY charge enable.

16.3 Use Cases

This is a standard USB 2.0 module, and is optimized for following applications and systems:

- Portable electronic devices
- High-bandwidth applications

It supports all typical USB connections, and Table 16-4 shows some examples.

Table 16-4. Typical Use Cases In Terms of Connections

Connectors (Receptacle)	Signals to Use	SS	HS/FS	LS (Host only)	Comments
USB 2.0 Micro-AB	DP, DM, VBUS, ID	N	Y	Y	Support Host or Device operation, depending on state of the ID pin.
USB 2.0 Type-A	DP, DM, VBUS	N	Y	Y	Support HS/FS/LS Host
USB 2.0 Type-B	DP, DM, VBUS	N	Y	N	Only used for USB2.0 Device (no LS)

16.3.1 USB Operational Mode Determination

As the USB controller modules present in this device are DRD (Dual-Role Devices), they can support operation as either a USB Host or a USB Device. The operational mode determination is made based on the state of the USBn_ID pin; when this pin is grounded, the controller will operate as a USB Host, when this pin is left floating, the controller assumes the role of a USB Device. For implementations that do not require DRD functionality, the USBn_ID pin can either be left floating or can be grounded in the board design depending on the static role required. For implementations that do require DRD functionality, the USBn_ID pin should be connected directly to the corresponding ID pin on a USB Micro-AB socket. In doing so, the USBn_ID pin will be correctly terminated (open or grounded) depending on the cable attached and the controller will enter Host or Device mode accordingly. Refer to Section 16.3.2 for more details.

16.3.2 Typical Pin Connections of AM437x Device

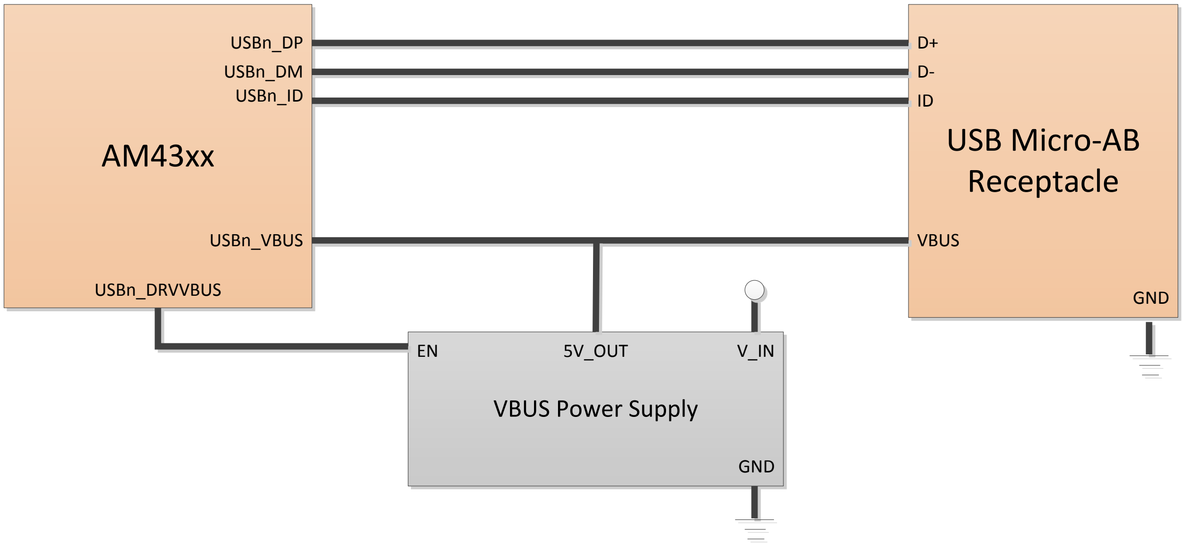


Figure 16-2. USB Dual-Role (Host or Device)

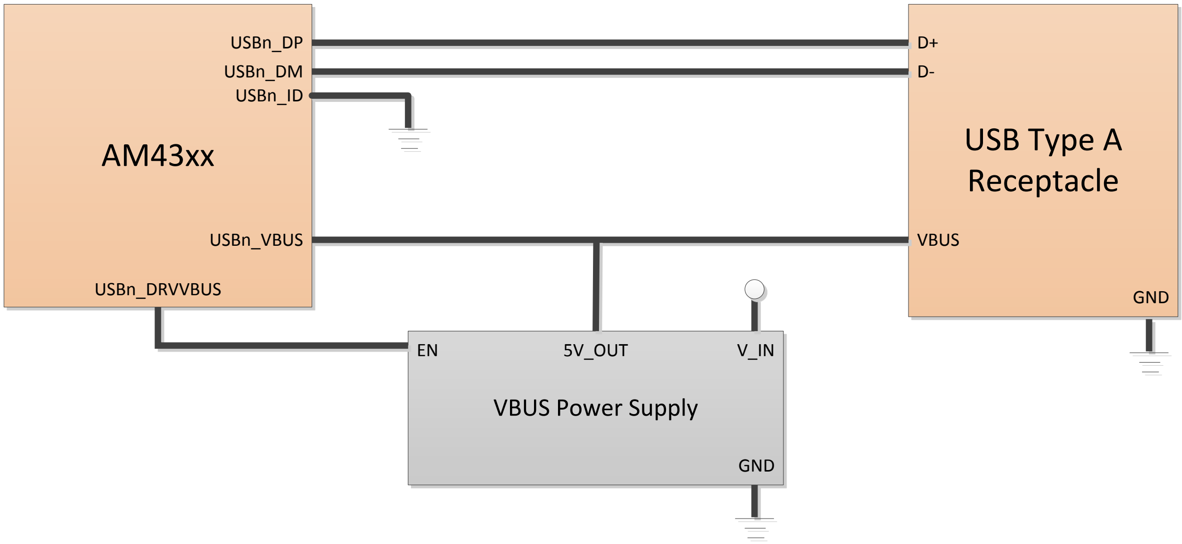


Figure 16-3. USB Host Only

Figure 16-4. USB Device Only

16.3.3 VBUS Voltage Sourcing Control

When either of the USB controllers assumes the role of a host, the controller is required to supply 5V to an attached device through its VBUS line. In order to achieve this task, the USB controller requires the use of external power logic (or a charge pump) capable of sourcing 5V power. The USB_DRVVBUS pin is used as a control signal to enable/disable this external power logic to either source or disable power on the VBUS line. The control on the USB_DRVVBUS is automatic and is handled by the USB controller. The control should be transparent to the user so long as the proper hardware connection and software initialization are in place. The USB controller drives the USB_DRVVBUS signal high when it assumes the role of a host while the controller is in session. When assuming the role of a device, the controller drives the USB_DRVVBUS signal low disabling the external charge pump/power logic; hence, no power is driven on the VBUS line (in this case, power is expected to be provided by the external host).

Note that both USB controllers are self-powered and the device does not rely on the voltage on the VBUS line sourced by an external host for controller operation when assuming the role of a device. The voltage present on the VBUS line is used to identify the presence of a Host. The USB PHY continually monitors the voltage on the VBUS and reports the status to USB controller.

16.3.4 Pull-up/Pull-down Resistors

As the USB controllers are dual role controllers, capable of assuming a role of a host or device, the required pull-up/pull-down resistors cannot exist external to the device. These pull-up/pulldown resistors exist internal to the device, within the PHY to be more specific, and are enabled or disabled based on the role the controller assumes allowing for dynamic hardware configuration. When assuming the role of a host, the data lines are pulled low by the PHY enabling the internal 15KΩ resistors. When assuming the role of a device the required 1.5KΩ pull-up resistor on the D+ line is enabled automatically to signify the USB capability to the external host as a FS device (HS operation is negotiated during reset bus condition).

16.3.5 Clock, PLL, and PHY Initialization

Prior to configuring the USB Module Registers, the USB Subsystem and PHY are required to be released from reset, relevant interconnects and clocks must be enabled and the PHY itself must be configured.

16.4 Reference Documentation

Universal Serial Bus 3.0 Specification, Revision 1.0, USB-IF, November 12, 2008

PHY Interface for the PCI Express and USB 3.0 Architecture, Version 3.00, Intel Corp.

USB 2.0 Transceiver Macrocell Interface (UTMI) Specification, Revision 1.05, Intel Corp. March 29, 2001

UTMI+ Specification, Revision 1.0, ULPI Working Group, February 25, 2004

eXtensible Host Controller Interface For Universal Serial Bus (xHCI), Revision 1.0 with errata to 6/13/11, Intel Corp., June 13, 2011 (Request from USB-IF)

Chapter 17

SPRUHL7H–April 2014–Revised January 2018

Multimedia Card (MMC)

This chapter describes the MMC of the device.

Topic **Page**

17.1 Introduction

17.1.1 MMCSD Features

The general features of the MMCSD host controller IP are:

- Built-in 1024-byte buffer for read or write
- Two DMA channels, one interrupt line
- Clock support
  - 96-MHz functional clock source input
  - up to 384Mbit/sec (48MByte/sec) in MMC mode 8-bit data transfer
  - up to 192Mbit/sec (24MByte/sec) in High-Speed SD mode 4-bit data transfer
  - up to 24Mbit/sec (3MByte/sec) in Default SD mode 1-bit data transfer
- Support for SDA 3.0 Part A2 programming model
- Serial link supports full compliance with:
  - MMC command/response sets as defined in the MMC standard specification v4.3.
  - SD command/response sets as defined in the SD Physical Layer specification v2.00
  - SDIO command/response sets and interrupt/read-wait suspend-resume operations as defined in the SD part E1 specification v 2.00
  - SD Host Controller Standard Specification sets as defined in the SD card specification Part A2 v2.00

17.1.2 Unsupported MMCSD Features

The MMCSD module features not supported in this device are shown in Table 17-1.

Table 17-1. Unsupported MMCSD Features

Feature	Reason
MMC Out-of-band interrupts	MMC_OBI input tied low
Master DMA operation	Disabled through synthesis parameter
Card Supply Control (MMCSD(1-2))	Signal not pinned out
Dual Data Rate (DDR) mode	Timing not supported

17.2 Integration

This device contains three instances of the Multimedia Card (MMC), Secure Digital (SD), and Secure Digital I/O (SDIO) high speed interface module (MMCHS). The controller provides an interface to an MMC, SD memory card or SDIO card.

The application interface is responsible for managing transaction semantics; the MMC/SDIO host controller deals with MMC/SDIO protocol at transmission level, packing data, adding CRC, start/end bit and checking for syntactical correctness. Figure 17-1 through Figure 17-3 below show examples of systems using the MMCSD controller. Note that the power switch control is only available on the MMCSD0 interface.

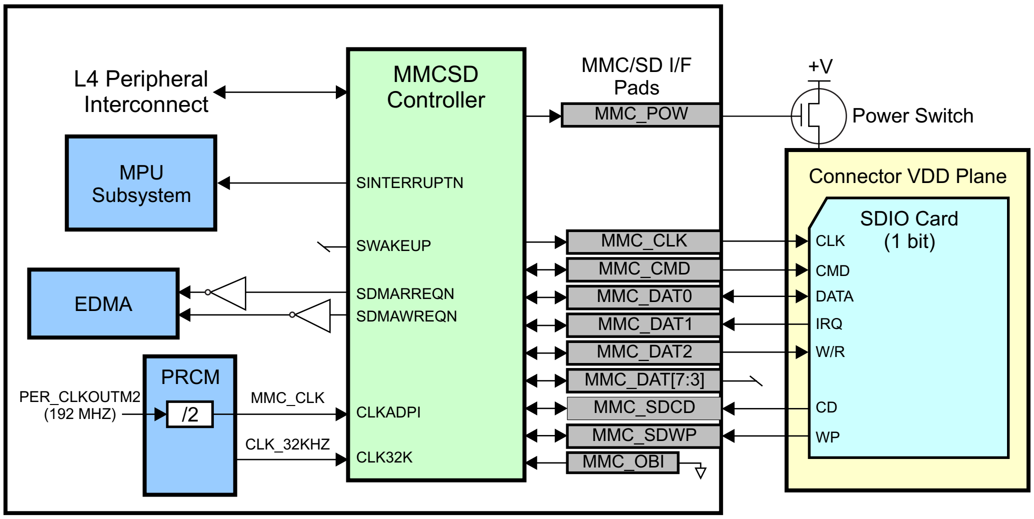


Figure 17-1. MMCSD Module SDIO Application

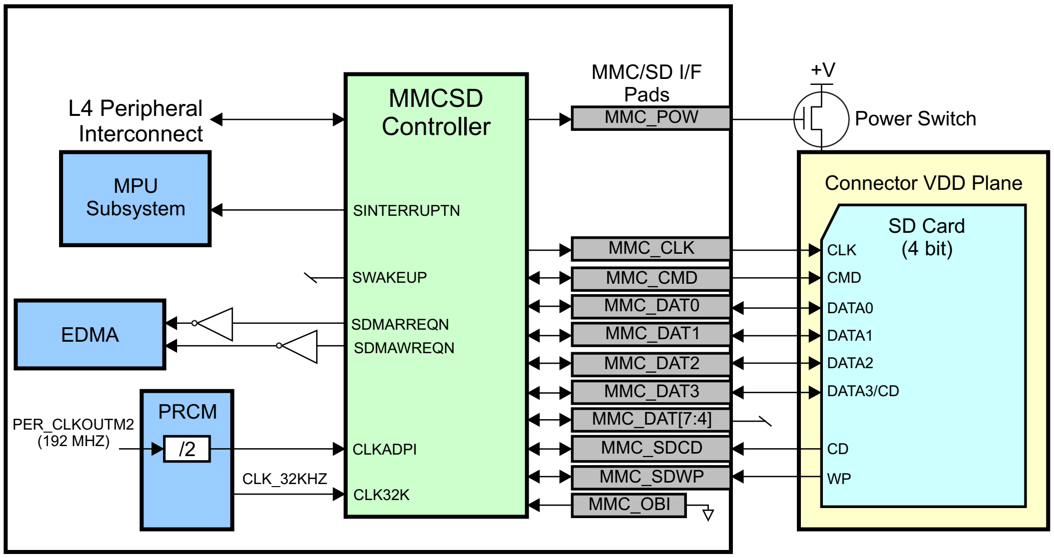


Figure 17-2. MMCSD (4-bit) Card Application

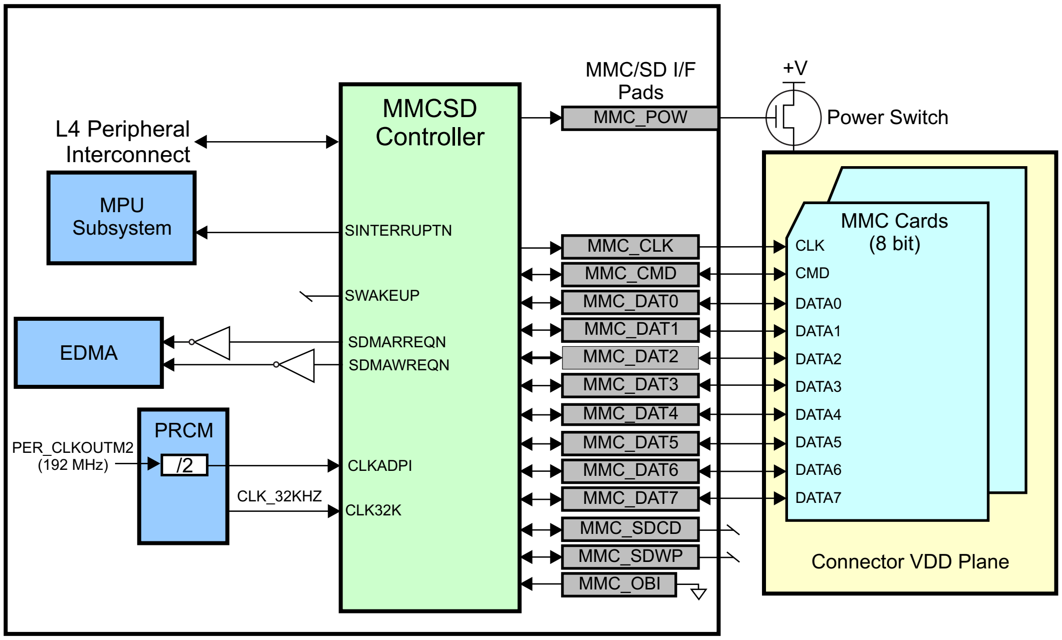


Figure 17-3. MMCSD Module MMC Application

17.2.1 MMCSD Connectivity Attributes

The general connectivity attributes for the three MMCSD modules are shown in Table 17-2.

Table 17-2. MMCSD Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L4LS_GCLK (OCP) PD_PER_MMC_FCLK (Func) CLK_32KHZ (Debounce)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	1 interrupt per instance to MPU Subsystem (MMCSDxINT)
DMA Requests	2 DMA requests per instance to EDMA (SDTXEVTx, SDRXEVTx) (Active low, need to be inverted in glue logic)
Physical Address	L4 Peripheral slave port

17.2.2 MMCSD Clock and Reset Management

The MMCSD controller has separate bus interface and functional clocks. The debounce clock is created by dividing the 48-MHz (24 MHz @ OPP50) clock in the PRCM by two and then dividing the resulting 24-MHz (12 MHz @ OPP50) clock by a fixed 732.4219 (366.2109 @ OPP50) in the Control Module to get a 32-kHz clock. This clock is fed back into the PRCM for clock gating. (See the CTRL_CLK32KDIVRATIO register in Chapter 7, *Control Module*, for more details).

Table 17-3. MMCSD Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
CLK Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk from PRCM
CLKADPI Functional clock	48 MHz	PER_CLKOUTM2 / 4	pd_per_mmc_fclk from PRCM
CLK32 Input de-bounce clock	32.768 KHz	CLK_24 / 732.4219	clk_32KHz from PRCM

17.2.3 MMCSD Pin List

The MMCSD interface pins are summarized in Table 17-4.

Table 17-4. MMCSD Pin List

Pin	Type	Description
MMC*x*_CLK	I/O(1)	MMC/SD serial clock output
MMC*x*_CMD	I/O	MMC/SD command signal
MMC*x*_DAT0	I/O	MMC/SD data signal
MMC*x*_DAT1	I/O	MMC/SD data signal, SDIO interrupt input
MMC*x*_DAT2	I/O	MMC/SD data signal, SDIO read wait output
MMC*x*_DAT[7:3]	I/O	MMC/SD data signals
MMC*x*_POW	O	MMC/SD power supply control (MMCSD0 only)
MMC*x*_SDCD	I	SD card detect (from connector)
MMC*x*_SDWP	I	SD write protect (from connector)
MMC*x*_OBI	I	MMC out of band interrupt

(1) These signals are also used as inputs to re-time or sync data. The associated CONF_<module>_pin_RXACTIVE bit for these signals must be set to 1 to enable the inputs back to the module. It is also recommended to place a 33-ohm resistor in series (close to the processor) on each of these signals to avoid signal reflections.

The direction of the data lines depends on the selected data transfer mode as summarized in Table 17-5.

Table 17-5. DAT Line Direction for Data Transfer Modes

	MMC/SD 1-bit mode	MMC/SD 4-bit mode	MMC/SD 8-bit mode	SDIO 1-bit mode	SDIO 4-bit mode
DAT[0]	I/O	I/O	I/O	I/O	I/O
DAT[1]	I(1)	I/O	I/O	I(2)	I/O or I(2)
DAT[2]	I(1)	I/O	I/O	I/O(3)	I/O or O(3)
DAT[3]	I(1)	I/O	I/O	I(1)	I/O
DAT[4]	I(1)	I(1)	I/O	I(1)	I(1)
DAT[5]	I(1)	I(1)	I/O	I(1)	I(1)
DAT[6]	I(1)	I(1)	I/O	I(1)	I(1)
DAT[7]	I(1)	I(1)	I/O	I(1)	I(1)

(1) Hi-Z state to avoid bus conflict.
(2) To support incoming interrupt from the SDIO card.
(3) To support read wait to the SDIO card. By default it is Input, Output only in read wait period.

The direction of the MMCSD data buffers are controlled by ADPDATDIROQ signals. ADPDATDIROQ[i] = 1 sets the corresponding DAT signal(s) in read position (input) and ADPDATDIROQ[i] = 0 sets the corresponding DAT signal(s) in write position (output). Additionally, the ADPDATDIRLS signals are provided (with opposite polarity) to control the direction of external level shifters. The value of these control signals for the various data modes are summarized in Table 17-6.

Table 17-6. ADPDATDIROQ and ADPDATDIRLS Signal States

	MMC/SD 1-bit mode	**MMC/SD 4-bit mode**	**MMC/SD 8-bit mode**	**SDIO 1-bit mode**	**SDIO 4-bit mode**
DAT[0]	ADPDATDIRLS[0] = 0 / 1 ADPDATDIROQ[0] = 1 / 0	ADPDATDIRLS[0] = 0 / 1 ADPDATDIROQ[0] = 1 / 0	ADPDATDIRLS[0] = 0 / 1 ADPDATDIROQ[0] = 1 / 0	ADPDATDIRLS[0] = 0 / 1	ADPDATDIRLS[0] = 0 / 1 ADPDATDIROQ[0] = 1 / 0
DAT[2]	ADPDATDIRLS[2] = 0 ADPDATDIROQ[2] = 1	ADPDATDIRLS[2] = 0 / 1 ADPDATDIROQ[2] = 1 / 0	ADPDATDIRLS[2] = 0 / 1 ADPDATDIROQ[2] = 1 / 0	ADPDATDIRLS[2] = 0 / 1 ADPDATDIROQ[2] = 1 / 0	ADPDATDIRLS[2] = 0 / 1 ADPDATDIROQ[2] = 1 / 0
DAT[1]	ADPDATDIRLS[1] = 0 ADPDATDIROQ[1] = 1	ADPDATDIRLS[1] = 0 / 1 ADPDATDIROQ[1] = 1 / 0	ADPDATDIRLS[1] = 0 / 1 ADPDATDIROQ[1] = 1 / 0	ADPDATDIRLS[1] = 0 ADPDATDIROQ[1] = 1	ADPDATDIRLS[1] = 0 / 1 ADPDATDIROQ[1] = 1 / 0
DAT[3]					
DAT[4]	ADPDATDIRLS[3] = 0 ADPDATDIROQ[3] = 1	ADPDATDIRLS[3] = 0 ADPDATDIROQ[3] = 1	ADPDATDIRLS[3] = 0 / 1 ADPDATDIROQ[3] = 1 / 0	ADPDATDIRLS[3] = 0 ADPDATDIROQ[3] = 1	ADPDATDIRLS[3] = 0 ADPDATDIROQ[3] = 1
DAT[5]					
DAT[6]					
DAT[7]					

ADPDATIRLSx = 0 for input and 1 for output — these signals are not pinned out on this device.

ADPDATIROQx = 1 for output and 1 for input.

Grayed cells indicate that the data line is not used in the selected transfer mode.

17.3 Functional Description

One MMC/SD/SDIO host controller can support one MMC memory card, one SD card, or one SDIO card.

Other combinations (for example, two SD cards, one MMC card, and one SD card) are not supported through a single controller.

17.3.1 MMC/SD/SDIO Functional Modes

17.3.1.1 MMC/SD/SDIO Connected to an MMC, an SD Card, or an SDIO Card

Figure 17-4 shows the MMC/SD/SDIO1 and MMC/SD/SDIO2 host controllers connected to an MMC, an SD, or an SDIO card and its related external connections.

Figure 17-4. MMC/SD1/2 Connectivity to an MMC/SD Card

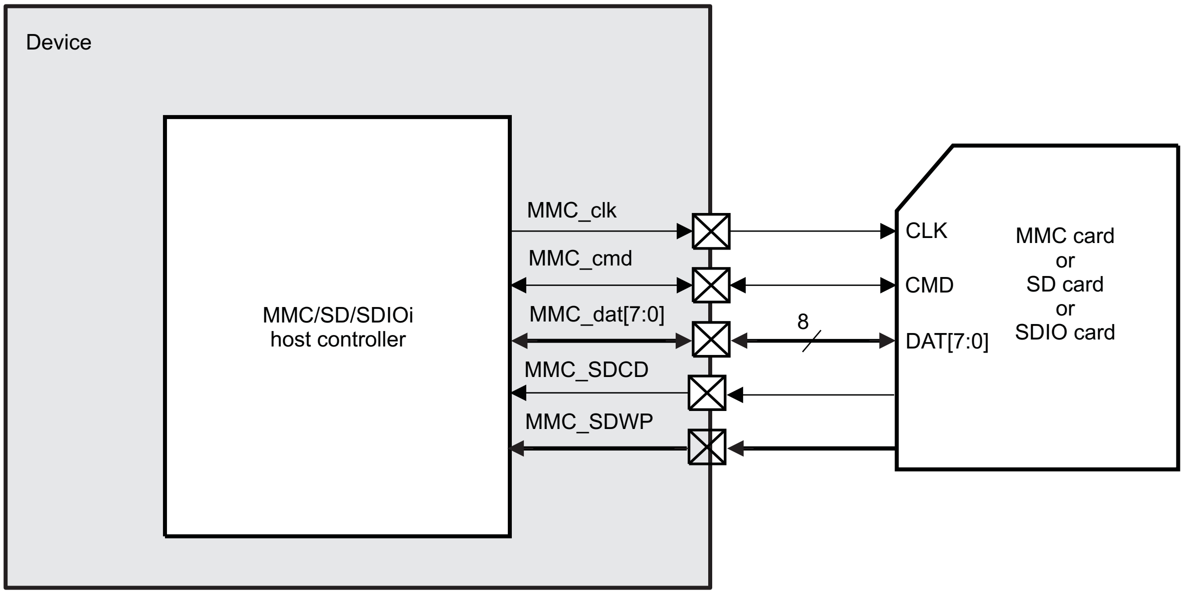


Figure 17-5. MMC/SD0 Connectivity to an MMC/SD Card

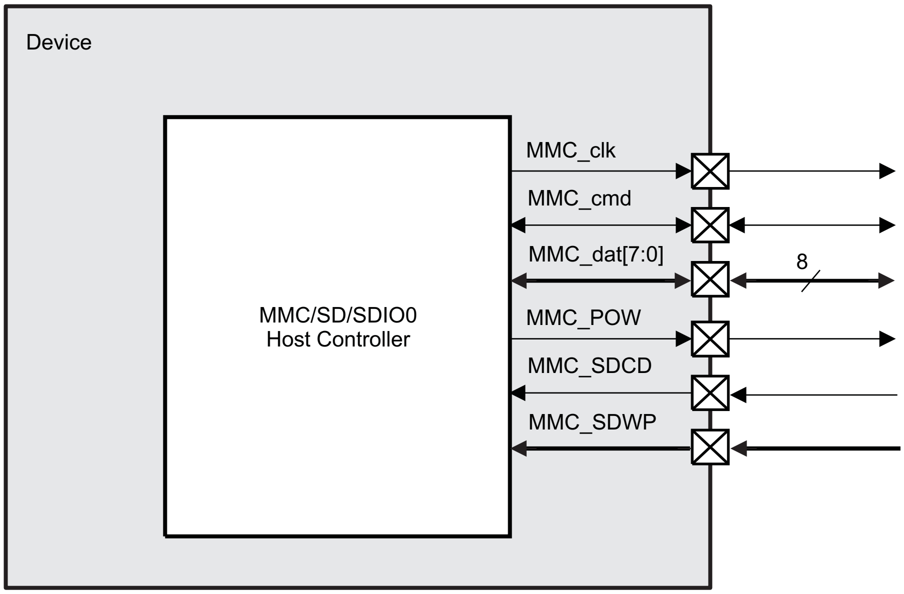


Figure 17-5 shows the MMC/SD/SDIO0 host controller connected to an MMC, SD, or SDIO card and its related external connections. Note that MMC/SD/SDIO0 uses the same signals as MMC/SD/SDIO1 and 2 but adds MMC_POW.

The following MMC/SD/SDIO controller pins are used

- **MMC_CMD** This pin is used for two-way communication between the connected card and the MMC/SD/SDIO controller. The MMC/SD/SDIO controller transmits commands to the card and the memory card drives responses to the commands on this pin.
- **MMC_DAT7-0** Depending on which type of card you are using, you may need to connect 1, 4, or 8 data lines. The number of DAT pins (the data bus width) is set by the Data Transfer Width (DTW) bit in the MMC control register (SD_HCTL). For more information, see Section 17.5.1, *MMCSD Registers*.
- **MMC_CLK** This pin provides the clock to the memory card from the MMC/SD controller.
- **MMC_POW** Used for MMC/SD card's cards on/off power supply control. When high, denotes power-on condition.
- **MMC_SDCD** This input pin serves as the MMC/SD/SDIO card detect. This signal is received from a mechanical switch on the slot.
- **MMC_SDWP** This input pin is used for the SD/SDIO card's write protect. This signal is received from a mechanical protect switch on the slot (system dependant). Applicable only for SD and SDIO cards that have a mechanical sliding tablet on the side of the card.

  Note: The MMC_CLK pin functions as an output but must be configured as an I/O to internally loopback the clock to time the inputs.

Table 17-7 provides a summary of these pins.

Table 17-7. MMC/SD/SDIO Controller Pins and Descriptions

Pin	Type	1-Bit Mode	4-Bit Mode	8-Bit Mode	Reset Value
MMC_CLK[1]	O	Clock Line	Clock Line	Clock Line	High impedance
MMC_CMD	I/O	Command Line	Command Line	Command Line	High impedance
MMC_DAT0	I/O	Data Line 0	Data Line 0	Data Line 0	0
MMC_DAT1	I/O	(not used)	Data Line 1	Data Line 1	0
MMC_DAT2	I/O	(not used)	Data Line 2	Data Line 2	0
MMC_DAT3	I/O	(not used)	Data Line 3	Data Line 3	0
MMC_DAT4	I/O	(not used)	(not used)	Data Line 4	0
MMC_DAT5	I/O	(not used)	(not used)	Data Line 5	0
MMC_DAT6	I/O	(not used)	(not used)	Data Line 6	0
MMC_DAT7	I/O	(not used)	(not used)	Data Line 7	0

(1) The MMC_CLK pin functions as an output but must be configured as an I/O to internally loopback the clock to time the inputs.

17.3.1.2 Protocol and Data Format

The bus protocol between the MMC/SD/SDIO host controller and the card is message-based. Each message is represented by one of the following parts:

Command: A command starts an operation. The command is transferred serially from the \MMC/SD/SDIO host controller to the card on the mmc_cmd line.

Response: A response is an answer to a command. The response is sent from the card to the MMC/SD/SDIO host controller. It is transferred serially on the mmc_cmd line.

Data: Data are transferred from the MMC/SD/SDIO host controller to the card or from a card to the MMC/SD/SDIO host controller using the DATA lines.

Busy: The mmc_dat0 signal is maintained low by the card as far as it is programming the data received.

CRC status: CRC result is sent by the card through the mmc_dat0 line when executing a write transfer. In the case of transmission error, occurring on any of the active data lines, the card sends a negative CRC status on mmc_dat0. In the case of successful transmission, over all active data lines, the card sends a positive CRC status on mmc_dat0 and starts the data programming procedure.

17.3.1.2.1 Protocol

There are two types of data transfer:

- Sequential operation
- Block-oriented operation

There are specific commands for each type of operation (sequential or block-oriented). See the *Multimedia Card System Specification*, the *SD Memory Card Specifications*, and the *SDIO Card Specification, Part E1* for details about commands and programming sequences supported by the MMC, SD, and SDIO cards.

CAUTION

Stream commands are supported only by MMC cards.

Figure 17-6 and Figure 17-7 show how sequential operations are defined. Sequential operation is only for 1-bit transfer and initiates a continuous data stream. The transfer terminates when a stop command follows on the mmc_cmd line.

Figure 17-6. Sequential Read Operation (MMC Cards Only)

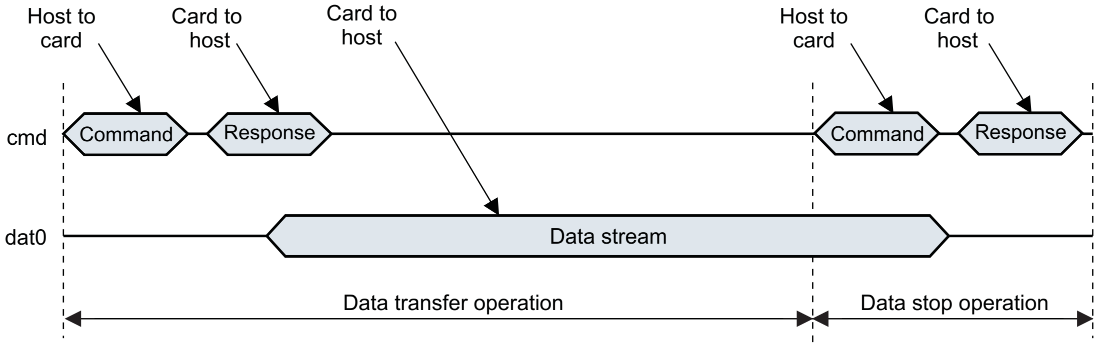


Figure 17-7. Sequential Write Operation (MMC Cards Only)

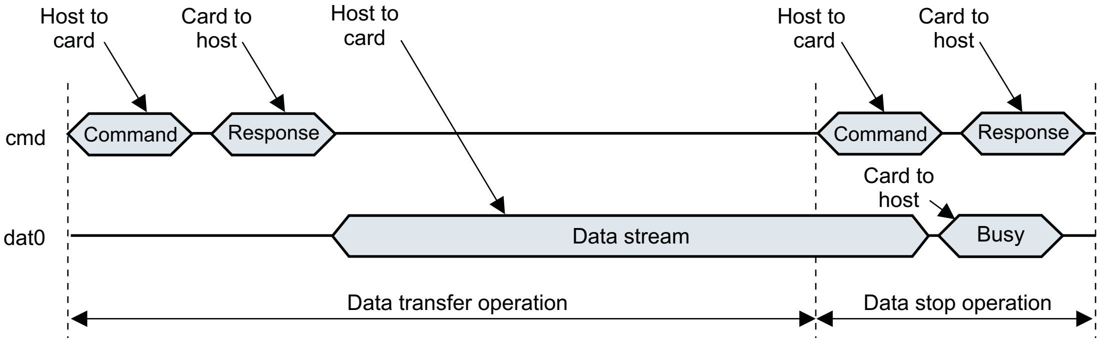


Figure 17-8 and Figure 17-9 show how multiple block-oriented operations are defined. A multiple block-oriented operation sends a data block plus CRC bits. The transfer terminates when a stop command follows on the mmc_cmd line. These operations are available for all kinds of cards.

Figure 17-8. Multiple Block Read Operation (MMC Cards Only)

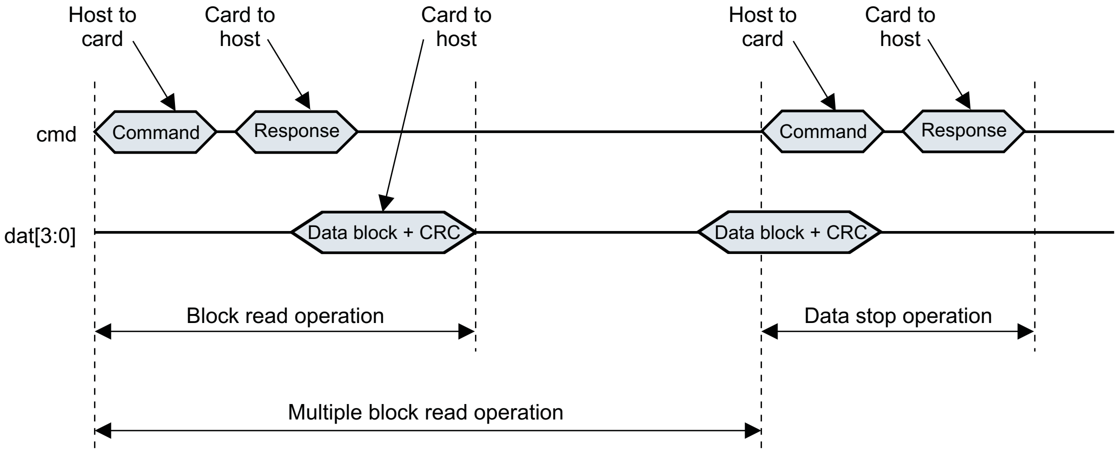
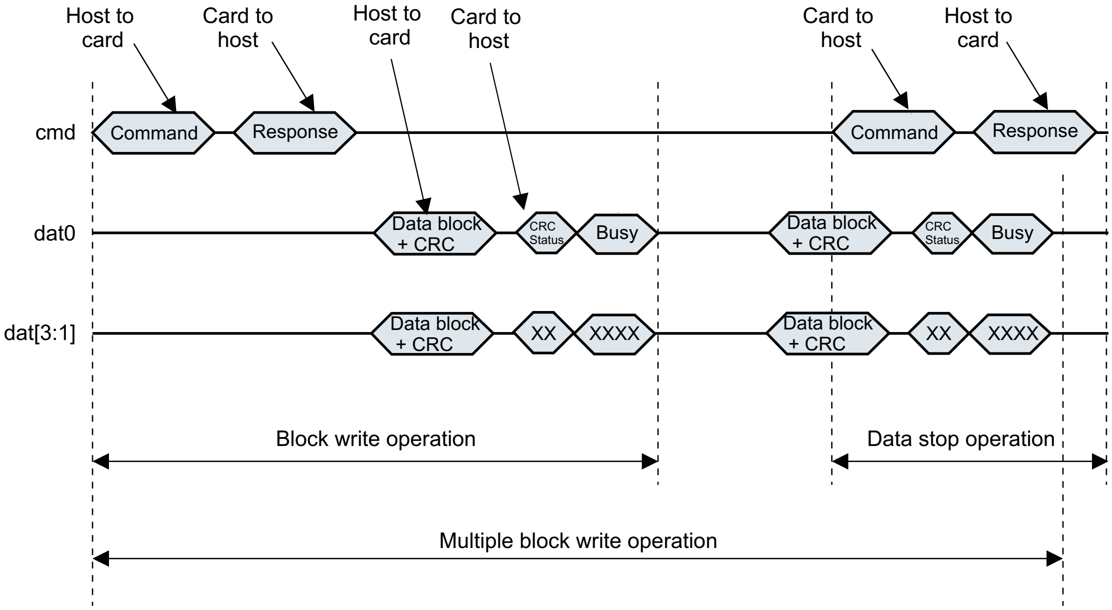


Figure 17-9. Multiple Block Write Operation (MMC Cards Only)



NOTE:

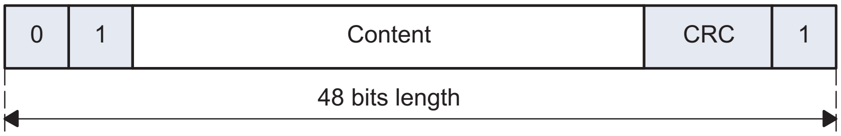
1. The card busy signal is not always generated by the card; the previous examples show a particular case.
2. It is the software's responsibility to do a software reset after a data timeout to ensure that mmc_clk is stopped. The software reset is done by setting bit 26 in the SD_SYSCTL register to 1.
3. For multiblock transfer, and especially for MMC cards, you can abort a transfer without using a stop command. Use a CMD23 before a data transfer to define the number of blocks that will be transferred, then the transfer stops automatically after the last block (provided the MMC card supports this feature).

17.3.1.2.2 Data Format

Coding Scheme for Command Token

Command packets always start with 0 and end with 1. The second bit is a transmitter bit1 for a host command. The content is the command index (coded by 6 bits) and an argument (for example, an address), coded by 32 bits. The content is protected by 7-bit CRC checksum (see Figure 17-10).

Figure 17-10. Command Token Format



Coding Scheme for Response Token

Response packets always start with 0 and end with a 1. The second bit is a transmitter bit0 for a card response. The content is different for each type of response (R1, R2, R3, R4, R5, and R6) and the content is protected by 7-bit CRC checksum. Depending on the type of commands sent to the card, the SD_CMD register must be configured differently to avoid false CRC or index errors to be flagged on command response (see Table 17-8). For more details about response types, see the *Multimedia Card System Specification*, the *SD Memory Card Specification*, or the *SDIO Card Specification*.

Table 17-8. Response Type Summary(1)

Response Type SD_CMD[17:16] RSP_TYPE	Index Check Enable SD_CMD[20] CICE	CRC Check Enable SD_CMD[19] CCCE	Name of Response Type
00	0	0	No Response
01	0	1	R2
10	0	0	R3 (R4 for SD cards)
10	1	1	R1, R6, R5 (R7 for SD cards)
11	1	1	R1b, R5b

(1) The MMC/SD/SDIO host controller assumes that both clocks may be switched off, whatever the value set in the SD_SYSCONFIG[9:8] CLOCKACTIVITY bit.

Figure 17-11 and Figure 17-12 depict the 48-bit and 136-bit response packets.

Figure 17-11. 48-Bit Response Packet (R1, R3, R4, R5, R6)

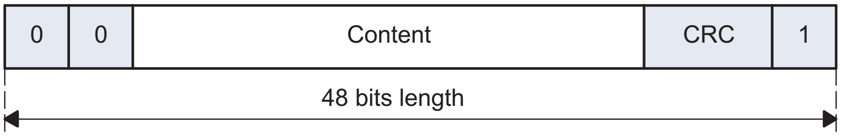
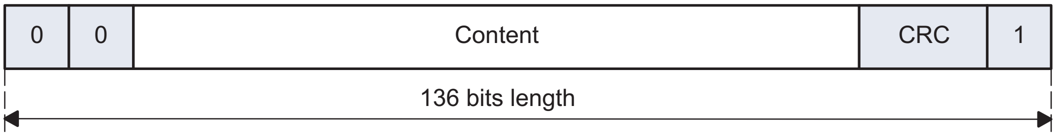


Figure 17-12. 136-Bit Response Packet (R2)



Coding Scheme for Data Token

Data tokens always start with 0 and end with 1 (see Figure 17-13, Figure 17-14, Figure 17-15, and Figure 17-16).

Figure 17-13. Data Packet for Sequential Transfer (1-Bit)

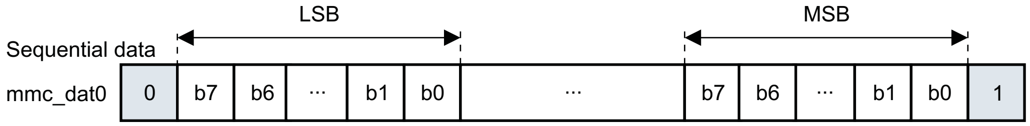


Figure 17-14. Data Packet for Block Transfer (1-Bit)

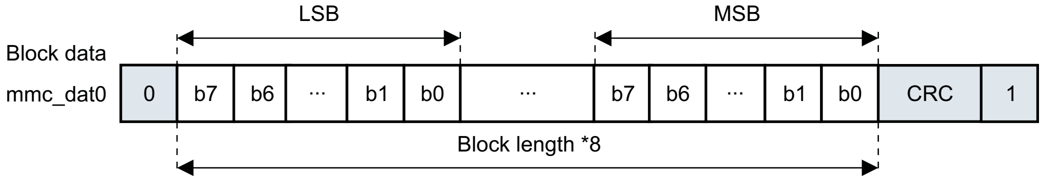


Figure 17-15. Data Packet for Block Transfer (4-Bit)

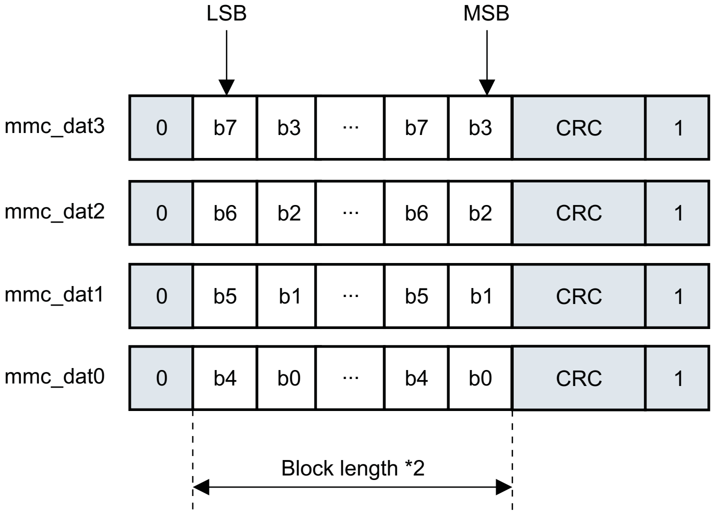
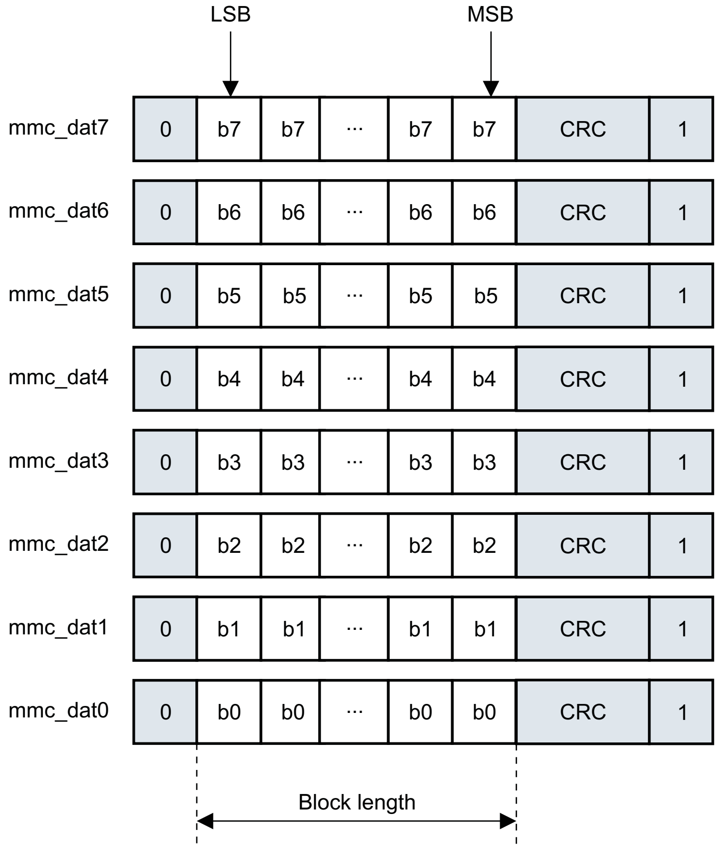


Figure 17-16. Data Packet for Block Transfer (8-Bit)



17.3.2 Resets

17.3.2.1 Hardware Reset

The module is reinitialized by the hardware.

The SD_SYSSTS[0] RESETDONE bit can be monitored by the software to check if the module is ready-to-use after a hardware reset.

This hardware reset signal has a global reset action on the module. All configuration registers and all state machines are reset in all clock domains.

This hardware reset signal has a global reset action on the module. All configuration registers and all state-machines are reset in all clock domains.

17.3.2.2 Software Reset

The module is reinitialized by software through the SD_SYSCONFIG[1] SOFTRESET bit. This bit has the same action on the module logic as the hardware signal except for:

- Debounce logic
- SD_PSTATE, SD_CAPA, and SD_CUR_CAPA registers (see corresponding register descriptions)

The SOFTRESET bit is active high. The bit is automatically reinitialized to 0 by the hardware. The SD_SYSCTL[24] SRA bit has the same action as the SOFTRESET bit on the design.

The SD_SYSSTS[0] RESETDONE bit can be monitored by the software to check if the module is ready-to-use after a software reset.

Moreover, two partial software reset bits are provided:

- SD_SYSCTL[26] SRD bit
- SD_SYSCTL[25] SRC bit

These two reset bits are useful to reinitialize data or command processes respectively in case of line conflict. When set to 1, a reset process is automatically released when the reset completes:

- The SD_SYSCTL[26] SRD bit resets all finite state-machines and status management that handle data transfers on both the interface and functional side.
- The SD_SYSCTL[25] SRC bit resets all finite state-machines and status management that handle command transfers on both the interface and functional side.

NOTE: If **any** of the clock inputs are not present for the MMC/SD/SDIO peripheral, the software reset will not complete.

17.3.3 Power Management

The MMC/SD/SDIO host controller can enter into different modes and save power:

- Normal mode
- Idle mode

The two modes are mutually exclusive (the module can be in normal mode or in idle mode). The MMC/SD/SDIO host controller is compliant with the PRCM module handshake protocol. When the MMC/SD/SDIO power domain is off, the only way to wake up the power domain and different MMC/SD/SDIO clocks is to monitor the mmc_dat1 input pin state via a different GPIO line for each MMC/SD/SDIO interface.

17.3.3.1 Normal Mode

The autogating of interface and functional clocks occurs when the following conditions are met:

- The SD_SYSCONFIG[0] AUTOIDLE bit is set to 1.
- There is no transaction on the MMC interface.

The autogating of interface and functional clocks stops when the following conditions are met:

- A register access occurs through the L3 (or L4) interconnect.
- A wake-up event occurs (an interrupt from a SDIO card).
- A transaction on the MMC/SD/SDIO interface starts.

Then the MMC/SD/SDIO host controller enters in low-power state even if SD_SYSCONFIG[0] AUTOIDLE is cleared to 0. The functional clock is internally switched off and only interconnect read and write accesses are allowed.

17.3.3.2 Idle Mode

The clocks provided to MMC/SD/SDIO are switched off upon a PRCM module request. They are switched back upon module request. The MMC/SD/SDIO host controller complies with the PRCM module handshaking protocol:

- Idle request from the system power manager
- Idle acknowledgment from the MMC/SD/SDIO host controller

The idle acknowledgment varies according to the SD_SYSCONFIG[4:3] SIDLEMODE bit field:

- 0: Force-idle mode. The MMC/SD/SDIO host controller acknowledges the system power manager request unconditionally.
- 1h: No-idle mode. The MMC/SD/SDIO host controller ignores the system power manager request and behaves normally as if the request was not asserted.
- 2h: Smart-idle mode. The MMC/SD/SDIO host controller acknowledges the system power manager request according to its internal state.

- 3h: Reserved.

During the smart-idle mode period, the MMC/SD/SDIO host controller acknowledges that the OCP and Functional clocks may be switched off whatever the value set in the SD_SYSCONFIG[9:8] CLOCKACTIVITY field.

17.3.3.3 Transition from Normal Mode to Smart-Idle Mode

Smart-idle mode is enabled when the SD_SYSCONFIG[4:3] SIDLEMODE bit field is set to 2h or 3h. The MMC/SD/SDIO host controller goes into idle mode when the PRCM issues an idle request, according to its internal activity. The MMC/SD/SDIO host controller acknowledges the idle request from the PRCM after ensuring the following:

- The current multi/single-block transfer is completed.
- Any interrupt or DMA request is asserted.
- There is no card interrupt on the SD_dat1 signal.

As long as the MMC/SD/SDIO controller does not acknowledge the idle request, if an event occurs, the MMC/SD/SDIO host controller can still generate an interrupt or a DMA request. In this case, the module ignores the idle request from the PRCM.

As soon as the MMC/SD/SDIO controller acknowledges the idle request from the PRCM:

- If Smart-Idle mode the module does not assert any new interrupt or DMA request

17.3.3.4 Transition from Smart-Idle Mode to Normal Mode

The MMC/SD/SDIO host controller detects the end of the idle period when the PRCM deasserts the idle request. For the wake-up event, there is a corresponding interrupt status in the SD_STAT register. The MMC/SD/SDIO host controller operates the conversion between wake-up and interrupt (or DMA request) upon exit from smart-idle mode if the associated enable bit is set in the SD_ISE register.

Interrupts and wake-up events have independent enable/disable controls, accessible through the SD_HCTL and SD_ISE registers. The overall consistency must be ensured by software.

The interrupt status register SD_STAT is updated with the event that caused the wake-up in the CIRQ bit when the SD_IE[8] CIRQ_ENABLE associated bit is enabled. Then, the wake-up event at the origin of the transition from smart-idle mode to normal mode is converted into its corresponding interrupt or DMA request. (The SD_STAT register is updated and the status of the interrupt signal changes.)

When the idle request from the PRCM is deasserted, the module switches back to normal mode. The module is fully operational.

17.3.3.5 Force-Idle Mode

Force-idle mode is enabled when the SD_SYSCONFIG[4:3] SIDLEMODE bit field is cleared to 0. Force-idle mode is an idle mode where the MMC/SD/SDIO host controller responds unconditionally to the idle request from the PRCM. Moreover, in this mode, the MMC/SD/SDIO host controller unconditionally deasserts interrupts and DMA request lines are asserted.

The transition from normal mode to force-idle mode does not affect the bits of the SD_STAT register. In force-idle mode, the interrupt and DMA request lines are deasserted. Interface Clock (OCP) and functional clock (CLKADPI) can be switched off.

CAUTION

In Force-idle mode, an idle request from the PRCM during a command or a data transfer can lead to an unexpected and unpredictable result. When the module is idle, any access to the module generates an error as long as the OCP clock is alive.

The module exits the force-idle mode when the PRCM deasserts the idle request. Then the module switches back to normal mode. The module is fully operational. Interrupt and DMA request lines are optionally asserted one clock cycle later.

17.3.3.6 Local Power Management

Table 17-9 describes power-management features available for the MMC/SD/SDIO modules.

Table 17-9. Local Power Management Features

Feature	Registers	Description
Clock Auto Gating	SD_SYSCONFIG AUTOIDLE bit	This bit allows a local power optimization inside module, by gating the OCP clock upon the interface activity or gating the CLKADPI clock upon the internal activity.
Slave Idle Modes	SD_SYSCONFIG SIDLEMODE bit	Force-idle, No-idle, and Smart-idle modes
Clock Activity	SD_SYSCONFIG CLOCKACTIVITY bit	Please see Table 17-10 for configuration details.
Global Wake-Up Enable	SD_SYSCONFIG ENAWAKEUP bit	This bit enables the wake-up feature at module level.
Wake-Up Sources Enable	SD_HCTL register	This register holds one active high enable bit per event source able to generate wake-up signal.

Table 17-10. Clock Activity Settings

CLOCKACTIVITY Values	Clock State When Module is in IDLE State		Features Available when Module is in IDLE State	Wake-Up Events
	OCP Clock	CLKADPI		
00	OFF	OFF	None	Card Interrupt
10	OFF	ON	None	
01	ON	OFF	None	
11	ON	ON	All	

CAUTION

The PRCM module has no hardware means of reading CLOCKACTIVITY settings. Thus, software must ensure consistent programming between the CLOCKACTIVITY and MMC clock PRCM control bits.

17.3.4 Interrupt Requests

Several internal module events can generate an interrupt. Each interrupt has a status bit, an interrupt enable bit, and a signal status enable:

- The status of each type of interrupt is automatically updated in the SD_STAT register; it indicates which service is required.
- The interrupt status enable bits of the SD_IE register enable/disable the automatic update of the SD_STAT register on an event-by-event basis.
- The interrupt signal enable bits of the SD_ISE register enable/disable the transmission of an interrupt request on the interrupt line MMC_IRQ (from the MMC/SD/SDIO host controller to the MPU subsystem interrupt controller) on an event-by-event basis.

If an interrupt status is disabled in the SD_IE register, then the corresponding interrupt request is not transmitted, and the value of the corresponding interrupt signal enable in the SD_ISE register is ignored.

When an interrupt event occurs, the corresponding status bit is automatically set to 1 (the MMC/SD/SDIO host controller updates the status bit) in the SD_STAT register. If later a mask is applied on the interrupt in the SD_ISE register, the interrupt request is deactivated.

When the interrupt source has not been serviced, if the interrupt status is cleared in the SD_STAT register and the corresponding mask is removed from the SD_ISE register, the interrupt status is not asserted again in the SD_STAT register and the MMC/SD/SDIOi host controller does not transmit an interrupt request.

CAUTION

If the buffer write ready interrupt (BWR) or the buffer read ready only interrupt (BRR) are not serviced and are cleared in the SD_STAT register, and the corresponding mask is removed, then the MMC/SD/SDIOi host controller will wait for the service of the interrupt without updating the status SD_STAT or transmitting an interrupt request.

Table 17-11 lists the event flags, and their mask, that can cause module interrupts.

Table 17-11. Events

Event Flag	Event Mask	Map To	Description
SD_STAT[29] BADA	SD_IE[29] BADA_ENABLE	MMC_IRQ	Bad Access to Data space. This bit is set automatically to indicate a bad access to buffer when not allowed. This bit is set during a read access to the data register (SD_DATA) while buffer reads are not allowed (SD_PSTATE[11] BRE=0). This bit is set during a write access to the data register (SD_DATA) while buffer writes are not allowed (SD_STATE[10] BWE=0)
SD_STAT[28] CERR	SD_IE[28] CERR_ENABLE	MMC_IRQ	Card Error. This bit is set automatically when there is at least one error in a response of type R1, R1b, R6, R5 or R5b. Only bits referenced as type E(error) in status field in the response can set a card status error. An error bit in the response is flagged only if corresponding bit in card status response errors SD_CSRE is set. There is not card detection for auto CMD12 command.
SD_STAT[25] ADMAE	SD_IE[25] ADMAE_ENABLE	MMC_IRQ	ADMA error. This bit is set when the host controller detects errors during ADMA based data transfer. The stat of the ADMA at an error occurrence is saved in the ADMA Error Status Register. In addition, the host controller generates this interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state.
SD_STAT[24] ACE	SD_IE[24] ACE_ENABLE	MMC_IRQ	Auto CMD12 error. This bit is set automatically when one of the bits in Auto CMD12 Error status register has changed from 0 to 1
SD_STAT[22] DEB	SD_IE[22] DEB_ENABLE	MMC_IRQ	Data End Bit error. This bit is set automatically when detecting a 0 at the end bit position of read data on DAT line or at the end position of the CRC status in write mode.

Table 17-11. Events (continued)

Event Flag	Event Mask	Map To	Description
SD_STAT[21] DCRC	SD_IE[21] DCRC_ENABLE	MMC_IRQ	Data CRC error. This bit is set automatically when there is a CRC16 error in the data phase response following a block read command or if there is a 3-bit CRC status different of a position "010" token during a block write command.
SD_STAT[20] DTO	SD_IE[20] DTO_ENABLE	MMC_IRQ	Data Timeout error. This bit is set automatically according to the following conditions: A) busy timeout for R1b, R5b response. B) busy timeout after write CRC status. C) write CRC status timeout, or D) read data timeout.
SD_STAT[19] CIE	SD_IE[19] CIE_ENABLE	MMC_IRQ	Command Index Error. This bit is set automatically when response index differs from corresponding command index previously emitted. The check is enabled through SD_CMD[20] CICE bit.
SD_STAT[18] CEB	SD_IE[18] CEB_ENABLE	MMC_IRQ	Command End Bit error. This bit is set automatically when detecting a 0 at the end bit position of a command response.
SD_STAT[17] CCRC	SD_IE[17] CCRC_ENABLE	MMC_IRQ	Command CRC error. This bit is set automatically when there is a CRC7 error in the command response. CRC check is enabled through the SD_CMD[19] CCCE bit.
SD_STAT[16] CTO	SD_IE[16] CTO_ENABLE	MMC_IRQ	Command Timeout error. This bit is set automatically when no response is received within 64 clock cycles from the end bit of the command. For commands the reply within 5 clock cycles, the timeout is still detected at 64 clock cycles.
SD_STAT[15] ERRI	SD_IE[15] ERRI_ENABLE	MMC_IRQ	Error Interrupt. If any of the bits in the Error Interrupt Status register (SD_STAT[24:15]) are set, the this bit is set to 1.
SD_STAT[10] BSR	SD_IE[10] BSR_ENABLE	MMC_IRQ	Boot Status Received interrupt. This bit is set automatically when SD_CON[18] BOOT_CF0 is set to 1 or 2h and boot status is received on the dat0 line. This interrupt is only used for MMC cards.
SD_STAT[8] CIRQ	SD_IE[8] CIRQ_ENABLE	MMC_IRQ	Card Interrupt. This bit is only used for SD, SDIO, and CE-ATA cards. In 1-bit mode, interrupt source is asynchronous (can be a source of asynchronous wake-up). In 4-bit mode, interrupt source is sampled during the interrupt cycle. In CE-ATA mode, interrupt source is detected when the card drive CMD line to zero during one cycle after data transmission end.
SD_STAT[5] BRR	SD_IE[5] BRR_ENABLE	MMC_IRQ	Buffer Read ready. This bit is set automatically during a read operation to the card when one block specified by SD_BLK[10:0] BLEN is completely written in the buffer. It indicates that the memory card has filled out the buffer and the local host needs to empty the buffer by reading it.
SD_STAT[4] BWR	SD_IE[4] BWR_ENABLE	MMC_IRQ	Buffer Write ready. This bit is automatically set during a write operation to the card when the host can write a complete block as specified by SD_BLK[10:0] BLEN. It indicates that the memory card has emptied one block from the bugger and the local host is able to write one block of data into the buffer.
SD_STAT[3] DMA	SD_IE[3] DMA_ENABLE	MMC_IRQ	DMA interrupt. This status is set when an interrupt is required in the ADMA instruction and after the data transfer is complete.
SD_STAT[2] BGE	SD_IE[2] BGE_ENABLE	MMC_IRQ	Block Gap event. When a stop at block gap is requested (SD_HCTL[16] SBGR), this bit is automatically set when transaction is stopped at the block gap during a read or write operation.
SD_STAT[1] TC	SD_IE[1] TC_ENABLE	MMC_IRQ	Transfer completed. This bit is always set when a read/write transfer is completed or between two blocks when the transfer is stopped due to a stop at block gap requested (SD_HCTL[16 SBGR). In read mode this bit is automatically set on completion of a read transfer (SD_PSTATE[9] RTA). In write mode, this bit is automatically set on completion of the DAT line use (SD_PSTATE[2] DLA).
SD_STAT[0] CC	SD_IE[0] CC_ENABLE	MMC_IRQ	Command complete. This bit is set when a 1-to-0 transition occurs in the register command inhibit (SD_PSTATE[0] CMDI). If the command is a type for which no response is expected, then the command complete interrupt is generated at the end of the command. A command timeout error (SD_STAT[16] CTO) has higher priority than command complete (SD_STAT[0] CC). If a response is expected but none is received, the a Command Timeout error is detected and signaled instead of the Command Complete interrupt.

17.3.4.1 Interrupt-Driven Operation

An interrupt enable bit must be set in the SD_IE register to enable the module internal source of interrupt.

When an interrupt event occurs, the single interrupt line is asserted and the LH must:

- Read the SD_STAT register to identify which event occurred.
- Write 1 into the corresponding bit of the SD_STAT register to clear the interrupt status and release the interrupt line (if a read is done after this write, this would return 0).

NOTE: In the SD_STAT register, Card Interrupt (CIRQ) and Error Interrupt (ERRI) bits cannot be cleared.

The SD_STAT[8] CIRQ status bit must be masked by disabling the SD_IE[8] CIRQ_ENABLE bit (cleared to 0), then the interrupt routine must clear SDIO interrupt source in SDIO card common control register (CCCR).

The SD_STAT[15] ERRI bit is automatically cleared when all status bits in SD_STAT[31:16] are cleared.

17.3.4.2 Polling

When the interrupt capability of an event is disabled in the SD_ISE register, the interrupt line is not asserted:

- Software can poll the status bit in the SD_STAT register to detect when the corresponding event occurs.
- Writing 1 into the corresponding bit of the SD_STAT register clears the interrupt status and does not affect the interrupt line state.

NOTE: Please see the note in Section 17.3.4.1 concerning CIRQ and ERRI bits clearing.

17.3.5 DMA Modes

The device supports DMA slave mode only. In this case, the controller is slave on DMA transaction managed by two separated requests (SDMAWREQN and SDMARREQN)

17.3.5.1 DMA Slave Mode Operations

The MMC/SD/SDIO controller can be interfaced with a DMA controller. At system level, the advantage is to discharge the local host (LH) of the data transfers. The module does not support wide DMA access (above 1024 bytes) for SD cards as specified in the *SD Card Specification*and *SD Host Controller Standard Specification*.

The DMA request is issued if the following conditions are met:

- The SD_CMD[0] DE bit is set to 1 to trigger the initial DMA request (the write must be done when running the data transfer command).
- A command was emitted on the SD_cmd line.
- There is enough space in the buffer of the MMC/SD/SDIO controller to write an entire block (BLEN writes).

17.3.5.1.1 DMA Receive Mode

In a DMA block read operation (single or multiple), the request signal SDMARREQN is asserted to its active level when a complete block is written in the buffer. The block size transfer is specified in the SD_BLK[10:0] BLEN field.

The SDMARREQN signal is deasserted to its inactive level when the sDMA has read one single word from the buffer. Only one request is sent per block; the DMA controller can make a 1-shot read access or several DMA bursts, in which case the DMA controller must manage the number of burst accesses, according to block size BLEN field.

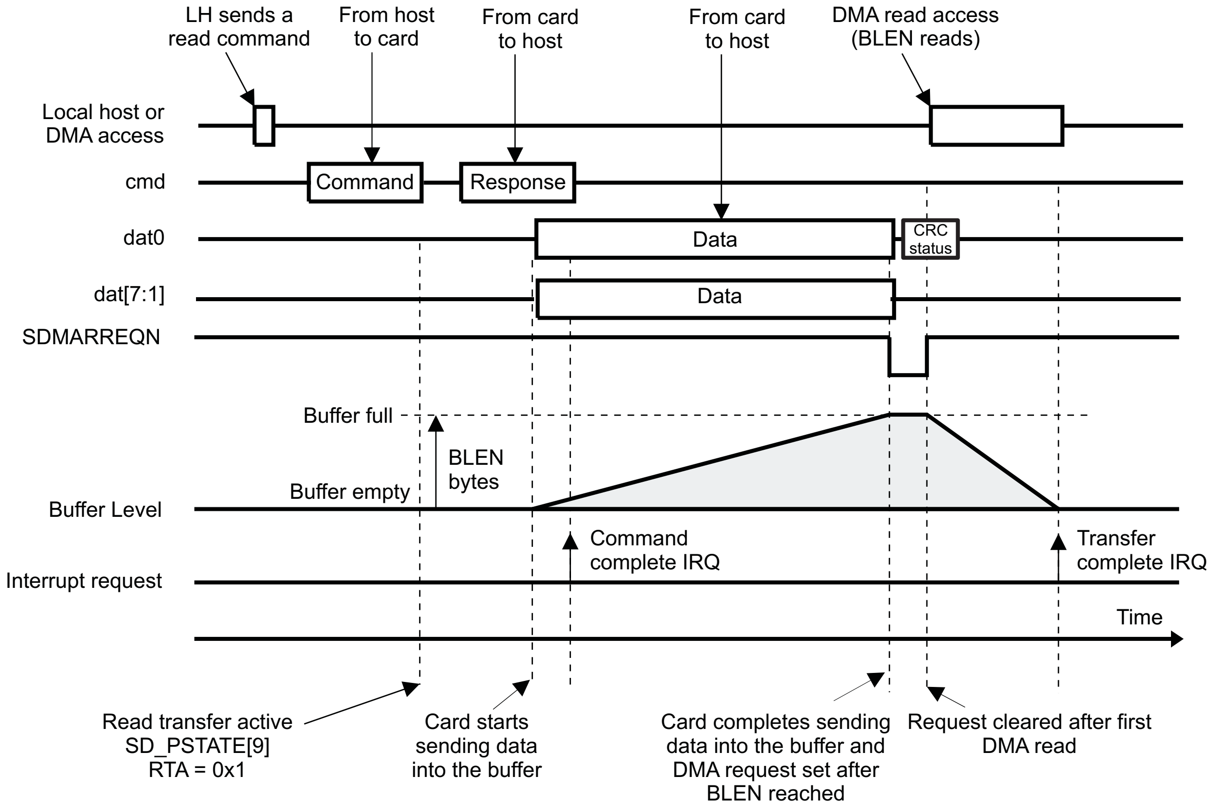
New DMA requests are internally masked if the sDMA has not read exactly BLEN bytes and a new complete block is not ready. As DMA accesses are in 32-bit, then the number of sDMA read is Integer(BLEN/4)+1.

The receive buffer never overflows. In multiple block transfers for block size above 512 bytes, when the buffer gets full, the MMC_CLK clock signal (provided to the card) is momentarily stopped until the sDMA or the MPU performs a read access, which reads a complete block in the buffer.

Figure 17-17 provides a summary:

- DMA transfer size = BLEN buffer size in one shot or by burst
- One DMA request per block

Figure 17-17. DMA Receive Mode



17.3.5.1.2 DMA Transmit Mode

In a DMA block write operation (single or multiple), the request signal SDMAWREQN is asserted to its active level when a complete block is to be written to the buffer. The block size transfer is specified in the SD_BLK[10:0] BLEN field.

The SDMAWREQN signal is deasserted to its inactive level when the sDMA has written one single word to the buffer.

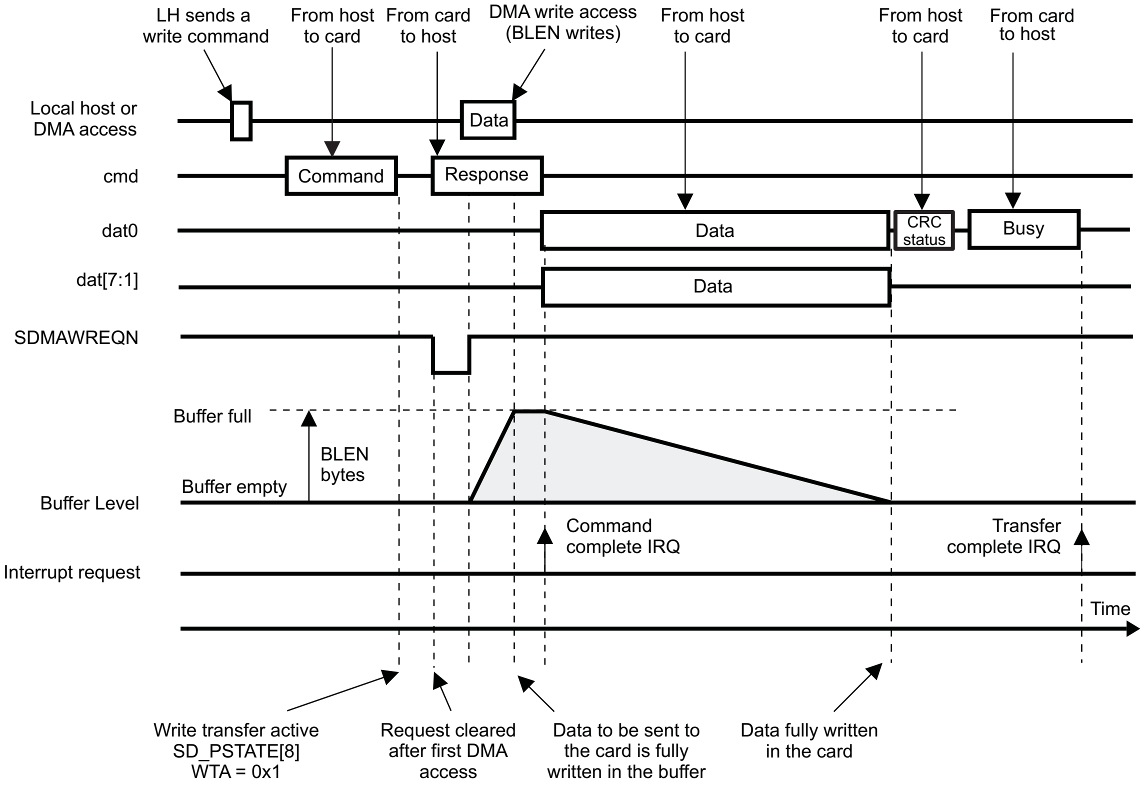
Only one request is sent per block; the DMA controller can make a 1-shot write access or multiple write DMA bursts, in which case the DMA controller must manage the number of burst accesses, according to block size BLEN field.

New DMA requests are internally masked if the sDMA has not written exactly BLEN bytes (as DMA accesses are in 32-bit, then the number of sDMA read is Integer(BLEN/4)+1) and if there is not enough memory space to write a complete block in the buffer.

Figure 17-18 provides a summary:

- DMA transfer size = BLEN buffer size in one shot or by burst
- One DMA request per block

Figure 17-18. DMA Transmit Mode



17.3.6 Mode Selection

The MMC/SD/SDIO host controller can be use in two modes, MMC and SD/SDIO modes. It has been designed to be the most transparent with the type of card. The type of the card connected is differentiated by the software initialization procedure.

Software identifies the type of card connected during software initialization. For each given card type, there are corresponding commands. Some commands are not supported by all cards. See the *Multimedia Card System Specification*, the *SD Memory Card Specifications*, and the *SDIO Card Specification, Part E1* for more details.

The purpose of the module is to transfer commands and data, to whatever card is connected, respecting the protocol of the connected card. Writes and reads to the card must respect the appropriate protocol of that card.

17.3.7 Buffer Management

17.3.7.1 Data Buffer

The MMC/SD/SDIO host controller uses a data buffer. This buffer transfers data from one data bus (Interconnect) to another data bus (SD, SDIO, or MMC card bus) and vice versa.

The buffer is the heart of the interface and ensures the transfer between the two interfaces (L4 and the card). To enhance performance, the data buffer is completed by a prefetch register and a post-write buffer that are not accessible by the host controller.

The read access time of the prefetch register is faster than the one of the data buffer. The prefetch register allows data to be read from the data buffer at an increased speed by preloading data into the prefetch register.

The entry point of the data buffer, the prefetch buffer, and the post-write buffer is the 32-bit register SD_DATA. A write access to the SD_DATA register followed by a read access from the SD_DATA register corresponds to a write access to the post-write buffer followed by a read access to the prefetch buffer. As a consequence, it is normal that the data of the write access to the SD_DATA register and the data of the read access to the SD_DATA register are different.

The number of 32-bit accesses to the SD_DATA register that are needed to read (or write) a data block with a size of SD_BLK[10:0] BLEN, and equals the rounded up result of BLEN divided by 4. The maximum block size supported by the host controller is hard-coded in the register SD_CAPA[17:16] MBL field and cannot be changed.

A read access to the SD_DATA register is allowed only when the buffer read enable status is set to 1 (SD_PSTATE[11] BRE); otherwise, a bad access (SD_STAT[29] BADA) is signaled.

A write access to the SD_DATA register is allowed only when the buffer write enable status is set to 1 (SD_PSTATE[10] BWE); otherwise, a bad access (SD_STAT[29] BADA) is signaled and the data is not written.

The data buffer has two modes of operation to store and read of the first and second portions of the data buffer:

- When the size of the data block to transfer is less than or equal to MEM_SIZE/2 (in double buffering), two data transfers can occur from one data bus to the other data bus and vice versa at the same time. The MMC/SD/SDIO controller uses the two portions of the data buffer in a ping-pong manner so that storing and reading of the first and second portions of the data buffer are automatically interchanged from time to time so that data may be read from one portion (for instance, through a DMA read access on the interconnect bus) while data (for instance, from the card) is being stored into the other portion and vice versa. When BLEN is less than or equal to 200h (that is, less or equal to 512Bytes), each of the two portions of the buffer that can be used have a size of BLEN (that is, 32-bits x BLEN div by 4). Not more than this total size of 2 times 32-bits × BLEN div by 4 can be used.
- When the size of the data block to transfer is larger than MEM_SIZE/2, only one data transfer can occur from one data bus to the other data bus at a time. The MMC/SD/SDIO host controller uses the entire data buffer as a single portion. In this mode, a bad access (SD_STAT[29] BADA) is signaled when two data transfers occur from one data bus to the other data bus and vice versa at the same time.

CAUTION

The SD_CMD[4] DDIR bit must be configured before a transfer to indicate the direction of the transfer.

Figure 17-19 shows the buffer management for writing and Figure 17-20 shows the buffer management for reading.

Figure 17-19. Buffer Management for a Write

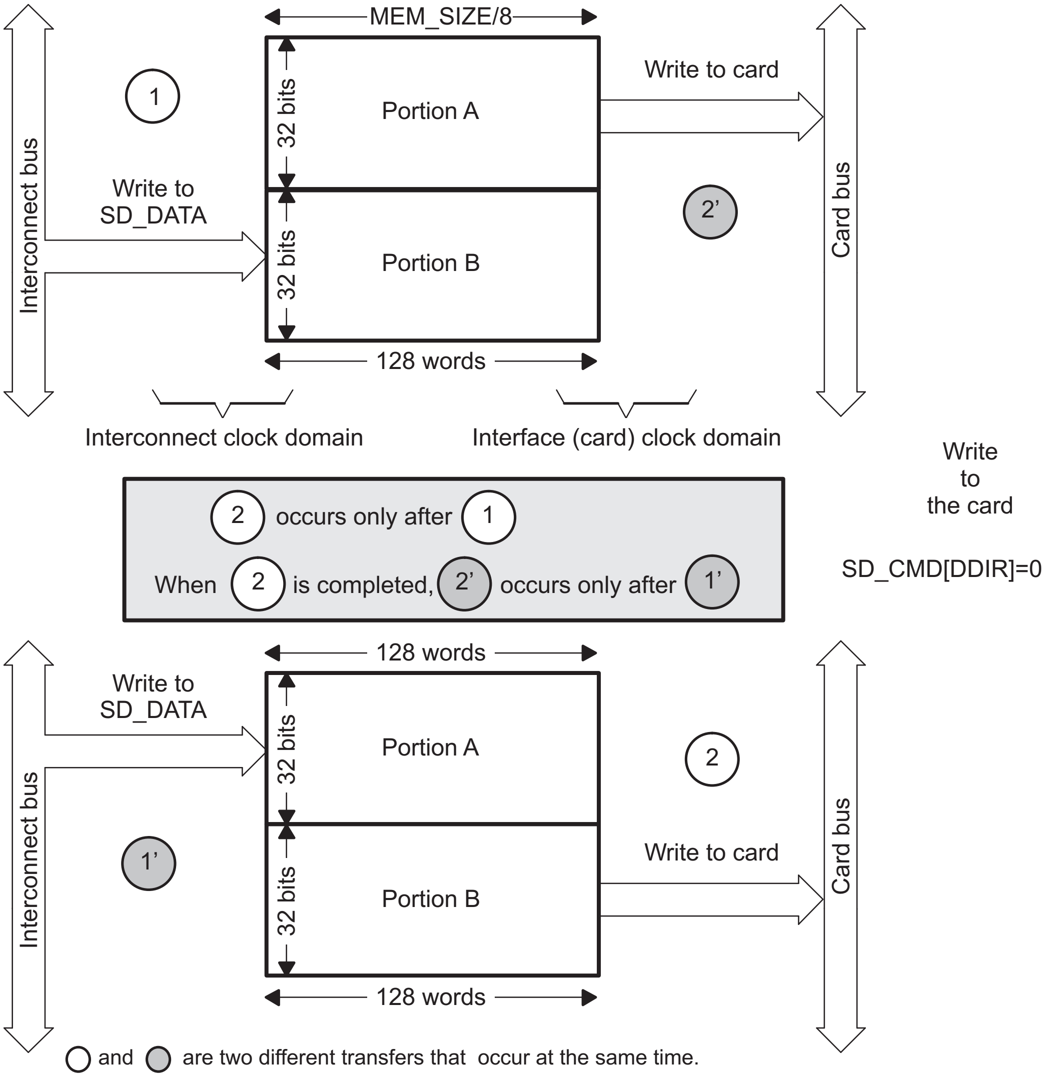
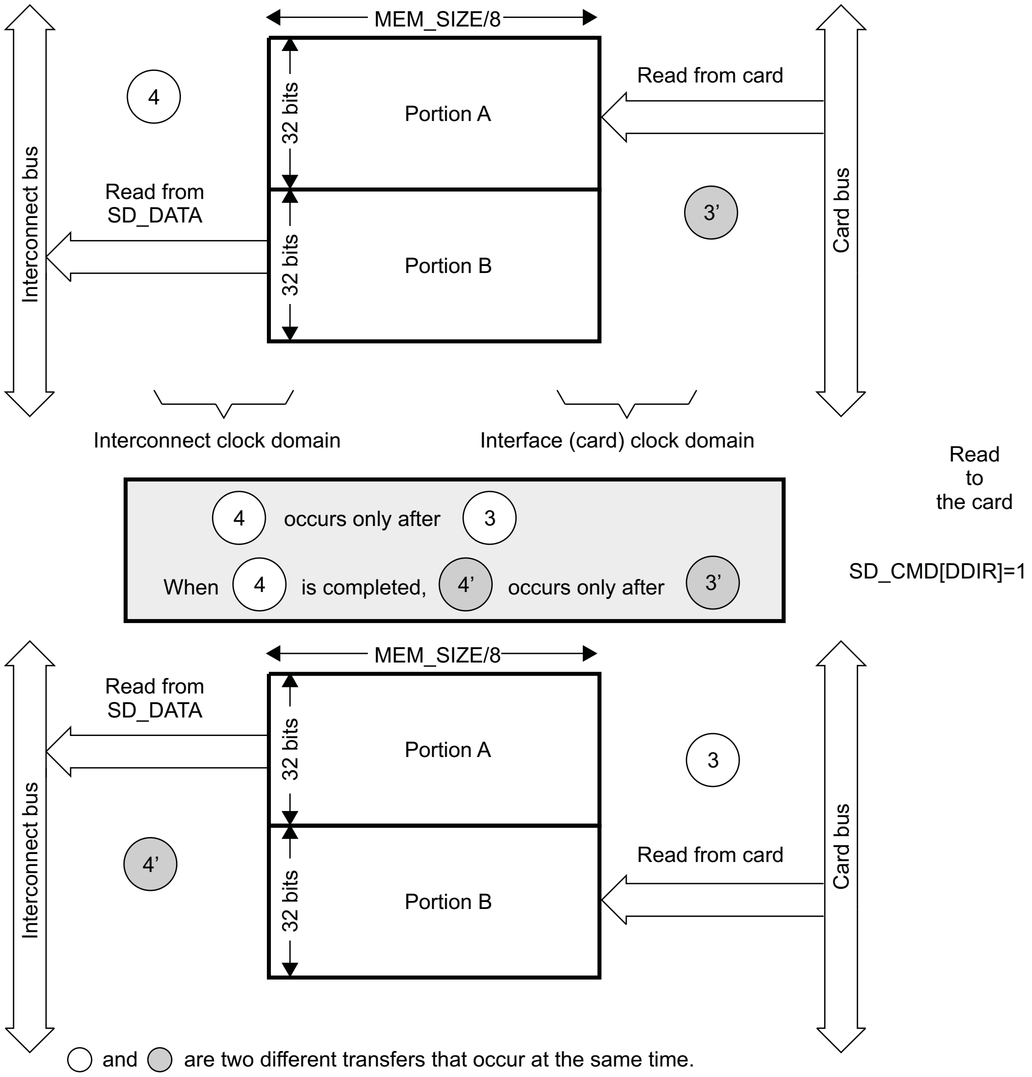


Figure 17-20. Buffer Management for a Read



17.3.7.1.1 Memory Size, Block Length, and Buffer Management Relationship

The maximum block length and buffer management that can be targeted by system depend on memory depth setting.

Table 17-12. Memory Size, BLEN, and Buffer Relationship

Memory Size([5:2] MEMSIZE in bytes)	512	1024	2048	4096
Maximum block length supported	512	1024	2048	2048
Double-buffering for maximum block length	N/A	BLEN <= 512	BLEN <= 1024	BLEN <= 2048
Single-buffering for block length	BLEN<=512	512 < BLEN <= 1024	1024 < BLEN <= 2048	N/A

17.3.7.1.2 Data Buffer Status

The data buffer status is defined in the following interrupt status register and status register:

- Interrupt status registers (see):
  - SD_STAT[29] BADA Bad access to data space
  - SD_STAT[5] BRR Buffer read ready
  - SD_STAT[4] BWR Buffer write ready
- Status registers (see):
  - SD_PSTATE[11] BRE Buffer read enable
  - SD_PSTATE[10] BWE Buffer write enable

17.3.8 Transfer Process

The process of a transfer is dependent on the type of command. It can be with or without a response, with or without data.

17.3.8.1 Different Types of Commands

Different types of commands are specific to MMC, SD, or SDIO cards. See the *Multimedia Card System Specification*, the *SD Memory Card Specifications*, the *SDIO Card Specification, Part E1*, or the *SD Card Specification, Part A2, SD Host Controller Standard Specification* for more details.

17.3.8.2 Different Types of Responses

Different types of responses are specific to MMC, SD, or SDIO cards. See the *Multimedia Card System Specification*, the *SD Memory Card Specifications*, the *SDIO Card Specification, Part E1*, or the *SD Card Specification, Part A2, SD Host Controller Standard Specification* for more details.

Table 17-13 shows how the MMC, SD, and SDIO responses are stored in the SD_RSPxx registers.

Table 17-13. MMC, SD, SDIO Responses in the SD_RSPxx Registers

Kind of Response	Response Field	Response Register
R1, R1b (normal response), R3, R4, R5, R5b, R6, R7	RESP[39:8][1]	SD_RSP10[31:0]
R1b (Auto CMD12 response)	RESP[39:8][1]	SD_RSP76[31:0]
R2	RESP[127:0][1]	SD_RSP76[31:0] SD_RSP54[31:0] SD_RSP32[31:0] SD_RSP10[31:0]

[1] RESP refers to the command response format described in the specifications mentioned above.

When the host controller modifies part of the SD_RSPxx registers, it preserves the unmodified bits.

The host controller stores the Auto CMD12 response in the SD_RSP76[31:0] register because the Host Controller may have a multiple block data DAT line transfer executing concurrently with a command. This allows the host controller to avoid overwriting the Auto CMD12 response with the command response stored in SD_RSP10 register and vice versa.

17.3.9 Transfer or Command Status and Error Reporting

Flags in the MMC/SD/SDIO host controller show status of communication with the card:

- A timeout (of a command, a data, or a response)
- A CRC

Error conditions generate interrupts. See Table 17-14 and register description for more details.

Table 17-14. CC and TC Values Upon Error Detected

Error hold in the SD_STAT Register		CC	TC	Comments
29	BADA			No dependency with CC or TC. BADA is related to the register accesses. Its assertion is not dependent of the ongoing transfer.
28	CERR	1		CC is set upon CERR.
22	DEB		1	TC is set upon DEB.
21	DCRC		1	TC is set upon DCRC.
20	DTO			DTO and TC are mutually exclusive. DCRC and DEB cannot occur with DTO.
19	CIE	1		CC is set upon CIE.
18	CEB	1		CC is set upon CEB.
17	CCRC	1		CC can be set upon CCRC - See CTO comment
16	CTO			CTO and CC are mutually exclusive. CIE, CEB and CERR cannot occur with CTO. CTO can occur at the same time as CCRC it indicates a command abort due to a contention on CMD line. In this case no CC appears.

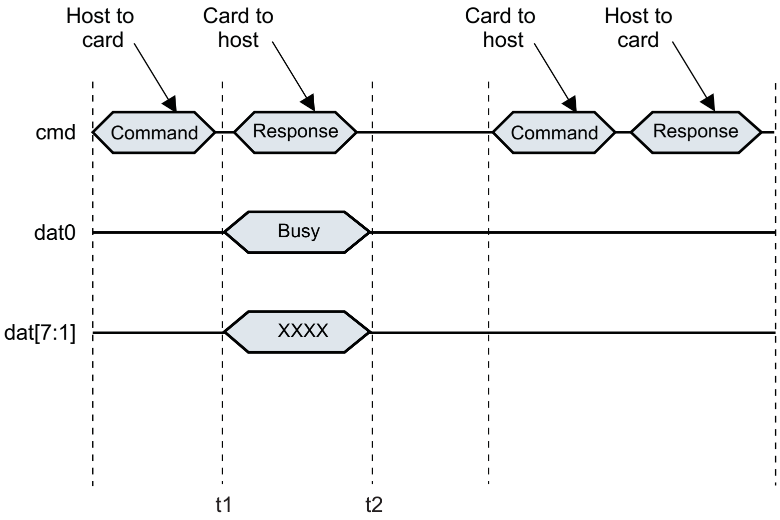
SD_STAT[21] DCRC event can be asserted in the following conditions:

- Busy timeout for R1b, R5b response type
- Busy timeout after write CRC status
- Write CRC status timeout
- Read data timeout
- Boot acknowledge timeout

17.3.9.1 Busy Timeout for R1b, R5b Response Type

Figure 17-21 shows DCRD event condition asserted when there is a busy timeout for R1b or R5b responses.

Figure 17-21. Busy Timeout for R1b, R5b Responses



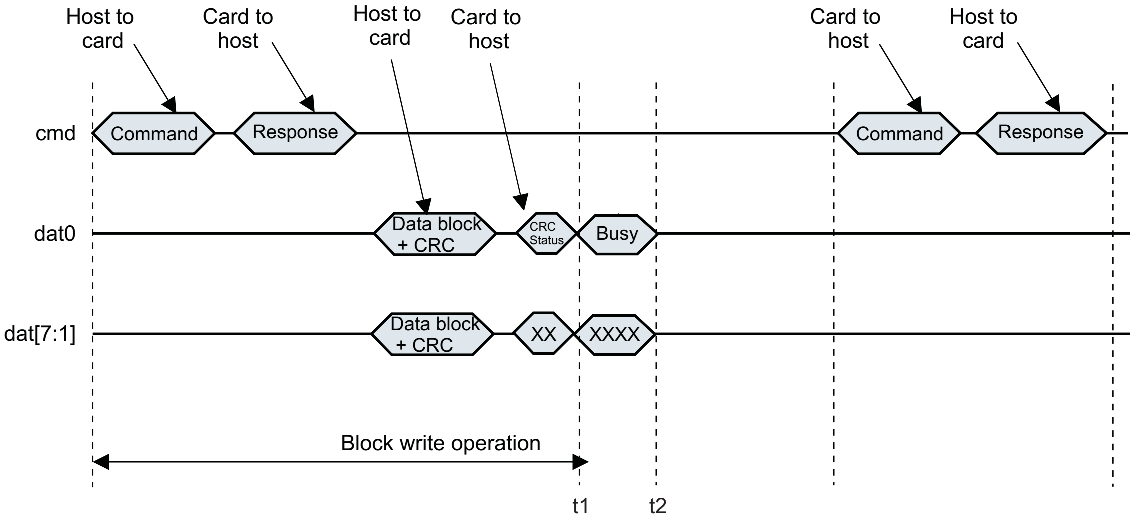
t1 - Data timeout counter is loaded and starts after R1b, R5b response type.

t2 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

17.3.9.2 Busy Timeout After Write CRC Status

Figure 17-22 shows DCRC event condition asserted when there is busy timeout after write CRC status.

Figure 17-22. Busy Timeout After Write CRC Status



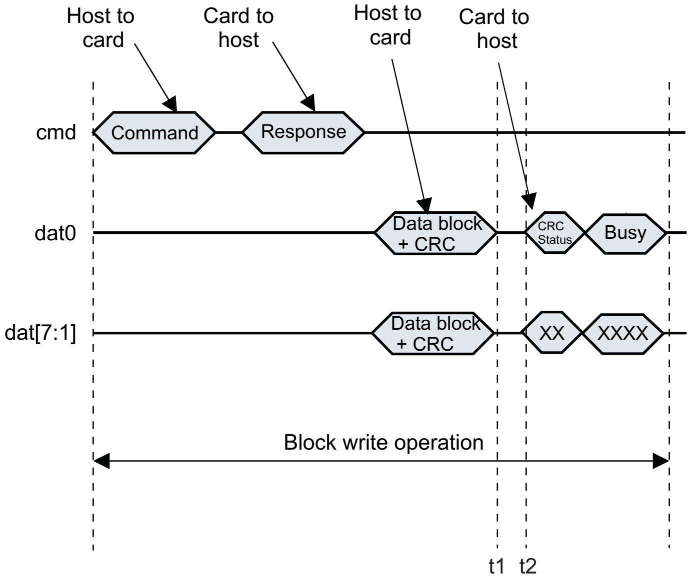
t1 - Data timeout counter is loaded and starts after CRC status.

t2 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

17.3.9.3 Write CRC Status Timeout

Figure 17-23 shows DCRC event condition asserted when there is write CRC status timeout.

Figure 17-23. Write CRC Status Timeout



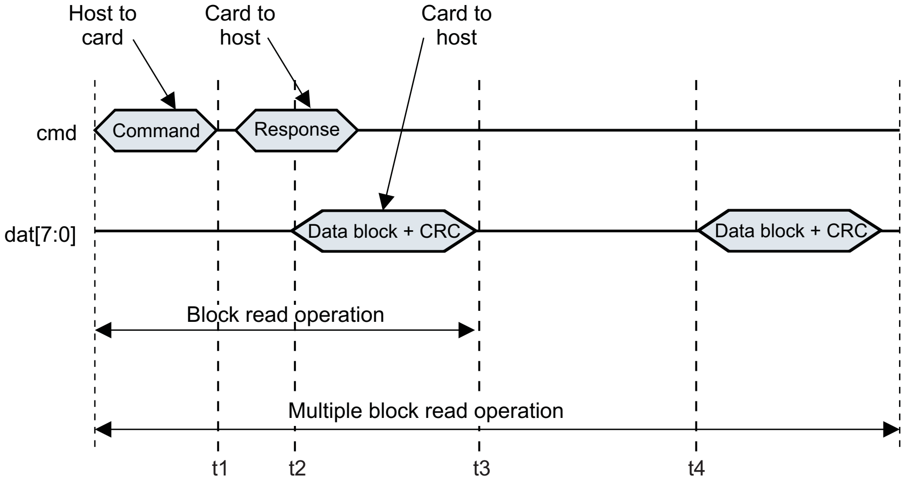
t1 - Data timeout counter is loaded and starts after Data block + CRC.

t2 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

17.3.9.4 Read Data Timeout

Figure 17-24 shows DCRC event condition asserted when there is read data timeout.

Figure 17-24. Read Data Timeout



t1 - Data timeout counter is loaded and starts after Command transmission.

t2 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

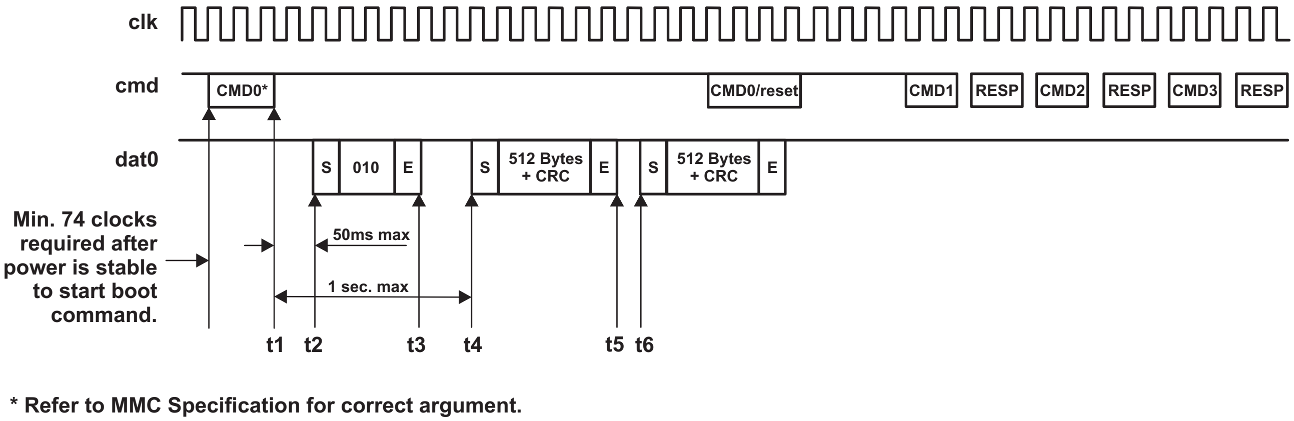
t3 - Data timeout counter is loaded and starts after Data block + CRC transmission.

t4 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

17.3.9.5 Boot Acknowledge Timeout

Figure 17-25 shows DCRC event condition asserted when there is boot acknowledge timeout and CMD0 is used.

Figure 17-25. Boot Acknowledge Timeout When Using CMD0



t1 - Data timeout counter is loaded and starts after CMD0.

t2 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

t3 - Data timeout counter is loaded and starts.

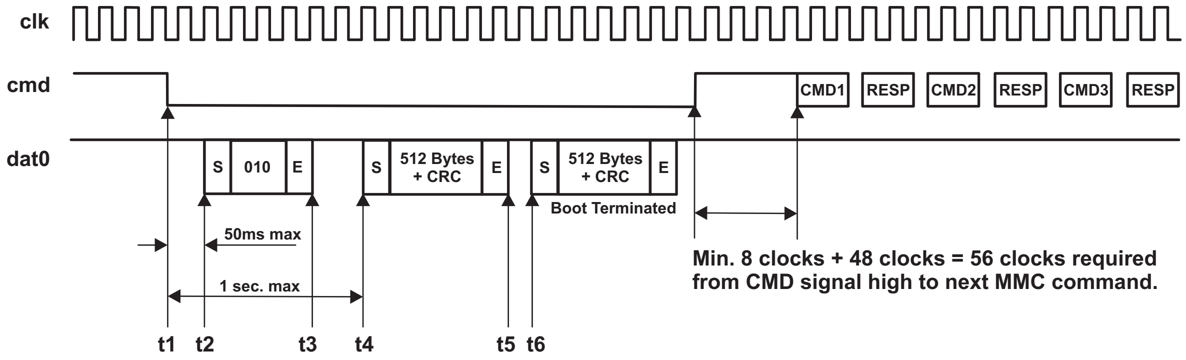
t4 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

t5 - Data timeout counter is loaded and starts after Data + CRC transmission.

t6 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

Figure 17-26 shows DCRC event condition asserted when there is boot acknowledge timeout and CMD line is held low.

Figure 17-26. Boot Acknowledge Timeout When CMD Held Low



t1 - Data timeout counter is loaded and starts after cmd line is tied to 0.

t2 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

t3 - Data timeout counter is loaded and starts.

t4 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

t5 - Data timeout counter is loaded and starts after Data + CRC transmission.

t6 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

17.3.10 Auto Command 12 Timings

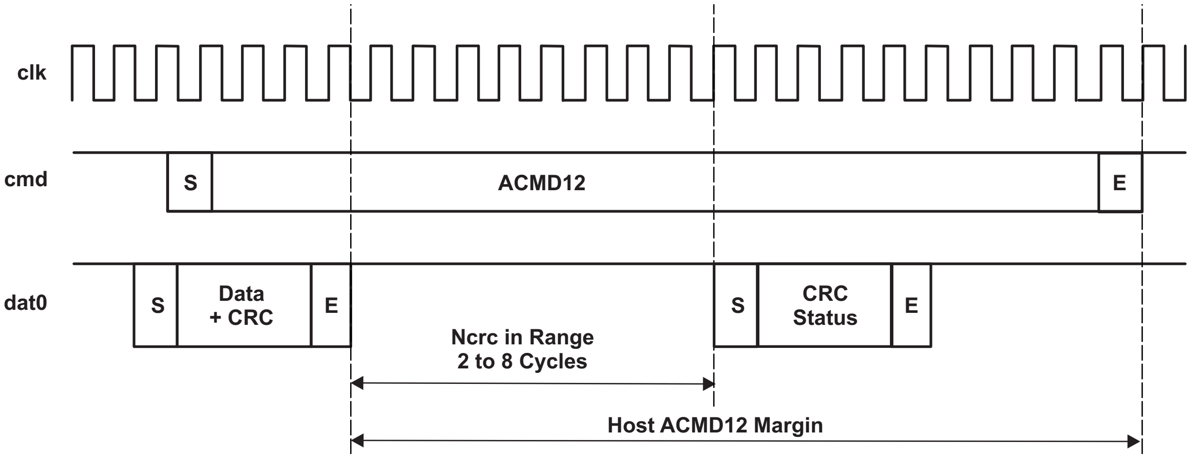
With the UHS definition of SD cards with higher frequency for MMC clocks up to 208, SD standard imposes a specific timing for Auto CMD12 "end bit" arrival.

17.3.10.1 Auto Command 12 Timings During Write Transfer

A margin named Ncrc in range of 2 to 8 cycles has been defined for SDR50 and SDR104 card components for write data transfers, as auto command 12 'end bit' shall arrive after the CRC status "end bit".

Figure 17-27 shows auto CMD12 timings during write transfer.

Figure 17-27. Auto CMD12 Timing During Write Transfer



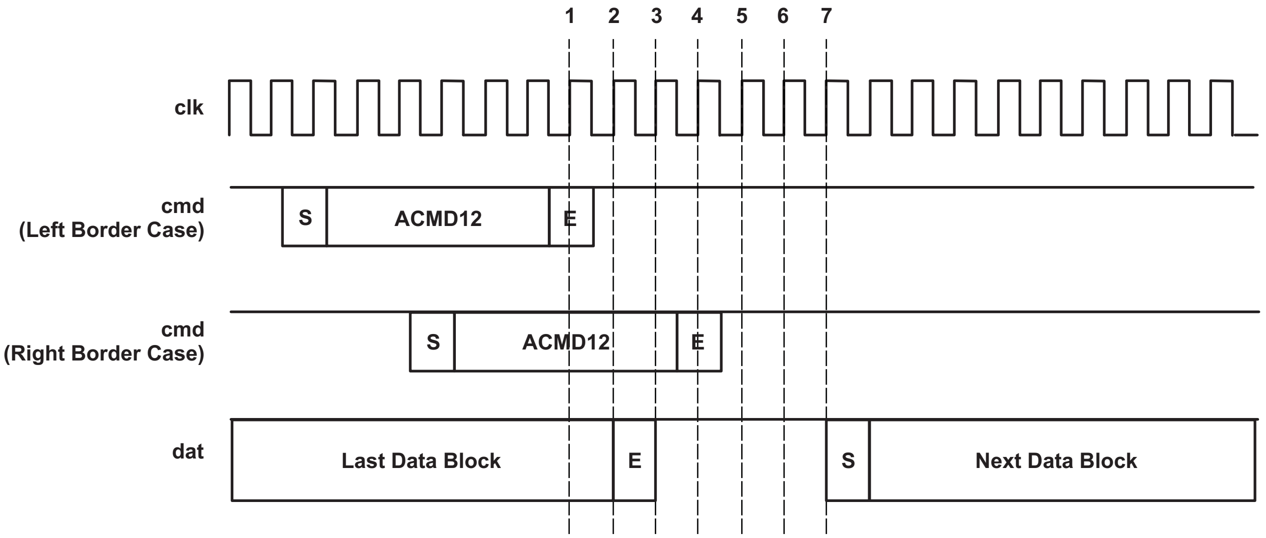
The Host controller has a margin of 18 clock cycles to make sure that auto CMD12 'end bit' arrives after the CRC status. This margin does not depend on MMC bus configuration, DDR or standard transfer, 1,4 or 8 bus width.

17.3.10.2 Auto Command 12 Timings During Read Transfer

With UHS very high speed cards gap timing between 2 successive cards has been extended to 4 cycles instead of 2. By the way it gives more flexibility for Host Auto CMD12 arrival in order to receive the last complete and reliable block. SD controller only follows the 'Left Border Case' defined by SD UHS specification.

Figure 17-28 shows ACMD12 timings during read transfer.

Figure 17-28. Auto Command 12 Timings During Read Transfer



The Auto CMD12 arrival sent by the Host controller is not sensitive to the MMC bus configuration whether it is DDR or standard transfer and whether it is a 1,4 or 8 bit bus width transfer.

17.3.11 Transfer Stop

Whenever a transfer is initiated, the transmission may be willed to stop whereas it is still not finished. Several cases can be faced depending on the transfer type:

- Multiple blocks oriented transfers (for which transfer length is known)
- Continuous stream transfers (which have an infinite length)

NOTE: Since the MMC/SD/SDIO controller manages transfers based on a block granularity, the buffer will accept a block only if there is enough space to completely store it. Consequently, if a block is pending in the buffer, no command will be sent to the card because the card clock will be shut off by the controller.

The MMC/SD/SDIO controller includes two features which make a transfer stop more convenient and easier to manage:

- Auto CMD12 (for MMC and SD only).

  This feature is enabled by setting the SD_CMD[2] ACEN bit to 1 (this setting is relevant for a MMC/SD transfer with a known number of blocks to transfer). When the Auto CMD12 feature is enabled, the MMC/SD/SDIO controller will automatically issue a CMD12 command when the expected number of blocks has been exchanged.
- Stop at block gap

  This feature is enabled by setting the SD_HCTL[16] SBGR bit to 1. When enabled, this capability holds the transfer on until the end of a block boundary. If a stop transmission is needed, software can use this pause to send a CMD12 to the card.

Table 17-15 shows the common ways to stop a transfer, indicating command to send and features to enable.

Table 17-15. MMC/SD/SDIO Controller Transfer Stop Command Summary

		WRITE Transfer		**READ Transfer**	
		MMC/SD	SDIO	MMC/SD	SDIO
Single block		Transfer ends automatically Wait TC	Transfer ends automatically Wait TC	Transfer ends automatically Wait TC	Transfer ends automatically Wait TC
Multi blocks (finite or infinite)	Before the programmed block boundary	Send CMD12 Wait TC	Send CMD52 Wait TC	Send CMD12 Wait TC	Send CMD52 Wait TC
	Stop at the end of the transfer (finite transfer only)	Auto CMD12 active Transfer ends automatically Wait TC	Set SD_HCTL[16] SBGR bit to 1. Send CMD52 Wait TC	Auto CMD12 active Transfer ends automatically Wait TC	**If READ_WAIT supported** Stop at block gap Wait TC
					If READ_WAIT not supported Send CMD52 Wait TC

NOTE: The MMC/SD/SDIO controller will send the stop command to the card on a block boundary, regardless the moment the command was written to the controller registers.

17.3.12 Output Signals Generation

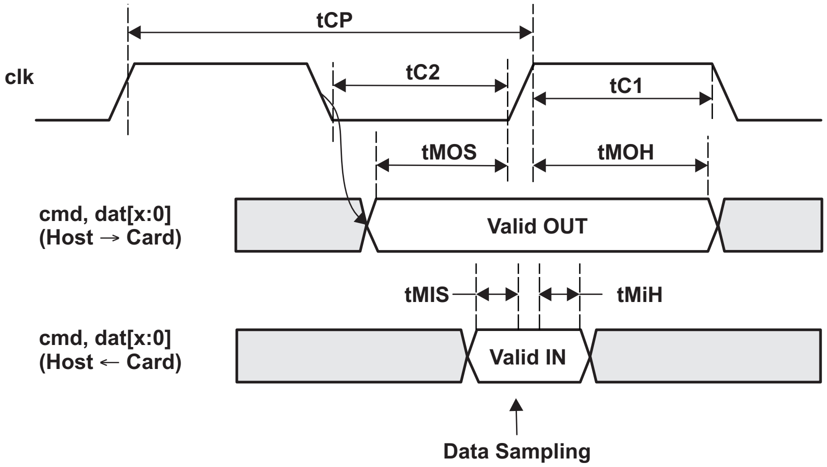
The MMC/SD/SDIO output signals can be driven on either falling edge or rising edge depending on the SD_HCTL[2] HSPE bit. This feature allows to reach better timing performance, and thus to increase data transfer frequency.

17.3.12.1 Generation on Falling Edge of MMC Clock

The controller is by default in this mode to maximize hold timings. In this case, SD_HCTL[2] HSPE bit is cleared to 0.

Figure 17-29 shows the output signals of the module when generating from the falling edge of the MMC clock.

Figure 17-29. Output Driven on Falling Edge



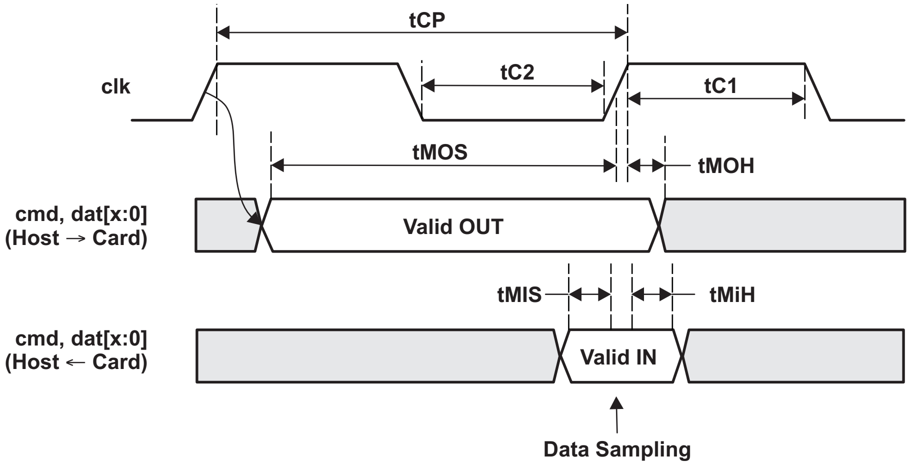
17.3.12.2 Generation on Rising Edge of MMC Clock

This mode increases setup timings and allows reaching higher bus frequency. This feature is activated by setting SD_HCTL[2] HSPE bit to 1. The controller shall be set in this mode to support SDR transfers.

NOTE: Do not use this feature in Dual Data Rate mode (when SD_CON[19] DDR is set to 1).

Figure 17-30 shows the output signals of the module when generating from the rising edge of the MMC clock.

Figure 17-30. Output Driven on Rising Edge



17.3.13 Card Boot Mode Management

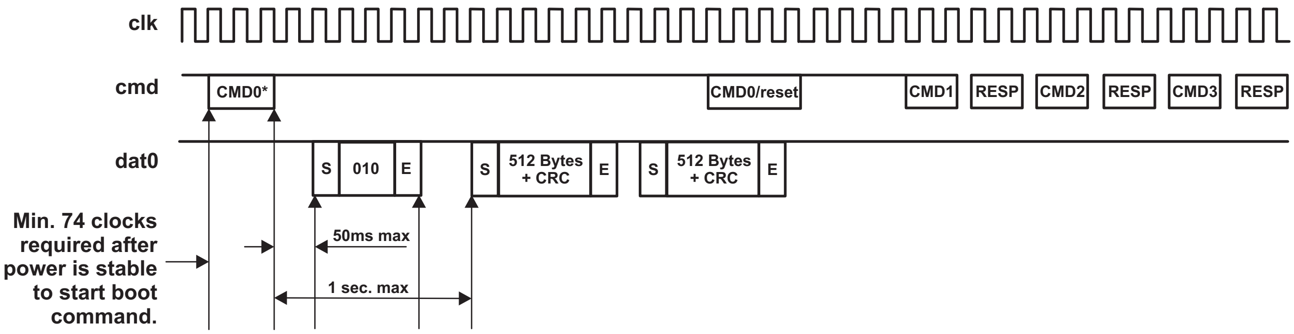
Boot Operation Mode allows the MMC/SD/SDIO host controller to read boot data from the connected slave (MMC device) by keeping CMD line low after power-on (or sending CMD0 with specific argument) before issuing CMD1. The data can be read from either boot area or user area, depending on register setting. Power-on boot defines a way for the boot-code to be accessed by the MMC/SD/SDIO host controller without an upper-level software driver, speeding the time it takes for a controller to access the boot code.

The two possible ways to issue a boot command (either issuing a CMD0 or driving the CMD line to 0 during the whole boot phase) are described in the following sections.

17.3.13.1 Boot Mode Using CMD0

Figure 17-31 shows the timing diagram of a boot sequence using CMD0.

Figure 17-31. Boot Mode With CMD0



*** Refer to MMC Specification for correct argument.**

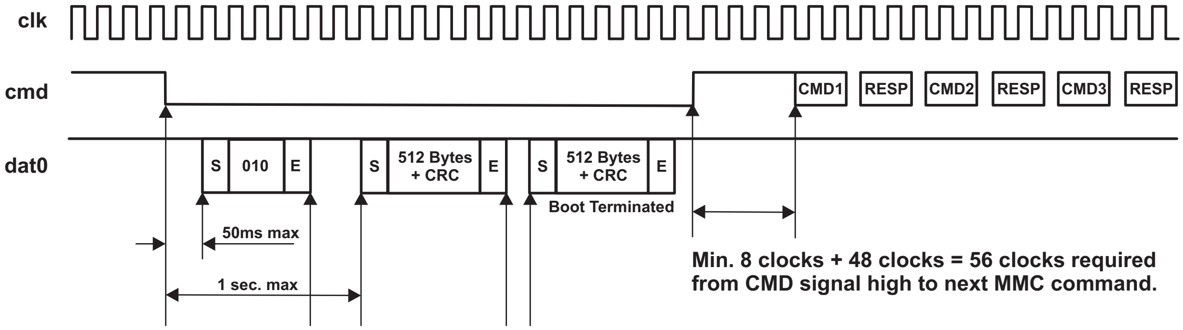
- Configure:
  - SD_CON[BOOT_CF0] to 0
  - SD_CON[BOOT_ACK] (if an acknowledge will be received) to 0x1
  - SD_BLK with the correct block length and number of block
  - SD_SYSCTL[DTO] for timeout

    If transfer is done in DDR mode also set SD_CON[DDR] to 1.
- Write register SD_ARG with correct argument (see MMC Specification).
- Write in SD_CMD register to start CMD0 transfer with these bit fields set:
  - INDX set to 0x00
  - DP set to ‘1’
  - DDIR set to ‘1’
  - MSBS set to ‘1’
  - BCE set to ‘1’
- If boot status is not received within the timing defined, the SD_STAT[DTO] will be generated. Otherwise the SD_STAT[BSR] is arisen.
- After the transfer is complete, the controller will generate the SD_STAT[TC], and then the system can emit another CMD0 (SD_CON[BOOT_ACK] previously cleared to 0x0) to exit the card from boot state.
- If the system wants to abort the boot sequence it must issue a CMD0 with SD_CMD[CMD_TYPE] set to 0x3 (SD_CON[BOOT_ACK] previously cleared to 0x0) during the transfer to abort transfer and enable card to exit from boot state.

17.3.13.2 Boot Mode With CMD Held Low

Figure 17-32 shows the timing diagram of a boot sequence with CMD line tied to 0.

Figure 17-32. Boot Mode With CMD Line Tied to 0



- Configure:
  - SD_CON[BOOT_CF0] and SD_CON[BOOT_ACK] (if an acknowledge will be received) to 0x1
  - SD_BLK with correct block length and number of block
  - SD_SYSCTL[DTO] for timeout

  If transfer is done in DDR mode also set SD_CON[DDR] to 1.
- Write in SD_CMD register to start boot sequence with:
  - DP set to '1'
  - DDIR set to '1'
  - MSBS set to '1'
  - BCE set to '1'

  This leads the controller to force CMD line to '0'.
- If the boot status is not received within the timing defined, the SD_STAT[DTO] will be generated. Otherwise the SD_STAT[BSR] is arisen.
- After the transfer is complete, the controller will generate the SD_STAT[TC], and then the system must clear SD_CON[BOOT_CF0] to 0x0 to release the CMD line and enable the card to exit from boot state.
- If the system wants to abort the boot sequence it must clear SD_CON[BOOT_CF0] to 0x0 during transfer to enable the card to exit from boot state.

17.3.14 CE-ATA Command Completion Disable Management

The MMC/SD/SDIO controller supports CE-ATA features, in particular the detection of command completion token. When a command that requires a command completion signal (SD_CON[12] CEATA and SD_CMD[2] ACEN set to 1) is launched, the host system is no longer allowed to emit a new command in parallel of data transfer unless it is a command completion disable token.

The settings to emit a command completion disable token follow:

- SD_CON[12] CEATA is set to 1.
- SD_CON[2] HR set to 1.
- Clear the SD_ARG register.
- Write into SD_CMD register with value 0000 0000h.

When a command completion disable token was emitted (that is, SD_STAT[0] CC received), the host system is again allowed to emit another type of command (for example a transfer abort command CMD12 to abort transfer).

A critical case can be met when command completion signal disable (CCSD) is emitted during the last data block transfer, the sequence on command line could be sent very close to command completion signal (CCS) token sent by the card.

Three cases can be met:

- CCS is receive just before CCSD is emitted:

  An interrupt CIRQ is generated with CCS detection, CCSD is transmitted to card then an interrupt CC is generated when CCSD ends. In this case, card consider the CCSD sequence.
- CCS is not generated or generated during the CCSD transfer:

  The CCS bit cannot be detected (conflict is not possible as they drive the same level on command line, then no CIRQ interrupt is generated; besides CC interrupt is generated when CCSD ends).
- CCS is generated without CCSD token required:

  Only the interrupt CIRQ is generated when CCS is detected.

17.3.15 Test Registers

Test registers are available to be compliant with SD Host controller specification. This feature is useful to generate interrupts manually for driver debugging. The Force Event register (SD_FE) is used to control the Error Interrupt Status and Auto CMD12 Error Status. The System Test register (SD_SYSTEST) is used to control the signals that connect to I/O pins when the module is configured in system test (SD_CON[4] MODE = 1) mode for boundary connectivity verification.

17.3.16 MMC/SD/SDIO Hardware Status Features

Table 17-16 summarizes the MMC/SD/SDIO hardware status features.

Table 17-16. MMC/SD/SDIO Hardware Status Features

Feature	Type	Register/Bit Field/Observability Control	Description
Interrupt flags		See Section 17.3.4.	
CMD line signal level	Status	[24] CLEV	Indicates the level of the cmd line
DAT lines signal level	Status	[23:20] DLEV	Indicates the level of the data lines
Buffer read enable	Status	[11] BRE	Readable data exists in the buffer.
Buffer write enable	Status	[10] BWE	Indicates whether there is enough space in the buffer to write BLEN bytes of data
Read transfer active	Status	[9] RTA	This status is used for detecting completion of a read transfer.
Write transfer active	Status	[8] WTA	This status indicates a write transfer active.
Data line active	Status	[2] DLA	Indicates whether the data lines are active
Command Inhibit (data lines)	Status	[1] DATI	Indicates whether issuing of command using data lines is allowed
Command inhibit (CMD line)	Status	[0] CMDI	Indicates whether issuing of command using CMD line is allowed

17.4 Low-Level Programming Models

17.4.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the module has to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the MMC/SD/SDIO modules.

Table 17-17. Global Init for Surrounding Modules

Surrounding Modules	Comments
PRCM	Module interface and functional clocks must be enabled. For more information, see Chapter 6, *Power, Reset, and Clock Management*.
Control module	Module-specific pad muxing and configuration must be set in the control module. See Chapter 7, *Control Module*.
(optional) MPU INTC	MPU INTC configuration must be done to enable the interrupts from the SD module. See Chapter 8, *Interrupts*.
(optional) EDMA	DMA configuration must be done to enable the module DMA channel requests. See Chapter 10, *EDMA*.
(optional) Interconnect	For more information about the interconnect configuration, see Chapter 4, *Interconnects*.

NOTE: The MPU interrupt controller and the EDMA configurations are necessary, if the interrupt and DMA based communication modes are used.

17.4.2 MMC/SD/SDIO Controller Initialization Flow

The next sections outline the four steps to initialize the MMC/SD/SDIO controller:

- Initialize Clocks
- Software reset of the controller
- Set module's hardware capabilities
- Set module's Idle and Wake-Up modes

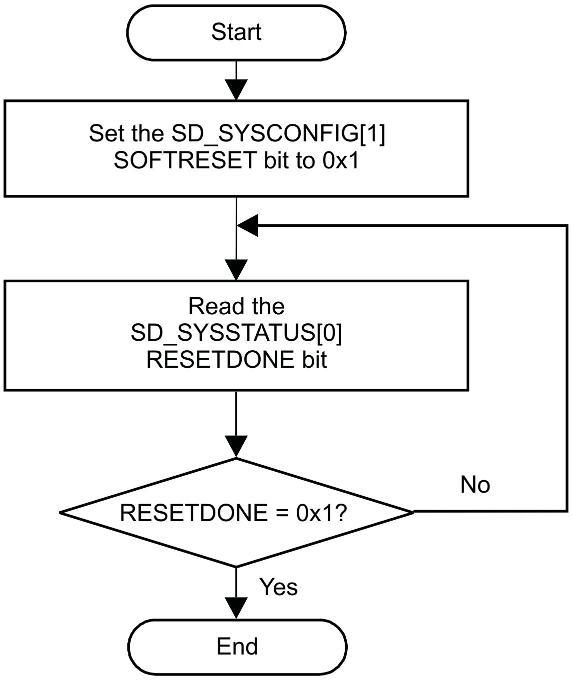
17.4.2.1 Enable OCP and CLKADPI Clocks

Prior to any SD register access one must enable the SD OCP clock and CLKADPI clock in PRCM module registers. For more information, see Chapter 6, *Power, Reset, and Clock Management*.

17.4.2.2 SD Soft Reset Flow

Figure 17-33 shows the soft reset process of MMC/SD/SDIO controller.

Figure 17-33. MMC/SD/SDIO Controller Software Reset Flow



17.4.2.3 Set SD Default Capabilities

Software must read capabilities (in boot ROM for instance) and is allowed to set (write) SD_CAPA[26:24] and SD_CUR_CAPA[23:0] registers before the MMC/SD/SDIO host driver is started.

17.4.2.4 Wake-Up Configuration

Table 17-18 details SD controller wake-up configuration.

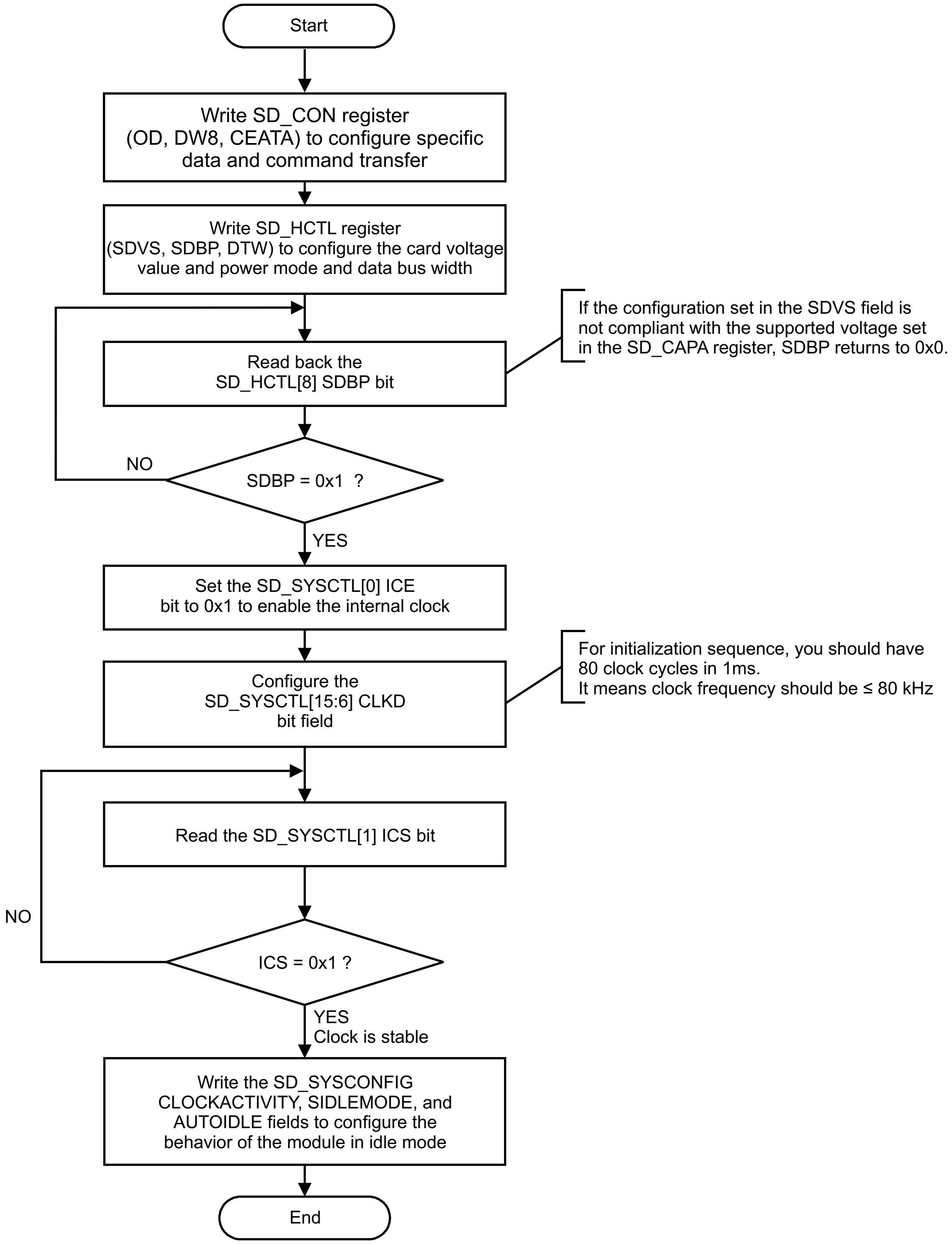
Table 17-18. MMC/SD/SDIO Controller Wake-Up Configuration

Step	Access Type	Register/Bit Field/Programming Model
Configure wake-up bit (if necessary).	W	SD_SYSCONFIG[2] ENAWAKEUP
Enable wake-up events on SD card interrupt (if necessary).	W	SD_HCTL[24] IWE
SDIO Card onlyEnable card interrupt (if necessary).	W	SD_IE[8] CIRQENABLE

17.4.2.5 MMC Host and Bus Configuration

Figure 17-34 details the MMC bus configuration process.

Figure 17-34. MMC/SD/SDIO Controller Bus Configuration Flow



17.4.3 Operational Modes Configuration

17.4.3.1 Basic Operations for MMC/SD/SDIO Host Controller

The MMC/SD/SDIO controller performs data transfers: data to card (referred to as write transfers) and data from card (referred to as read transfers).

The host controller requires transfers to run on a block-by-block basis, rather than on a DMA burst size basis. A single DMA request (or block request interrupt) is signaled for each block. Pipelining is supported as long as the block size is less than one half of the memory buffer size.

17.4.3.2 Card Detection, Identification, and Selection

Figure 17-35 and Figure 17-36 show the card identification and selection process.

Figure 17-35. MMC/SD/SDIO Controller Card Identification and Selection - Part 1

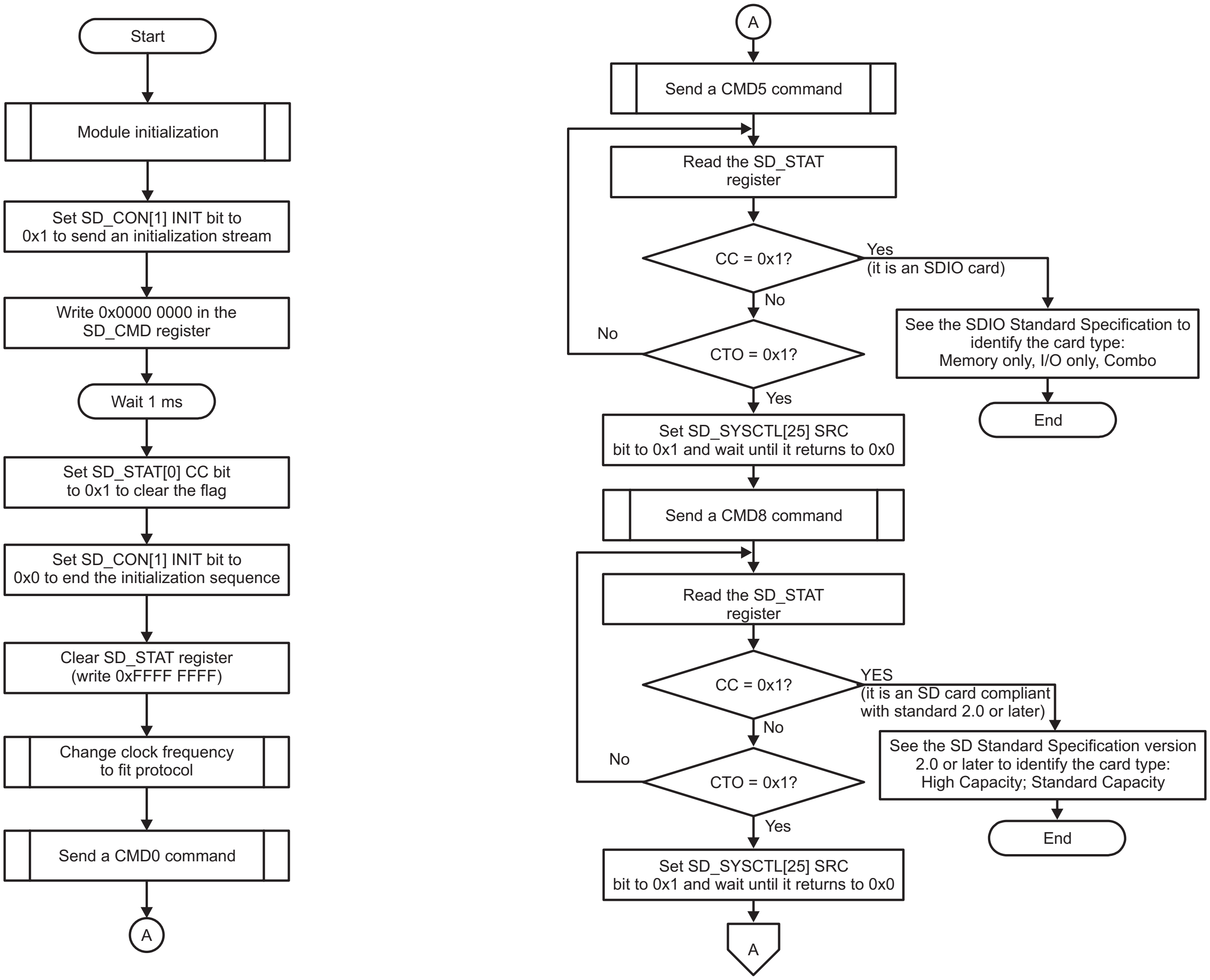
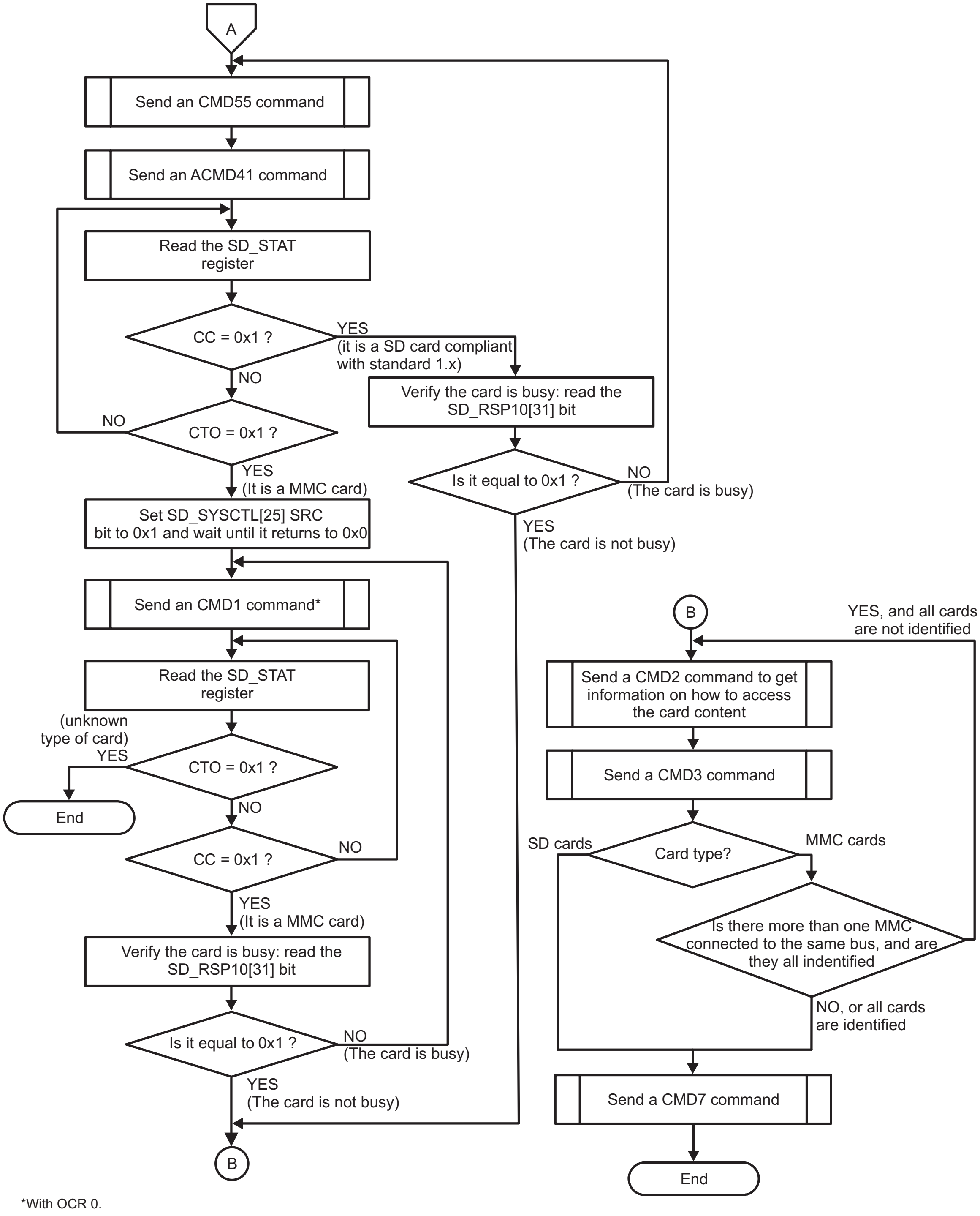


Figure 17-36. MMC/SD/SDIO Controller Card Identification and Selection - Part 2



*With OCR 0.

17.5 MMC/SD Registers

17.5.1 MMCSD Registers

Table 17-19 lists the memory-mapped registers for the MMCSD. All register offset addresses not listed in Table 17-19 should be considered as reserved locations and the register contents should not be modified.

Table 17-19. MMCSD Registers

Offset	Acronym	Register Name	Section
110h	SD_SYSCONFIG	System Configuration	Section 17.5.1.1
114h	SD_SYSSTATUS	System Status	Section 17.5.1.2
124h	SD_CSRE	Card status response error	Section 17.5.1.3
128h	SD_SYSTEST	System Test	Section 17.5.1.4
12Ch	SD_CON	Configuration	Section 17.5.1.5
130h	SD_PWCNT	Power counter	Section 17.5.1.6
200h	SD_SDMASA	SDMA System address:	Section 17.5.1.7
204h	SD_BLK	Transfer Length Configuration	Section 17.5.1.8
208h	SD_ARG	Command argument	Section 17.5.1.9
20Ch	SD_CMD	Command and transfer mode	Section 17.5.1.10
210h	SD_RSP10	Command Response 0 and 1	Section 17.5.1.11
214h	SD_RSP32	Command Response 2 and 3	Section 17.5.1.12
218h	SD_RSP54	Command Response 4 and 5	Section 17.5.1.13
21Ch	SD_RSP76	Command Response 6 and 7	Section 17.5.1.14
220h	SD_DATA	Data	Section 17.5.1.15
224h	SD_PSTATE	Present state	Section 17.5.1.16
228h	SD_HCTL	Host Control	Section 17.5.1.17
22Ch	SD_SYSCTL	SD system control	Section 17.5.1.18
230h	SD_STAT	SD interrupt status	Section 17.5.1.19
234h	SD_IE	SD interrupt enable	Section 17.5.1.20
238h	SD_ISE	SD interrupt enable set	Section 17.5.1.21
23Ch	SD_AC12	Auto CMD12 Error Status	Section 17.5.1.22
240h	SD_CAPA	Capabilities	Section 17.5.1.23
248h	SD_CUR_CAPA	Maximum current capabilities	Section 17.5.1.24
250h	SD_FE	Force Event	Section 17.5.1.25
254h	SD_ADMAES	ADMA Error Status	Section 17.5.1.26
258h	SD_ADMASAL	ADMA System address Low bits	Section 17.5.1.27
25Ch	SD_ADMASAH	ADMA System address High bits	Section 17.5.1.28
2FCh	SD_REV	Versions	Section 17.5.1.29

17.5.1.1 SD_SYSCONFIG Register (Offset = 110h) [reset = 0h]

SD_SYSCONFIG is shown in Figure 17-37 and described in Table 17-20.

Return to Summary Table.

This register allows controlling various parameters of the OCP interface.

Figure 17-37. SD_SYSCONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED		STANDBYMODE		RESERVED		CLOCKACTIVITY	
R-0h		R/W-0h		R-0h		R/W-0h	

7	6	5	4	3	2	1	0
RESERVED			SIDLEMODE		ENAWAKEUP	SOFTRESET	AUTOIDLE
R-0h			R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 17-20. SD_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	
13-12	STANDBYMODE	R/W	0h	Master interface power Management, standby/wait control. The bit field is only useful when generic parameter MADMA_EN (Master ADMA enable) is set as active, otherwise it is a read only register read a 0. 0h (R/W) = Force-standby. Mstandby is forced unconditionally. 1h (R/W) = No-standby. Mstandby is never asserted. 2h (R/W) = Smart-standby modelocal initiator standby status depends on local conditions, i.e. the module's functional requirement from the initiator. IP module shall not generate (initiator-related) wake-up events. 3h (R/W) = Smart-Standby wake-up-capable modelocal initiator standby status depends on local conditions, i.e. the module's functional requirement from the initiator. IP module can generate (master-related) wake-up events when in standby state. Mode is only relevant if the appropriate IP module "mwake-up" output is implemented. Functional clock is maintained. Interface clock may be switched off.
11-10	RESERVED	R	0h	
9-8	CLOCKACTIVITY	R/W	0h	Clocks activity during wake up mode period. Bit 8 is the Interface clock. Bit 9 is the Functional clock. 0h (R/W) = Interface and Functional clock may be switched off. 1h (R/W) = Interface clock is maintained. Functional clock may be switched-off. 2h (R/W) = Functional clock is maintained. Interface clock may be switched-off. 3h (R/W) = Interface and Functional clocks are maintained.
7-5	RESERVED	R	0h	

Table 17-20. SD_SYSCONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-3	SIDLEMODE	R/W	0h	Power management 0h (R/W) = If an idle request is detected, the MMC/SD/SDIO host controller acknowledges it unconditionally and goes in Inactive mode. Interrupt and DMA requests are unconditionally deasserted. 1h (R/W) = If an idle request is detected, the request is ignored and the module keeps on behaving normally. 2h (R/W) = If an idle request is detected, the module will switch to wake up mode based on its internal activity, and the wake up capability can be used if the wake up capability is enabled (bit SD_SYSCONFIG[2] ENAWAKEUP bit is set to 1). 3h (R/W) = Reserved.
2	ENAWAKEUP	R/W	0h	Wake-up feature control 0h (R/W) = Wake-up capability is disabled. 1h (R/W) = Wake-up capability is enabled.
1	SOFTRESET	R/W	0h	Software reset. The bit is automatically reset by the hardware. During reset, it always returns 0. 0h (W) = No effect 0h (R) = Normal mode 1h (W) = Trigger a module reset. 1h (R) = The module is reset.
0	AUTOIDLE	R/W	0h	Internal Clock gating strategy 0h (R/W) = Clocks are free-running. 1h (R/W) = Automatic clock gating strategy is applied, based on the interconnect and MMC interface activity.

17.5.1.2 SD_SYSSTATUS Register (Offset = 114h) [reset = 0h]

SD_SYSSTATUS is shown in Figure 17-38 and described in Table 17-21.

Return to Summary Table.

This register provides status information about the module excluding the interrupt status information.

Figure 17-38. SD_SYSSTATUS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							RESETDONE
R-0h							R-0h

Table 17-21. SD_SYSSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	RESETDONE	R	0h	Internal Reset Monitoring. Note the debounce clock, the interface clock and the functional clock shall be provided to the MMC/SD/SDIO host controller to allow the internal reset monitoring. 0h (R/W) = Internal module reset is on-going 1h (R/W) = Reset completed

17.5.1.3 SD_CSRE Register (Offset = 124h) [reset = 0h]

SD_CSRE is shown in Figure 17-39 and described in Table 17-22.

Return to Summary Table.

This register enables the host controller to detect card status errors of response type R1, R1b for all cards and of R5, R5b and R6 response for cards types SD or SDIO. When a bit SD_CSRE[i] is set to 1, if the corresponding bit at the same position in the response SD_RSP10[I] is set to 1, the host controller indicates a card error (SD_STAT[28] CERR bit) interrupt status to avoid the host driver reading the response register (SD_RSP10). No automatic card error detection for autoCMD12 is implemented; the host system has to check autoCMD12 response register (SD_RSP76) for possible card errors.

Figure 17-39. SD_CSRE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSRE																															
R/W-0h																															

Table 17-22. SD_CSRE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CSRE	R/W	0h	Card status response error

17.5.1.4 SD_SYSTEST Register (Offset = 128h) [reset = 0h]

SD_SYSTEST is shown in Figure 17-40 and described in Table 17-23.

Return to Summary Table.

This register is used to control the signals that connect to I/O pins when the module is configured in system test (SYSTEST) mode for boundary connectivity verification. In SYSTEST mode, a write into SD_CMD register will not start a transfer. The buffer behaves as a stack accessible only by the local host (push and pop operations). In this mode, the Transfer Block Size (SD_BLK[10:0] BLEN bits) and the Blocks count for current transfer (SD_BLK[31:16] NBLK bits) are needed to generate a Buffer write ready interrupt (SD_STAT[4] BWR bit) or a Buffer read ready interrupt (SD_STAT[5] BRR bit) and DMA requests if enabled.

Figure 17-40. SD_SYSTEST Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							OBI
R-0h							R/W-0h

15	14	13	12	11	10	9	8
SDCD	SDWP	WAKD	SSB	D7D	D6D	D5D	D4D
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
D3D	D2D	D1D	D0D	DDIR	CDAT	CDIR	MCKD
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 17-23. SD_SYSTEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	OBI	R/W	0h	Out-of-band interrupt (OBI) data value. 0h (R/W) = The out-of-band interrupt pin is driven low. 1h (R/W) = The out-of-band interrupt pin is driven high.
15	SDCD	R/W	0h	Card detect input signal (SDCD) data value 0h (R/W) = The card detect pin is driven low. 1h (R/W) = The card detect pin is driven high.
14	SDWP	R/W	0h	Write protect input signal (SDWP) data value 0h (R/W) = The write protect pin SDWP is driven low. 1h (R/W) = The write protect pin SDWP is driven high.
13	WAKD	R/W	0h	Wake request output signal data value. 0h (W) = The pin SWAKEUP is driven low. 0h (R) = No action. Returns 0. 1h (W) = The pin SWAKEUP is driven high. 1h (R) = No action. Returns 1.
12	SSB	R/W	0h	Set status bit. This bit must be cleared prior attempting to clear a status bit of the interrupt status register (SD_STAT). 0h (W) = Clear this SSB bit field. Writing 0 does not clear already set status bits. 0h (R) = No action. Returns 0. 1h (W) = Force to 1 all status bits of the interrupt status register (SD_STAT) only if the corresponding bit field in the Interrupt signal enable register (SD_ISE) is set. 1h (R) = No action. Returns 1.

Table 17-23. SD_SYSTEST Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	D7D	R/W	0h	DAT7 input/output signal data value. 0h (W) = If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT7 line is driven low. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. 0h (R) = If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT7 line (low). If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0. 1h (R) = If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT7 line (high) If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1. 1h (W) = If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT7 line is driven high. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.
10	D6D	R/W	0h	DAT6 input/output signal data value. 0h (W) = If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT6 line is driven low. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. 0h (R) = If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT6 line (low). If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0. 1h (R) = If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT6 line (high) If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1. 1h (W) = If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT6 line is driven high. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.
9	D5D	R/W	0h	DAT5 input/output signal data value. 0h (W) = If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT5 line is driven low. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. 0h (R) = If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT5 line (low). If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0. 1h (W) = If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT5 line is driven high. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. 1h (R) = If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT5 line (high) If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1.
8	D4D	R/W	0h	DAT4 input/output signal data value. 0h (W) = If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT4 line is driven low. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. 0h (R) = If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT4 line (low). If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0. 1h (R) = If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT4 line (high) If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1. 1h (W) = If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT4 line is driven high. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.

Table 17-23. SD_SYSTEST Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	D3D	R/W	0h	DAT3 input/output signal data value. 0h (W) = If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT3 line is driven low. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. 0h (R) = If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT3 line (low). If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0. 1h (W) = If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT3 line is driven high. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. 1h (R) = If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT3 line (high) If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1.
6	D2D	R/W	0h	DAT2 input/output signal data value. 0h (W) = If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT2 line is driven low. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. 0h (R) = If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT2 line (low). If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0. 1h (R) = If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT2 line (high) If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1. 1h (W) = If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT2 line is driven high. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.
5	D1D	R/W	0h	DAT1 input/output signal data value. 0h (W) = If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT1 line is driven low. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. 0h (R) = If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT1 line (low). If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0. 1h (R) = If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT1 line (high) If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1. 1h (W) = If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT1 line is driven high. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.
4	D0D	R/W	0h	DAT0 input/output signal data value. 0h (W) = If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT0 line is driven low. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. 0h (R) = If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT0 line (low). If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0. 1h (W) = If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT0 line is driven high. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. 1h (R) = If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT0 line (high) If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1.
3	DDIR	R/W	0h	Control of the DAT [7:0] pins direction. 0h (W) = The DAT lines are outputs (host to card). 0h (R) = No action. Returns 0. 1h (W) = The DAT lines are inputs (card to host). 1h (R) = No action. Returns 1.

Table 17-23. SD_SYSTEST Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	CDAT	R/W	0h	CMD input/output signal data value 0h (W) = If SD_SYSTEST[1] CDIR bit = 0 (output mode direction), the CMD line is driven low. If SD_SYSTEST[1] CDIR bit = 1 (input mode direction), no effect. 0h (R) = If SD_SYSTEST[1] CDIR bit = 1 (input mode direction), returns the value on the CMD line (low). If SD_SYSTEST[1] CDIR bit = 0 (output mode direction), returns 0 . 1h (R) = If SD_SYSTEST[1] CDIR bit = 1 (input mode direction), returns the value on the CMD line (high) If SD_SYSTEST[1] CDIR bit = 0 (output mode direction), returns 1 . 1h (W) = If SD_SYSTEST[1] CDIR bit = 0 (output mode direction), the CMD line is driven high. If SD_SYSTEST[1] CDIR bit = 1 (input mode direction), no effect.
1	CDIR	R/W	0h	Control of the CMD pin direction 0h (W) = The CMD line is an output (host to card). 0h (R) = No action. Returns 0. 1h (W) = The CMD line is an input (card to host) . 1h (R) = No action. Returns 1.
0	MCKD	R/W	0h	MMC clock output signal data value 0h (W) = The output clock is driven low. 0h (R) = No action. Returns 0. 1h (W) = The output clock is driven high. 1h (R) = No action. Returns 1.

17.5.1.5 SD_CON Register (Offset = 12Ch) [reset = 0h]

SD_CON is shown in Figure 17-41 and described in Table 17-24.

Return to Summary Table.

This register is used: To select the functional mode for any card. To send an initialization sequence to any card. To send an initialization sequence to any card. To enable the detection on the mmc_dat[1] signal of a card interrupt for SDIO cards only. It also configures the parameters related to the card detect and write protect input signals

Figure 17-41. SD_CON Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED		SDMA_LnE	DMA_MnS	DDR	BOOT_CF0	BOOT_ACK	CLKEXTFREE
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
PADEN	RESERVED		CEATA	CTPL	DVAL		WPP
R/W-0h	R-0h		R/W-0h	R/W-0h	R/W-0h		R/W-0h
7	6	5	4	3	2	1	0
CDP	MIT	DW8	MODE	STR	HR	INIT	OD
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 17-24. SD_CON Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21	SDMA_LnE	R/W	0h	Slave DMA Level/Edge Request. The waveform of the DMA request can be configured either edge sensitive with early de-assertion on first access to SD_DATA register or late de-assertion, request remains active until last allowed data written into SD_DATA. 0h (R/W) = Slave DMA edge sensitive. 1h (R/W) = Slave DMA level sensitive.
20	DMA_MnS	R/W	0h	DMA Master or Slave selection. When this bit is set and the controller is configured to use the DMA, Ocp master interface is used to get datas from system using ADMA2 procedure (direct access to the memory). This option is only available if generic parameter MADMA_EN is asserted to 1. 0h (R/W) = The controller is slave on data transfers with system. 1h (R/W) = Not available on this device.
19	DDR	R/W	0h	Dual Data Rate mode. When this register is set, the controller uses both clock edge to emit or receive data. Odd bytes are transmitted on falling edges and even bytes are transmitted on rise edges. It only applies on Data bytes and CRC, Start, end bits and CRC status are kept full cycle. This bit field is only meaningful and active for even clock divider ratio of SD_SYSCTL[CLKD], it is insensitive to SD_HCTL[HSPE] setting. Note: DDR mode is not supported on this device. Always set this bit to 0. 0h (R/W) = Standard modeData are transmitted on a single edge. 1h (R/W) = Data Bytes and CRC are transmitted on both edges.

Table 17-24. SD_CON Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	BOOT_CF0	R/W	0h	Boot Status Supported. This register is set when the CMD line needs to be forced to 0 for a boot sequence. CMD line is driven to 0 after writing in SD_CMD. The line is released when this bit field is de-asserted and aborts data transfer in case of a pending transaction. 0h (W) = CMD line forced to 0 is enabled. 0h (R) = CMD line not forced. 1h (R) = CMD line is released when it was previously forced to 0 by a boot sequence. 1h (W) = CMD line forced to 0 is enabled and will be active after writing into SD_CMD register.
17	BOOT_ACK	R/W	0h	Book acknowledge received. When this bit is set the controller should receive a boot status on DAT0 line after next command issued. If no status is received a data timeout will be generated. 0h (R/W) = No acknowledge to be received. 1h (R/W) = A boot status will be received on DAT0 line after issuing a command.
16	CLKEXTFREE	R/W	0h	External clock free running. This register is used to maintain card clock out of transfer transaction to enable slave module (for example to generate a synchronous interrupt on mmc_dat[1]). The Clock will be maintain only if SD_SYSCTL[2] CEN bit is set. 0h (R/W) = External card clock is cut off outside active transaction period. 1h (R/W) = External card clock is maintain even out of active transaction period only if SD_SYSCTL[2] CEN bit is set.
15	PADEN	R/W	0h	Control power for MMC lines. This register is only useful when MMC PADs contain power saving mechanism to minimize its leakage power. It works as a GPIO that directly control the ACTIVE pin of PADs. Excepted for mmc_dat[1] , the signal is also combine outside the module with the dedicated power control SD_CON[11] CTPL bit. 0h (R/W) = ADPIDLE module pin is not forced, it is automatically generated by the MMC fsms. 1h (R/W) = ADPIDLE module pin is forced to active state
14-13	RESERVED	R	0h	
12	CEATA	R/W	0h	CE-ATA control mode (MMC cards compliant with CE-ATA). This bit selects the active level of the out-of-band interrupt coming from MMC cards. The usage of the Out-of-Band signal (OBI) is not supported. 0h (R/W) = Standard MMC/SD/SDIO mode. 1h (R/W) = CE-ATA mode. Next commands are considered as CE-ATA commands.
11	CTPL	R/W	0h	Control Power for mmc_dat[1] line (SD cards). By default, this bit is cleared to 0 and the host controller automatically disables all the input buffers outside of a transaction to minimize the leakage current. SDIO cards. When this bit is set to 1, the host controller automatically disables all the input buffers except the buffer of mmc_dat[1] outside of a transaction in order to detect asynchronous card interrupt on mmc_dat[1] line and minimize the leakage current of the buffers. 0h (R/W) = Disable all the input buffers outside of a transaction. 1h (R/W) = Disable all the input buffers except the buffer of mmc_dat[1] outside of a transaction.

Table 17-24. SD_CON Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-9	DVAL	R/W	0h	Debounce filter value (all cards). This register is used to define a debounce period to filter the card detect input signal (SDCD). The usage of the card detect input signal (SDCD) is optional and depends on the system integration and the type of the connector housing that accommodates the card. 0h (R/W) = 33 us debounce period 1h (R/W) = 231 us debounce period 2h (R/W) = 1 ms debounce period 3h (R/W) = 8.4 ms debounce period
8	WPP	R/W	0h	Write protect polarity (for SD and SDIO cards only). This bit selects the active level of the write protect input signal (SDWP). The usage of the write protect input signal (SDWP) is optional and depends on the system integration and the type of the connector housing that accommodates the card. 0h = Active high level 1h = Active low level
7	CDP	R/W	0h	Card detect polarity (all cards). This bit selects the active level of the card detect input signal (SDCD). The usage of the card detect input signal (SDCD) is optional and depends on the system integration and the type of the connector housing that accommodates the card. 0h = Active high level 1h = Active low level
6	MIT	R/W	0h	MMC interrupt command (MMC cards only). This bit must be set to 1, when the next write access to the command register (SD_CMD) is for writing a MMC interrupt command (CMD40) requiring the command timeout detection to be disabled for the command response. 0h (R/W) = Command timeout enabled. 1h (R/W) = Command timeout disabled.
5	DW8	R/W	0h	8-bit mode MMC select (MMC cards only). For SD/SDIO cards, this bit must be cleared to 0. For MMC card, this bit must be set following a valid SWITCH command (CMD6) with the correct value and extend CSD index written in the argument. Prior to this command, the MMC card configuration register (CSD and EXT_CSD) must be verified for compliancy with MMC standard specification. 0h (R/W) = 1-bit or 4-bit data width 1h (R/W) = 8-bit data width
4	MODE	R/W	0h	Mode select (all cards). This bit selects the functional mode. 0h (R/W) = Functional mode. Transfers to the MMC/SD/SDIO cards follow the card protocol. The MMC clock is enabled. MMC/SD transfers are operated under the control of the SD_CMD register. 1h (R/W) = SYSTEST mode. SYSTEST mode. The signal pins are configured as general-purpose input/output and the 1024-byte buffer is configured as a stack memory accessible only by the local host or system DMA. The pins retain their default type (input, output or in-out). SYSTEST mode is operated under the control of the SYSTEST register.

Table 17-24. SD_CON Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	STR	R/W	0h	Stream command (MMC cards only). This bit must be set to 1 only for the stream data transfers (read or write) of the adtc commands. Stream read is a class 1 command (CMD11READ_DAT_UNTIL_STOP). Stream write is a class 3 command (CMD20WRITE_DAT_UNTIL_STOP). 0h (R/W) = Block oriented data transfer 1h (R/W) = Stream oriented data transfer
2	HR	R/W	0h	Broadcast host response (MMC cards only). This register is used to force the host to generate a 48-bit response for bc command type. It can be used to terminate the interrupt mode by generating a CMD40 response by the core. In order to have the host response to be generated in open drain mode, the register SD_CON[OD] must be set to 1. When SD_CON[12] CEATA bit is set to 1 and SD_ARG cleared to 0, when writing 0000 0000h into SD_CMD register, the host controller performs a 'command completion signal disable' token (i.e., mmc_cmd line held to 0 during 47 cycles followed by a 1). 0h (R/W) = The host does not generate a 48-bit response instead of a command. 1h (R/W) = The host generates a 48-bit response instead of a command or a command completion signal disable token.
1	INIT	R/W	0h	Send initialization stream (all cards). When this bit is set to 1, and the card is idle, an initialization sequence is sent to the card. An initialization sequence consists of setting the mmc_cmd line to 1 during 80 clock cycles. The initialization sequence is mandatory - but it is not required to do it through this bit - this bit makes it easier. Clock divider (SD_SYSCTL [15:6] CLKD bits) should be set to ensure that 80 clock periods are greater than 1ms. Note: In this mode, there is no command sent to the card and no response is expected. A command complete interrupt will be generated once the initialization sequence is completed. SD_STAT[0] CC bit can be polled. 0h (R/W) = The host does not send an initialization sequence 1h (R/W) = The host sends an initialization sequence
0	OD	R/W	0h	Card open drain mode (MMC cards only). This bit must be set to 1 for MMC card commands 1, 2, 3 and 40, and if the MMC card bus is operating in open-drain mode during the response phase to the command sent. Typically, during card identification mode when the card is either in idle, ready or ident state. It is also necessary to set this bit to 1, for a broadcast host response (see Broadcast host response register SD_CON[2] HR bit). 0h (R/W) = No open drain 1h (R/W) = Open drain or broadcast host response

17.5.1.6 SD_PWCNT Register (Offset = 130h) [reset = 0h]

SD_PWCNT is shown in Figure 17-42 and described in Table 17-25.

Return to Summary Table.

This register is used to program a mmc counter to delay command transfers after activating the PAD power, this value depends on PAD characteristics and voltage.

Figure 17-42. SD_PWCNT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">PWRCNT</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">R/W-0h</td></tr>
</table>

Table 17-25. SD_PWCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	PWRCNT	R/W	0h	Power counter register. This register is used to introduce a delay between the PAD ACTIVE pin assertion and the command issued. 0h (R/W) = No additional delay added 1h (R/W) = TCF delay (card clock period) 2h (R/W) = TCF x 2 delay (card clock period) FFFEh (R/W) = TCF x 65534 delay (card clock period) FFFFh (R/W) = TCF x 65535 delay (card clock period)

17.5.1.7 SD_SDMASA Register (Offset = 200h) [reset = 0h]

SD_SDMASA is shown in Figure 17-43 and described in Table 17-26.

Return to Summary Table.

This register is used to program a mmc counter to delay command transfers after activating the PAD power. This value depends on PAD characteristics and voltage.

Figure 17-43. SD_SDMASA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDMA_SYSADDR																															
R-0h																															

Table 17-26. SD_SDMASA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SDMA_SYSADDR	R	0h	This register contains the system memory address for a SDMA transfer. When the Host Controller stops a SDMA transfer, this register shall point to the system address of the next contiguous data position. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value. The Host Driver shall initialize this register before starting a SDMA transaction. After SDMA has stopped, the next system address of the next contiguous data position can be read from this register. The SDMA transfer waits at the every boundary specified by the Host SDMA Buffer Boundary in the Block Size register. The Host Controller generates DMA Interrupt to request the Host Driver to update this register. The Host Driver sets the next system address of the next data position to this register. When the most upper byte of this register (003h) is written, the Host Controller restarts the SDMA transfer. When restarting SDMA by the Resume command or by setting Continue Request in the Block Gap Control register, the Host Controller shall start at the next contiguous address stored here in the SDMA System Address register. ADMA does not use this register.

17.5.1.8 SD_BLK Register (Offset = 204h) [reset = 0h]

SD_BLK is shown in Figure 17-44 and described in Table 17-27.

Return to Summary Table.

This register shall be used for any card. SD_BLK[BLEN] is the block size register. SD_BLK[NBLK] is the block count register.

Figure 17-44. SD_BLK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NBLK															
R/W-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				BLEN											
R-0h				R/W-0h											

Table 17-27. SD_BLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NBLK	R/W	0h	Blocks count for current transfer. This register is enabled when Block count Enable (SD_CMD[1] BCE bit) is set to 1 and is valid only for multiple block transfers. Setting the block count to 0 results no data blocks being transferred. Note: The host controller decrements the block count after each block transfer and stops when the count reaches zero. This register can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value and write operation will be ignored. In suspend context, the number of blocks yet to be transferred can be determined by reading this register. When restoring transfer context prior to issuing a Resume command, The local host shall restore the previously saved block count. 0h (R/W) = Stop count 1h (R/W) = 1 block 2h (R/W) = 2 blocks FFFFh (R/W) = 65535 blocks
15-12	RESERVED	R	0h	
11-0	BLEN	R/W	0h	Transfer block size. This register specifies the block size for block data transfers. Read operations during transfers may return an invalid value, and write operations are ignored. When a CMD12 command is issued to stop the transfer, a read of the BLEN field after transfer completion (SD_STAT[1] TC bit set to 1) will not return the true byte number of data length while the stop occurs but the value written in this register before transfer is launched. 0h (R/W) = No data transfer 1h (R/W) = 1 byte block length 2h (R/W) = 2 bytes block length 3h (R/W) = 3 bytes block length 1FFh (R/W) = 511 bytes block length 200h (R/W) = 512 bytes block length 7FFh (R/W) = 2047 bytes block length 800h (R/W) = 2048 bytes block length

17.5.1.9 SD_ARG Register (Offset = 208h) [reset = 0h]

SD_ARG is shown in Figure 17-45 and described in Table 17-28.

Return to Summary Table.

This register contains command argument specified as bit 39-8 of Command-Format. These registers must be initialized prior to sending the command itself to the card (write action into the register SD_CMD register). Only exception is for a command index specifying stuff bits in arguments, making a write unnecessary.

Figure 17-45. SD_ARG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARG																															
R/W-0h																															

Table 17-28. SD_ARG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ARG	R/W	0h	Command argument bits [31:0] .

17.5.1.10 SD_CMD Register (Offset = 20Ch) [reset = 0h]

SD_CMD is shown in Figure 17-46 and described in Table 17-29.

Return to Summary Table.

SD_CMD[31:16] = the command register. SD_CMD[15:0] = the transfer mode. This register configures the data and command transfers. A write into the most significant byte send the command. A write into SD_CMD[15:0] registers during data transfer has no effect. This register can be used for any card. In SYSTEST mode, a write into SD_CMD register will not start a transfer.

Figure 17-46. SD_CMD Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">INDX</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="6">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="2">CMD_TYPE</td><td>DP</td><td>CICE</td><td>CCCE</td><td>RESERVED</td><td colspan="2">RSP_TYPE</td></tr>
<tr><td colspan="2">R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R-0h</td><td colspan="2">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">RESERVED</td><td>MSBS</td><td>DDIR</td><td>RESERVED</td><td>ACEN</td><td>BCE</td><td>DE</td></tr>
<tr><td colspan="2">R-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 17-29. SD_CMD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	
29-24	INDX	R/W	0h	Command index binary encoded value from 0 to 63 specifying the command number send to card (CMD0 or ACMD0 to CMD63 or ACMD63).
23-22	CMD_TYPE	R/W	0h	Command type. This register specifies three types of special commands: Suspend, Resume and Abort. These bits shall be cleared to 0b00 for all other commands. 0h (R/W) = Others commands 1h (R/W) = Upon CMD52 "Bus Suspend" operation 2h (R/W) = Upon CMD52 "Function Select" operation 3h (R/W) = Upon CMD12 or CMD52 "I/O Abort" command
21	DP	R/W	0h	Data present select. This register indicates that data is present and mmc_dat line shall be used. It must be cleared to 0 in the following conditions: Command using only mmc_cmd line. Command with no data transfer but using busy signal on mmc_dat0. Resume command. 0h (R/W) = Command with no data transfer 1h (R/W) = Command with data transfer
20	CICE	R/W	0h	Command Index check enable. This bit must be set to 1 to enable index check on command response to compare the index field in the response against the index of the command. If the index is not the same in the response as in the command, it is reported as a command index error (SD_STAT[19] CIE bit set to1) Note: The CICE bit cannot be configured for an Auto CMD12, then index check is automatically checked when this command is issued. 0h (R/W) = Index check disable 1h (R/W) = Index check enable

Table 17-29. SD_CMD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	CCCE	R/W	0h	Command CRC check enable. This bit must be set to 1 to enable CRC7 check on command response to protect the response against transmission errors on the bus. If an error is detected, it is reported as a command CRC error (SD_STAT[17] CCRC bit set to 1). Note: The CCCE bit cannot be configured for an Auto CMD12, and then CRC check is automatically checked when this command is issued. 0h (R/W) = CRC7 check disable 1h (R/W) = CRC7 check enable
18	RESERVED	R	0h	
17-16	RSP_TYPE	R/W	0h	Response type. This bits defines the response type of the command. 0h (R/W) = No response 1h (R/W) = Response Length 136 bits 2h (R/W) = Response Length 48 bits 3h (R/W) = Response Length 48 bits with busy after response
15-6	RESERVED	R	0h	
5	MSBS	R/W	0h	Multi/Single block select. This bit must be set to 1 for data transfer in case of multi block command. For any others command this bit shall be cleared to 0. 0h (R/W) = Single block. If this bit is 0, it is not necessary to set the register SD_BLK[31:16] NBLK bits. 1h (R/W) = Multi block. When Block Count is disabled (SD_CMD[1] BCE bit is cleared to 0) in Multiple block transfers (SD_CMD[5] MSBS bit is set to 1), the module can perform infinite transfer.
4	DDIR	R/W	0h	Data transfer Direction. Select This bit defines either data transfer will be a read or a write. 0h (R/W) = Data Write (host to card) 1h (R/W) = Data Read (card to host)
3	RESERVED	R	0h	
2	ACEN	R/W	0h	Auto CMD12 Enable (SD cards only). When this bit is set to 1, the host controller issues a CMD12 automatically after the transfer completion of the last block. The Host Driver shall not set this bit to issue commands that do not require CMD12 to stop data transfer. In particular, secure commands do not require CMD12. For CE-ATA commands (SD_CON[12] CEATA bit set to 1), auto CMD12 is useless therefore when this bit is set the mechanism to detect command completion signal, named CCS, interrupt is activated. 0h (R/W) = Auto CMD12 disable 1h (R/W) = Auto CMD12 enable or CCS detection enabled.
1	BCE	R/W	0h	Block Count Enable (Multiple block transfers only). This bit is used to enable the block count register (SD_BLK [31:16] NBLK bits). When Block Count is disabled (SD_CMD[1] BCE bit is cleared to 0) in Multiple block transfers (SD_CMD[5] MSBS bits is set to 1), the module can perform infinite transfer. 0h (R/W) = Block count disabled for infinite transfer. 1h (R/W) = Block count enabled for multiple block transfer with known number of blocks
0	DE	R/W	0h	DMA Enable. This bit is used to enable DMA mode for host data access. 0h (R/W) = DMA mode disable 1h (R/W) = DMA mode enable

17.5.1.11 SD_RSP10 Register (Offset = 210h) [reset = 0h]

SD_RSP10 is shown in Figure 17-47 and described in Table 17-30.

Return to Summary Table.

This 32-bit register holds bits positions [31:0] of command response type R1, R1b, R2, R3, R4, R5, R5b, or R6.

Figure 17-47. SD_RSP10 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RSP1</td><td colspan="16">RSP0</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">R-0h</td></tr>
</table>

Table 17-30. SD_RSP10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RSP1	R	0h	Command Response [31:16]
15-0	RSP0	R	0h	Command Response [15:0]

17.5.1.12 SD_RSP32 Register (Offset = 214h) [reset = 0h]

SD_RSP32 is shown in Figure 17-48 and described in Table 17-31.

Return to Summary Table.

This 32-bit register holds bits positions [63:32] of command response type R2.

Figure 17-48. SD_RSP32 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSP3																RSP2															
R-0h																R-0h															

Table 17-31. SD_RSP32 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RSP3	R	0h	Command Response [63:48]
15-0	RSP2	R	0h	Command Response [47:32]

17.5.1.13 SD_RSP54 Register (Offset = 218h) [reset = 0h]

SD_RSP54 is shown in Figure 17-49 and described in Table 17-32.

Return to Summary Table.

This 32-bit register holds bits positions [95:64] of command response type R2.

Figure 17-49. SD_RSP54 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RSP5</td><td colspan="16">RSP4</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">R-0h</td></tr>
</table>

Table 17-32. SD_RSP54 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RSP5	R	0h	Command Response [95:80]
15-0	RSP4	R	0h	Command Response [79:64]

17.5.1.14 SD_RSP76 Register (Offset = 21Ch) [reset = 0h]

SD_RSP76 is shown in Figure 17-50 and described in Table 17-33.

Return to Summary Table.

This 32-bit register holds bits positions [127:96] of command response type R2.

Figure 17-50. SD_RSP76 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RSP7</td><td colspan="16">RSP6</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">R-0h</td></tr>
</table>

Table 17-33. SD_RSP76 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RSP7	R	0h	Command Response [127:112]
15-0	RSP6	R	0h	Command Response [111:96]

17.5.1.15 SD_DATA Register (Offset = 220h) [reset = 0h]

SD_DATA is shown in Figure 17-51 and described in Table 17-34.

Return to Summary Table.

This register is the 32-bit entry point of the buffer for read or write data transfers. The buffer size is 32bitsx256(1024 bytes). Bytes within a word are stored and read in little endian format. This buffer can be used as two 512 byte buffers to transfer data efficiently without reducing the throughput. Sequential and contiguous access is necessary to increment the pointer correctly. Random or skipped access is not allowed. In little endian, if the local host accesses this register byte-wise or 16bit-wise, the least significant byte (bits [7:0]) must always be written/read first. The update of the buffer address is done on the most significant byte write for full 32-bit DATA register or on the most significant byte of the last word of block transfer. Example 1Byte or 16-bit access: Mbyteen[3:0]=0001 (1-byte) => Mbyteen[3:0]=0010 (1-byte) => Mbyteen[3:0]=1100 (2-bytes) OK. Mbyteen[3:0]=0001 (1-byte) => Mbyteen[3:0]=0010 (1-byte) => Mbyteen[3:0]=0100 (1-byte) OK. Mbyteen[3:0]=0001 (1-byte) => Mbyteen[3:0]=0010 (1-byte) => Mbyteen[3:0]=1000 (1-byte) Bad.

Figure 17-51. SD_DATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
R/W-0h																															

Table 17-34. SD_DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R/W	0h	Data register [31:0]. In functional mode (SD_CON[4] MODE bit set to the default value 0): A read access to this register is allowed only when the buffer read enable status is set to 1 (SD_PSTATE[11] BRE bit), otherwise a bad access (SD_STAT[29] BADA bit) is signaled. A write access to this register is allowed only when the buffer write enable status is set to 1 (SD_PSTATE[10] BWE bit), otherwise a bad access (SD_STAT[29] BADA bit) is signaled and the data is not written.

17.5.1.16 SD_PSTATE Register (Offset = 224h) [reset = 0h]

SD_PSTATE is shown in Figure 17-52 and described in Table 17-35.

Return to Summary Table.

The Host can get the status of the Host controller from this 32-bit read only register.

Figure 17-52. SD_PSTATE Register

31	30	29	28	27	26	25	24
RESERVED							CLEV
R-0h							R-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DLEV				WP	CDPL	CSS	CINS
R-0h				R-0h	R-0h	R-0h	R-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED				BRE	BWE	RTA	WTA
R-0h				R-0h	R-0h	R-0h	R-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					DLA	DATI	CMDI
R-0h					R-0h	R-0h	R-0h

Table 17-35. SD_PSTATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	
24	CLEV	R	0h	mmc_cmd line signal level. This status is used to check the mmc_cmd line level to recover from errors, and for debugging. The value of this register after reset depends on the mmc_cmd line level at that time. 0h (R/W) = The mmc_cmd line level is 0. 1h (R/W) = The mmc_cmd line level is 1.
23-20	DLEV	R	0h	mmc_dat [3:0] line signal level mmc_dat3 equal to or greater than bit 23. mmc_dat2 equal to or greater than bit 22. mmc_dat1 equal to or greater than bit 21. mmc_dat0 equal to or greater than bit 20. This status is used to check mmc_dat line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from mmc_dat0 . The value of these registers after reset depends on the mmc_dat lines level at that time.
19	WP	R	0h	Write Protect. MMC/SD/SDIO1 only. SDIO cards only. This bit reflects the write protect input pin (SDWP) level. The value of this register after reset depends one the protect input pin (SDWP) level at that time. 0h (R/W) = If SD_CON[8] WPP is cleared to 0 (default), the card is write protected, otherwise the card is not write protected. 1h (R/W) = If SD_CON[8] WPP is cleared to 0 (default), the card is not write protected, otherwise the card is write protected.

Table 17-35. SD_PSTATE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	CDPL	R	0h	Card Detect Pin Level. MMC/SD/SDIO1 only. SDIO cards only. This bit reflects the inverse value of the card detect input pin (SDCD). Debouncing is not performed on this bit and is valid only when Card State is stable. (SD_PSTATE[17] is set to 1). This bit must be debounced by software. The value of this register after reset depends on the card detect input pin (SDCD) level at that time. 0h (R/W) = The value of the card detect input pin (SDCD) is 1. 1h (R/W) = The value of the card detect input pin (SDCD) is 0.
17	CSS	R	0h	Card State Stable. This bit is used for testing. It is set to 1 only when Card Detect Pin Level is stable (SD_PSTATE[18] CPDL). Debouncing is performed on the card detect input pin (SDCD) to detect card stability. This bit is not affected by software reset. 0h (R/W) = Reset or Debouncing. 1h (R/W) = Reset or Debouncing.
16	CINS	R	0h	Card inserted. This bit is the debounced value of the card detect input pin (SDCD). An inactive to active transition of the card detect input pin (SDCD) will generate a card insertion interrupt (SD_STAT[CINS]). A active to inactive transition of the card detect input pin (SDCD) will generate a card removal interrupt (SD_STAT[REM]). This bit is not affected by a software reset. 0h (R/W) = If SD_CON[CDP] is cleared to 0 (default), no card is detected. The card may have been removed from the card slot. If SD_CON[CDP] is set to 1, the card has been inserted. 1h (R/W) = If SD_CON[CDP] is cleared to 0 (default), the card has been inserted from the card slot. If SD_CON[CDP] is set to 1, no card is detected. The card may have been removed from the card slot.
15-12	RESERVED	R	0h	
11	BRE	R	0h	Buffer read enable. This bit is used for non-DMA read transfers. It indicates that a complete block specified by SD_BLK [10:0] BLEN bits has been written in the buffer and is ready to be read. It is cleared to 0 when the entire block is read from the buffer. It is set to 1 when a block data is ready in the buffer and generates the Buffer read ready status of interrupt (SD_STAT[5] BRR bit). 0h (R/W) = Read BLEN bytes disable 1h (R/W) = Read BLEN bytes enable. Readable data exists in the buffer.
10	BWE	R	0h	Buffer Write enable. This status is used for non-DMA write transfers. It indicates if space is available for write data. 0h (R/W) = There is no room left in the buffer to write BLEN bytes of data. 1h (R/W) = There is enough space in the buffer to write BLEN bytes of data.

Table 17-35. SD_PSTATE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	RTA	R	0h	Read transfer active. This status is used for detecting completion of a read transfer. It is set to 1 after the end bit of read command or by activating a continue request (SD_HCTL[17] CR bit) following a stop at block gap request. This bit is cleared to 0 when all data have been read by the local host after last block or after a stop at block gap request. 0h (R/W) = No valid data on the mmc_dat lines. 1h (R/W) = Read data transfer on going.
8	WTA	R	0h	Write transfer active. This status indicates a write transfer active. It is set to 1 after the end bit of write command or by activating a continue request (SD_HCTL[17] CR bit) following a stop at block gap request. This bit is cleared to 0 when CRC status has been received after last block or after a stop at block gap request. 0h (R/W) = No valid data on the mmc_dat lines. 1h (R/W) = Write data transfer on going.
7-3	RESERVED	R	0h	
2	DLA	R	0h	mmc_dat line active. This status bit indicates whether one of the mmc_dat lines is in use. In the case of read transactions (card to host)This bit is set to 1 after the end bit of read command or by activating continue request SD_HCTL[17] CR bit. This bit is cleared to 0 when the host controller received the end bit of the last data block or at the beginning of the read wait mode. In the case of write transactions (host to card)This bit is set to 1 after the end bit of write command or by activating continue request SD_HCTL[17] CR bit. This bit is cleared to 0 on the end of busy event for the last block. The host controller must wait 8 clock cycles with line not busy to really consider not "busy state" or after the busy block as a result of a stop at gap request. 0h (R/W) = mmc_dat line inactive 1h (R/W) = mmc_dat line active
1	DATI	R	0h	Command inhibit (mmc_dat). This status bit is generated if either mmc_dat line is active (SD_PSTATE[2] DLA bit) or Read transfer is active (SD_PSTATE[9] RTA bit) or when a command with busy is issued. This bit prevents the local host to issue a command. A change of this bit from 1 to 0 generates a transfer complete interrupt (SD_STAT[1] TC bit). 0h (R/W) = Issuing of command using the mmc_dat lines is allowed 1h (R/W) = Issuing of command using mmc_dat lines is not allowed
0	CMDI	R	0h	Command inhibit(mmc_cmd). This status bit indicates that the mmc_cmd line is in use. This bit is cleared to 0 when the most significant byte is written into the command register. This bit is not set when Auto CMD12 is transmitted. This bit is cleared to 0 in either the following cases: After the end bit of the command response, excepted if there is a command conflict error (SD_STAT[17] CCRC bit or SD_STAT[18] CEB bit set to 1) or a Auto CMD12 is not executed (SD_AC12[0] ACNE bit). After the end bit of the command without response (SD_CMD [17:16] RSP_TYPE bits set to "00"). In case of a command data error is detected (SD_STAT[19] CTO bit set to 10, this register is not automatically cleared. 0h (R/W) = Issuing of command using mmc_cmd line is allowed 1h (R/W) = Issuing of command using mmc_cmd line is not allowed

17.5.1.17 SD_HCTL Register (Offset = 228h) [reset = 0h]

SD_HCTL is shown in Figure 17-53 and described in Table 17-36.

Return to Summary Table.

This register defines the host controls to set power, wake-up and transfer parameters. SD_HCTL[31:24] = Wake-up control. SD_HCTL[23:16] = Block gap control. SD_HCTL[15:8] = Power control. SD_HCTL[7:0] = Host control. If your device does not support MMC cards, then those bits in this register which are meant for MMC card use should be assumed to be reserved.

Figure 17-53. SD_HCTL Register

31	30	29	28	27	26	25	24
RESERVED				OBWE	REM	INS	IWE
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				IBG	RWC	CR	SBGR
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED				SDVS			SDBP
R-0h				R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
CDSS	CDTL	RESERVED	DMAS		HSPE	DTW	RESERVED
R/W-0h	R/W-0h	R-0h	R/W-0h		R/W-0h	R/W-0h	R-0h

Table 17-36. SD_HCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27	OBWE	R/W	0h	Wake-up event enable for 'out-of-band' Interrupt. This bit enables wake-up events for 'out-of-band' assertion. Wake-up is generated if the wake-up feature is enabled (SD_SYSCONFIG[2] ENAWAKEUP bit). The write to this register is ignored when SD_CON[14] OBIE bit is not set. 0h (R/W) = Disable wake-up on 'out-of-band' Interrupt 1h (R/W) = Enable wake-up on 'out-of-band' Interrupt
26	REM	R/W	0h	Wake-up event enable on SD card removal. This bit enables wake-up events for card removal assertion. Wake-up is generated if the wake-up feature is enabled (SD_SYSCONFIG[2] ENAWAKEUP bit). 0h (R/W) = Disable wake-up on card removal 1h (R/W) = Enable wake-up on card removal
25	INS	R/W	0h	Wake-up event enable on SD card insertion This bit enables wake-up events for card insertion assertion. Wake-up is generated if the wake-up feature is enabled (SD_SYSCONFIG[2] ENAWAKEUP bit). 0h (R/W) = Disable wake-up on card insertion 1h (R/W) = Enable wake-up on card insertion
24	IWE	R/W	0h	Wake-up event enable on SD card interrupt. This bit enables wake-up events for card interrupt assertion. Wake-up is generated if the wake-up feature is enabled (SD_SYSCONFIG[2] ENAWAKEUP bit) and enable status bit is set (SD_IE[8] CIRQ_ENABLE bit). 0h (R/W) = Disable wake-up on card interrupt 1h (R/W) = Enable wake-up on card interrupt
23-20	RESERVED	R	0h	

Table 17-36. SD_HCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	IBG	R/W	0h	Interrupt block at gap. This bit is valid only in 4-bit mode of SDIO card to enable interrupt detection in the interrupt cycle at block gap for a multiple block transfer. For MMC cards and for SD card this bit should be cleared to 0. 0h (R/W) = Disable interrupt detection at the block gap in 4-bit mode 1h (R/W) = Enable interrupt detection at the block gap in 4-bit mode
18	RWC	R/W	0h	Read wait control. The read wait function is optional only for SDIO cards. If the card supports read wait, this bit must be enabled, then requesting a stop at block gap (SD_HCTL[16] SBGR bit) generates a read wait period after the current end of block. Be careful, if read wait is not supported it may cause a conflict on mmc_dat line. 0h (R/W) = Disable read wait control. Suspend/resume cannot be supported. 1h (R/W) = Enable read wait control
17	CR	R/W	0h	Continue request. This bit is used to restart a transaction that was stopped by requesting a stop at block gap (SD_HCTL[16] SBGR bit). Set this bit to 1 restarts the transfer. The bit is automatically cleared to 0 by the host controller when transfer has restarted, that is, mmc_dat line is active (SD_PSTATE[2] DLA bit) or transferring data (SD_PSTATE[8] WTA bit). The Stop at block gap request must be disabled (SD_HCTL[16] SBGR bit =0) before setting this bit. 0h (R/W) = No affect 1h (R/W) = Transfer restart
16	SBGR	R/W	0h	Stop at block gap request. This bit is used to stop executing a transaction at the next block gap. The transfer can restart with a continue request (SD_HCTL[17] CR bit) or during a suspend/resume sequence. In case of read transfer, the card must support read wait control. In case of write transfer, the host driver shall set this bit after all block data written. Until the transfer completion (SD_STAT[1] TC bit set to 1), the host driver shall leave this bit set to 1.If this bit is set, the local host shall not write to the data register (SD_DATA). 0h (R/W) = Transfer mode 1h (R/W) = Stop at block gap
15-12	RESERVED	R	0h	
11-9	SDVS	R/W	0h	SD bus voltage select (All cards). The host driver should set these bits to select the voltage level for the card according to the voltage supported by the system (SD_CAPA[26] VS18 bit, SD_CAPA[25] VS30 bit, SD_CAPA[24] VS33 bit) before starting a transfer. If MMCSD 2: This field must be set to 5h. If MMCSD 3: This field must be set to 5h. 5h (R/W) = 1.8 V (Typical) 6h (R/W) = 3.0 V (Typical) 7h (R/W) = 3.3 V (Typical)

Table 17-36. SD_HCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	SDBP	R/W	0h	SD bus power. Before setting this bit, the host driver shall select the SD bus voltage (SD_HCTL [11:9] SDVS bits). If the host controller detects the No card state, this bit is automatically cleared to 0. It will not automatically change to 1 if the card is inserted. This needs to be done with software. If the module is power off, a write in the command register (SD_CMD) will not start the transfer. A write to this bit has no effect if the selected SD bus voltage is not supported according to capability register (SD_CAPA[VS*]). 0h (R/W) = Power off 1h (R/W) = Power on
7	CDSS	R/W	0h	Card Detect Signal Selection. This bit selects source for the card detection. When the source for the card detection is switched, the interrupt should be disabled during the switching period by clearing the Interrupt Status/Signal Enable register in order to mask unexpected interrupt being caused by the glitch. The Interrupt Status/Signal Enable should be disabled during over the period of debouncing. 0h (R/W) = SDCD# is selected (for normal use). 1h (R/W) = The Card Detect Test Level is selected (for test purposes).
6	CDTL	R/W	0h	Card Detect Test Level. This bit is enabled while the Card Detect Signal Selection is set to 1 and it indicates card inserted or not. 0 = No card 1 = Card inserted.
5	RESERVED	R	0h	
4-3	DMAS	R/W	0h	DMA Select. One of the supported DMA modes can be selected. The host driver shall check support of DMA modes by referencing the Capabilities register. Use of selected DMA is determined by DMA Enable of the Transfer Mode register. This register is only meaningful when MADMA_EN is set to 1. When MADMA_EN is cleared to 0 the bit field is read only and returned value is 0. 0h (R/W) = Reserved 1h (R/W) = Reserved 2h (R/W) = 32-bit Address ADMA2 is selected. 3h (R/W) = Reserved
2	HSPE	R/W	0h	High Speed Enable. Before setting this bit, the Host Driver shall check the High Speed Support in the Capabilities register. If this bit is cleared to 0 (default), the Host Controller outputs CMD line and DAT lines at the falling edge of the SD Clock. If this bit is set to 1, the Host Controller outputs CMD line and DAT lines at the rising edge of the SD Clock. This bit shall not be set when dual data rate mode is activated in SD_CON[DDR]. 0h (R/W) = Normal speed mode 1h (R/W) = High speed mode
1	DTW	R/W	0h	Data transfer width. This bit must be set following a valid SET_BUS_WIDTH command (ACMD6) with the value written in bit 1 of the argument. Prior to this command, the SD card configuration register (SCR) must be verified for the supported bus width by the SD card. 0h (R/W) = 1-bit Data width (mmc_dat0 used) 1h (R/W) = 4-bit Data width (mmc_dat[3:0] used)
0	RESERVED	R	0h	

17.5.1.18 SD_SYSCTL Register (Offset = 22Ch) [reset = 0h]

SD_SYSCTL is shown in Figure 17-54 and described in Table 17-37.

Return to Summary Table.

This register defines the system controls to set software resets, clock frequency management and data timeout. SD_SYSCTL[31:24] = Software resets. SD_SYSCTL[23:16] = Timeout control. SD_SYSCTL[15:0] = Clock control.

Figure 17-54. SD_SYSCTL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					SRD	SRC	SRA	RESERVED				DTO			
R-0h					R/W-0h	R/W-0h	R/W-0h	R-0h				R/W-0h			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKD										RESERVED			CEN	ICS	ICE
R/W-0h										R-0h			R/W-0h	R-0h	R/W-0h

Table 17-37. SD_SYSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	
26	SRD	R/W	0h	Software reset for mmc_dat line. This bit is set to 1 for reset and released to 0 when completed. Due to additional implementation logic, the reset does not immediately start when asserted. The proper procedure is: (a) Set to 1 to start reset, (b) Poll for 1 to identify start of reset, and (c) Poll for 0 to identify reset is complete. mmc_dat finite state machine in both clock domain are also reset. These registers are cleared by the SD_SYSCTL[26] SRD bit: SD_DATA. SD_PSTATEBRE, BWE, RTA, WTA, DLA and DATI. SD_HCTLSBGR and CR. SD_STATBRR, BWR, BGE and TC Interconnect and MMC buffer data management is reinitialized. Note: If a soft reset is issued when an interrupt is asserted, data may be lost. 0h (R/W) = Reset completed 1h (R/W) = Software reset for mmc_dat line
25	SRC	R/W	0h	Software reset for mmc_cmd line. This bit is set to 1 for reset and released to 0 when completed. Due to additional implementation logic, the reset does not immediately start when asserted. The proper procedure is: (a) Set to 1 to start reset, (b) Poll for 1 to identify start of reset, and (c) Poll for 0 to identify reset is complete. mmc_cmd finite state machine in both clock domain are also reset. These registers are cleared by the SD_SYSCTL[25] SRC bit: SD_PSTATECMDI. SD_STATCC Interconnect and MMC command status management is reinitialized. Note: If a soft reset is issued when an interrupt is asserted, data may be lost. 0h (R/W) = Reset completed 1h (R/W) = Software reset for mmc_cmd line
24	SRA	R/W	0h	Software reset for all. This bit is set to 1 for reset , and released to 0 when completed. This reset affects the entire host controller except for the card detection circuit and capabilities registers. 0h (R/W) = Reset completed 1h (R/W) = Software reset for all the design
23-20	RESERVED	R	0h	

Table 17-37. SD_SYSCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-16	DTO	R/W	0h	Data timeout counter value and busy timeout. This value determines the interval by which mmc_dat lines timeouts are detected. The host driver needs to set this bit field based on: The maximum read access time (NAC) (Refer to the SD Specification Part1 Physical Layer). The data read access time values (TAAC and NSAC) in the card specific data register (CSD) of the card. The timeout clock base frequency (SD_CAPA [5:0] TCF bits). If the card does not respond within the specified number of cycles, a data timeout error occurs (SD_STAT[20] DTO bit). The SD_SYSCTL[19,16] DTO bit field is also used to check busy duration, to generate busy timeout for commands with busy response or for busy programming during a write command. Timeout on CRC status is generated if no CRC token is present after a block write. 0h (R/W) = TCF x 2^{13} 1h (R/W) = TCF x 2^{14} Eh (R/W) = TCF x 2^{27} Fh (R/W) = Reserved
15-6	CLKD	R/W	0h	Clock frequency select. These bits define the ratio between a reference clock frequency (system dependant) and the output clock frequency on the mmc_clk pin of either the memory card (MMC, SD, or SDIO). 0h (R/W) = Clock Ref bypass 1h (R/W) = Clock Ref bypass 2h (R/W) = Clock Ref / 2 3h (R/W) = Clock Ref / 3 3FFh (R/W) = Clock Ref / 1023
5-3	RESERVED	R	0h	
2	CEN	R/W	0h	Clock enable. This bit controls if the clock is provided to the card or not. 0h (R/W) = The clock is not provided to the card . Clock frequency can be changed . 1h (R/W) = The clock is provided to the card and can be automatically gated when SD_SYSCONFIG[0] AUTOIDLE bit is set to 1 (default value). The host driver shall wait to set this bit to 1 until the Internal clock is stable (SD_SYSCTL[1] ICS bit).
1	ICS	R	0h	Internal clock stable (status)This bit indicates either the internal clock is stable or not. 0h (R/W) = The internal clock is not stable. 1h (R/W) = The internal clock is stable after enabling the clock (SD_SYSCTL[0] ICE bit) or after changing the clock ratio (SD_SYSCTL[15:6] CLKD bits).
0	ICE	R/W	0h	Internal clock enable. This register controls the internal clock activity. In very low power state, the internal clock is stopped. Note: The activity of the debounce clock (used for wake-up events) and the interface clock (used for reads and writes to the module register map) are not affected by this register. 0h (R/W) = The internal clock is stopped (very low power state). 1h (R/W) = The internal clock oscillates and can be automatically gated when SD_SYSCONFIG[0] AUTOIDLE bit is set to 1 (default value).

17.5.1.19 SD_STAT Register (Offset = 230h) [reset = 0h]

SD_STAT is shown in Figure 17-55 and described in Table 17-38.

Return to Summary Table.

The interrupt status regroups all the status of the module internal events that can generate an interrupt. SD_STAT[31:16] = Error Interrupt Status. SD_STAT[15:0] = Normal Interrupt Status. The error bits are located in the upper 16 bits of the SD_STAT register. All bits are cleared by writing a 1 to them. Additionally, bits 15 and 8 serve as special error bits. These cannot be cleared by writing a 1 to them. Bit 15 (ERRI) is automatically cleared when the error causing to ERRI to be set is handled. (that is, when bits 31:16 are cleared, bit 15 will be automatically cleared). Bit 8 (CIRQ) is cleared by writing a 0 to SD_IE[8] (masking the interrupt) and servicing the interrupt.

Figure 17-55. SD_STAT Register

31	30	29	28	27	26	25	24
RESERVED		BADA	CERR	RESERVED		ADMAE	ACE
R-0h		R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	DEB	DCRC	DTO	CIE	CEB	CCRC	CTO
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
ERRI	RESERVED				BSR	OBI	CIRQ
R-0h	R-0h				R/W-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
CREM	CINS	BRR	BWR	DMA	BGE	TC	CC
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 17-38. SD_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	
29	BADA	R/W	0h	Bad access to data space. This bit is set automatically to indicate a bad access to buffer when not allowed: During a read access to the data register (SD_DATA) while buffer reads are not allowed (SD_PSTATE[11] BRE bit =0). During a write access to the data register (SD_DATA) while buffer writes are not allowed (SD_PSTATE[10] BWE bit=0). 0h (W) = Status bit unchanged 0h (R) = No interrupt 1h (W) = Status is cleared. 1h (R) = Bad access
28	CERR	R/W	0h	Card error. This bit is set automatically when there is at least one error in a response of type R1, R1b, R6, R5 or R5b. Only bits referenced as type E (error) in status field in the response can set a card status error. An error bit in the response is flagged only if corresponding bit in card status response error SD_CSRE in set. There is no card error detection for autoCMD12 command. The host driver shall read SD_RSP76 register to detect error bits in the command response. 0h (W) = Status bit unchanged 0h (R) = No error 1h (W) = Status is cleared. 1h (R) = Card error
27-26	RESERVED	R	0h	

Table 17-38. SD_STAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	ADMAE	R/W	0h	ADMA Error. This bit is set when the Host Controller detects errors during ADMA based data transfer. The state of the ADMA at an error occurrence is saved in the ADMA Error Status Register. In addition, the Host Controller generates this interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state. ADMA Error State in the ADMA Error Status indicates that an error occurs in ST_FDS state. The Host Driver may find that Valid bit is not set at the error descriptor. 0h (W) = Status bit unchanged 0h (R) = No interrupt 1h (W) = Status is cleared. 1h (R) = ADMA error
24	ACE	R/W	0h	Auto CMD12 error. This bit is set automatically when one of the bits in Auto CMD12 Error status register has changed from 0 to 1. 0h (W) = Status bit unchanged 0h (R) = No error 1h (W) = Status is cleared. 1h (R) = AutoCMD12 error
23	RESERVED	R	0h	
22	DEB	R/W	0h	Data End Bit error. This bit is set automatically when detecting a 0 at the end bit position of read data on mmc_dat line or at the end position of the CRC status in write mode. 0h (W) = Status bit unchanged 0h (R) = No error 1h (W) = Status is cleared. 1h (R) = Data end bit error
21	DCRC	R/W	0h	Data CRC Error. This bit is set automatically when there is a CRC16 error in the data phase response following a block read command or if there is a 3-bit CRC status different of a position "010" token during a block write command. 0h (W) = Status bit unchanged 0h (R) = No error 1h (W) = Status is cleared. 1h (R) = Data CRC error
20	DTO	R/W	0h	Data timeout error. This bit is set automatically according to the following conditions: Busy timeout for R1b, R5b response type. Busy timeout after write CRC status. Write CRC status timeout. Read data timeout. 0h (W) = Status bit unchanged 0h (R) = No error 1h (W) = Status is cleared. 1h (R) = Time out
19	CIE	R/W	0h	Command index error. This bit is set automatically when response index differs from corresponding command index previously emitted. It depends on the enable bit (SD_CMD[20] CICE). 0h (W) = Status bit unchanged 0h (R) = No error 1h (W) = Status is cleared. 1h (R) = Command index error

Table 17-38. SD_STAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	CEB	R/W	0h	Command end bit error. This bit is set automatically when detecting a 0 at the end bit position of a command response. 0h (W) = Status bit unchanged 0h (R) = No error 1h (W) = Status is cleared. 1h (R) = Command end bit error
17	CCRC	R/W	0h	Command CRC error. This bit is set automatically when there is a CRC7 error in the command response depending on the enable bit (SD_CMD[19] CCCE). 0h (W) = Status bit unchanged 0h (R) = No error 1h (W) = Status is cleared. 1h (R) = Command CRC error
16	CTO	R/W	0h	Command timeout error. This bit is set automatically when no response is received within 64 clock cycles from the end bit of the command. For commands that reply within 5 clock cycles - the timeout is still detected at 64 clock cycles. 0h (W) = Status bit unchanged 0h (R) = No error 1h (W) = Status is cleared. 1h (R) = Time Out
15	ERRI	R	0h	Error interrupt. If any of the bits in the Error Interrupt Status register (SD_STAT [31:16]) are set, then this bit is set to 1. Therefore the host driver can efficiently test for an error by checking this bit first. Writes to this bit are ignored. 0h (R/W) = No interrupt 1h (R/W) = Error interrupt event(s) occurred
14-11	RESERVED	R	0h	
10	BSR	R/W	0h	Boot Status Received Interrupt. This bit is set automatically when SD_CON[BOOT] is set 1 or 2 and a boot status is received on DAT[0] line. This interrupt is only useful for MMC card. 0h (W) = Status bit unchanged 0h (R) = No interrupt 1h (W) = Status is cleared. 1h (R) = Boot Status Received Interrupt occurred.
9	OBI	R	0h	Out-of-band interrupt (This interrupt is only useful for MMC card). This bit is set automatically when SD_CON[14] OBIE bit is set and an out-of-band interrupt occurs on OBI pin. The interrupt detection depends on polarity controlled by SD_CON[13] OBIP bit. The out-of-band interrupt signal is a system specific feature for future use, this signal is not required for existing specification implementation. 0h (W) = Status bit unchanged 0h (R) = No out-of-band interrupt 1h (W) = Status is cleared. 1h (R) = Interrupt out-of-band occurs

Table 17-38. SD_STAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	CIRQ	R	0h	Card interrupt. This bit is only used for SD and SDIO cards. In 1-bit mode, interrupt source is asynchronous (can be a source of asynchronous wake-up). In 4-bit mode, interrupt source is sampled during the interrupt cycle. In CE-ATA mode, interrupt source is detected when the card drives mmc_cmd line to zero during one cycle after data transmission end. All modes above are fully exclusive. The controller interrupt must be clear by setting SD_IE[8] CIRQ_ENABLE to 0, then the host driver must start the interrupt service with card (clearing card interrupt status) to remove card interrupt source. Otherwise the Controller interrupt will be reasserted as soon as SD_IE[8] CIRQ_ENABLE is set to 1. Writes to this bit are ignored. 0h (R/W) = No card interrupt 1h (R/W) = Generate card interrupt
7	CREM	R/W	0h	Card Removal. This bit is set automatically when SD_PSTATE[CINS] changes from 1 to 0. A clear of this bit doesn't affect Card inserted present state (SD_PSTATE[CINS]). 0h (W) = Status bit unchanged 0h (R) = Card State stable or debouncing 1h (R) = Card Removed 1h (W) = Status is cleared
6	CINS	R/W	0h	Card Insertion. This bit is set automatically when SD_PSTATE[CINS] changes from 0 to 1. A clear of this bit doesn't affect Card inserted present state (SD_PSTATE[CINS]). 0h (W) = Status bit unchanged 0h (R) = Card State stable or debouncing 1h (W) = Status is cleared. 1h (R) = Card inserted
5	BRR	R/W	0h	Buffer read ready. This bit is set automatically during a read operation to the card (see class 2 - block oriented read commands) when one block specified by the SD_BLK [10:0] BLEN bit field is completely written in the buffer. It indicates that the memory card has filled out the buffer and that the local host needs to empty the buffer by reading it. Note: If the DMA receive-mode is enabled, this bit is never set instead a DMA receive request to the main DMA controller of the system is generated. 0h (W) = Status bit unchanged 0h (R) = Not ready to read buffer 1h (W) = Status is cleared. 1h (R) = Ready to read buffer

Table 17-38. SD_STAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	BWR	R/W	0h	Buffer write ready. This bit is set automatically during a write operation to the card (see class 4 - block oriented write command) when the host can write a complete block as specified by SD_BLK [10:0] BLEN. It indicates that the memory card has emptied one block from the buffer and that the local host is able to write one block of data into the buffer. Note: If the DMA transmit mode is enabled, this bit is never set instead, a DMA transmit request to the main DMA controller of the system is generated. 0h (W) = Status bit unchanged 0h (R) = Not ready to write buffer 1h (W) = Status is cleared. 1h (R) = Ready to write buffer
3	DMA	R/W	0h	DMA Interrupt. This status is set when an interrupt is required in the ADMA instruction and after the data transfer completion. 0h (W) = Status bit unchanged 0h (R) = DMA Interrupt detected 1h (W) = Status is cleared. 1h (R) = No DMA Interrupt
2	BGE	R/W	0h	Block gap event. When a stop at block gap is requested (SD_HCTL[16] SBGR bit), this bit is automatically set when transaction is stopped at the block gap during a read or write operation. 0h (W) = Status bit unchanged 0h (R) = No block gap event 1h (R) = Transaction stopped at block gap 1h (W) = Status is cleared
1	TC	R/W	0h	Transfer completed. This bit is always set when a read/write transfer is completed or between two blocks when the transfer is stopped due to a stop at block gap request (SD_HCTL[16] SBGR bit). 0h (W) = Status bit unchanged 0h (R) = No transfer complete 1h (W) = Status is cleared 1h (R) = Data transfer complete
0	CC	R/W	0h	Command complete. This bit is set when a 1-to-0 transition occurs in the register command inhibit (SD_PSTATE[0] CMDI bit) 0h (W) = Status bit unchanged 0h (R) = No command complete 1h (W) = Status is cleared 1h (R) = Command complete

17.5.1.20 SD_IE Register (Offset = 234h) [reset = 0h]

SD_IE is shown in Figure 17-56 and described in Table 17-39.

Return to Summary Table.

This register allows to enable/disable the module to set status bits, on an event-by-event basis. SD_IE[31:16] = Error Interrupt Status Enable. SD_IE[15:0] = Normal Interrupt Status Enable.

Figure 17-56. SD_IE Register

31	30	29	28	27	26	25	24
RESERVED		BADA_ENABLE	CERR_ENABLE	RESERVED		ADMA_ENABLE	ACE_ENABLE
R-0h		R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	DEB_ENABLE	DCRC_ENABLE	DTO_ENABLE	CIE_ENABLE	CEB_ENABLE	CCRC_ENABLE	CTO_ENABLE
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
NULL	RESERVED				BSR_ENABLE	OBI_ENABLE	CIRQ_ENABLE
R-0h	R-0h				R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CREM_ENABLE	CINS_ENABLE	BRR_ENABLE	BWR_ENABLE	DMA_ENABLE	BGE_ENABLE	TC_ENABLE	CC_ENABLE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 17-39. SD_IE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	
29	BADA_ENABLE	R/W	0h	Bad access to data space interrupt enable 0h (R/W) = Masked 1h (R/W) = Enabled
28	CERR_ENABLE	R/W	0h	Card error interrupt enable 0h (R/W) = Masked 1h (R/W) = Enabled
27-26	RESERVED	R	0h	
25	ADMA_ENABLE	R/W	0h	ADMA error Interrupt Enable 0h (R/W) = Masked 1h (R/W) = Enabled
24	ACE_ENABLE	R/W	0h	Auto CMD12 error interrupt enable 0h (R/W) = Masked 1h (R/W) = Enabled
23	RESERVED	R	0h	
22	DEB_ENABLE	R/W	0h	Data end bit error interrupt enable 0h (R/W) = Masked 1h (R/W) = Enabled
21	DCRC_ENABLE	R/W	0h	Data CRC error interrupt enable 0h (R/W) = Masked 1h (R/W) = Enabled
20	DTO_ENABLE	R/W	0h	Data timeout error interrupt enable 0h (R/W) = Masked 1h (R/W) = Enabled

Table 17-39. SD_IE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	CIE_ENABLE	R/W	0h	Command index error interrupt enable 0h (R/W) = Masked 1h (R/W) = Enabled
18	CEB_ENABLE	R/W	0h	Command end bit error interrupt enable 0h (R/W) = Masked 1h (R/W) = Enabled
17	CCRC_ENABLE	R/W	0h	Command CRC error interrupt enable 0h (R/W) = Masked 1h (R/W) = Enabled
16	CTO_ENABLE	R/W	0h	Command timeout error interrupt enable 0h (R/W) = Masked 1h (R/W) = Enabled
15	NULL	R	0h	Fixed to 0. The host driver shall control error interrupts using the Error Interrupt Signal Enable register. Writes to this bit are ignored.
14-11	RESERVED	R	0h	
10	BSR_ENABLE	R/W	0h	Boot Status Interrupt Enable A write to this register when SD_CON[BOOT] is cleared to 0 is ignored. 0h (R/W) = Masked 1h (R/W) = Enabled
9	OBI_ENABLE	R/W	0h	Out-of-band interrupt enable A write to this register when SD_CON[14] OBIE is cleared to 0 is ignored. 0h (R/W) = Masked 1h (R/W) = Enabled
8	CIRQ_ENABLE	R/W	0h	Card interrupt enable. A clear of this bit also clears the corresponding status bit. During 1-bit mode, if the interrupt routine does not remove the source of a card interrupt in the SDIO card, the status bit is reasserted when this bit is set to 1. This bit must be set to 1 when entering in smart idle mode to enable system to identity wake-up event and to allow controller to clear internal wake-up source. 0h (R/W) = Masked 1h (R/W) = Enabled
7	CREM_ENABLE	R/W	0h	Card Removal interrupt Enable This bit must be set to 1 when entering in smart idle mode to enable system to identity wake-up event and to allow controller to clear internal wake-up source. 0h (R/W) = Masked 1h (R/W) = Enabled
6	CINS_ENABLE	R/W	0h	Card Insertion interrupt Enable This bit must be set to 1 when entering in smart idle mode to enable system to identity wake-up event and to allow controller to clear internal wake-up source. 0h (R/W) = Masked 1h (R/W) = Enabled
5	BRR_ENABLE	R/W	0h	Buffer read ready interrupt enable 0h (R/W) = Masked 1h (R/W) = Enabled
4	BWR_ENABLE	R/W	0h	Buffer write ready interrupt enable 0h (R/W) = Masked 1h (R/W) = Enabled
3	DMA_ENABLE	R/W	0h	DMA interrupt enable 0h (R/W) = Masked 1h (R/W) = Enable

Table 17-39. SD_IE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	BGE_ENABLE	R/W	0h	Block gap event interrupt enable 0h (R/W) = Masked 1h (R/W) = Enabled
1	TC_ENABLE	R/W	0h	Transfer completed interrupt enable 0h (R/W) = Masked 1h (R/W) = Enabled
0	CC_ENABLE	R/W	0h	Command completed interrupt enable 0h (R/W) = Masked 1h (R/W) = Enabled

17.5.1.21 SD_ISE Register (Offset = 238h) [reset = 0h]

SD_ISE is shown in Figure 17-57 and described in Table 17-40.

Return to Summary Table.

This register allows you to enable/disable the module to set status bits, on an event-by-event basis. SD_ISE[31:16] = Error Interrupt Signal Enable. SD_ISE[15:0] = Normal Interrupt Signal Enable.

Figure 17-57. SD_ISE Register

31	30	29	28	27	26	25	24
RESERVED		BADA_SIGEN	CERR_SIGEN	RESERVED		ADMA_SIGEN	ACE_SIGEN
R-0h		R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	DEB_SIGEN	DCRC_SIGEN	DTO_SIGEN	CIE_SIGEN	CEB_SIGEN	CCRC_SIGEN	CTO_SIGEN
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
NULL	RESERVED				BSR_SIGEN	OBI_SIGEN	CIRQ_SIGEN
R-0h	R-0h				R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CREM_SIGEN	CINS_SIGEN	BRR_SIGEN	BWR_SIGEN	DMA_SIGEN	BGE_SIGEN	TC_SIGEN	CC_SIGEN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 17-40. SD_ISE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	
29	BADA_SIGEN	R/W	0h	Bad access to data space interrupt enable 0h (R/W) = Masked 1h (R/W) = Enabled
28	CERR_SIGEN	R/W	0h	Card error interrupt signal status enable 0h (R/W) = Masked 1h (R/W) = Enabled
27-26	RESERVED	R	0h	
25	ADMA_SIGEN	R/W	0h	ADMA error signal status enable 0h (R/W) = Masked 1h (R/W) = Enabled
24	ACE_SIGEN	R/W	0h	Auto CMD12 error signal status enable 0h (R/W) = Masked 1h (R/W) = Enabled
23	RESERVED	R	0h	
22	DEB_SIGEN	R/W	0h	Data end bit error signal status enable 0h (R/W) = Masked 1h (R/W) = Enabled
21	DCRC_SIGEN	R/W	0h	Data CRC error signal status enable 0h (R/W) = Masked 1h (R/W) = Enabled
20	DTO_SIGEN	R/W	0h	Data timeout error signal status enable 0h (R/W) = Masked. The host controller provides the clock to the card until the card sends the data or the transfer is aborted. 1h (R/W) = Enabled
19	CIE_SIGEN	R/W	0h	Command index error signal status enable 0h (R/W) = Masked 1h (R/W) = Enabled

Table 17-40. SD_ISE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	CEB_SIGEN	R/W	0h	Command end bit error signal status enable 0h (R/W) = Masked 1h (R/W) = Enabled
17	CCRC_SIGEN	R/W	0h	Command CRC error signal status enable 0h (R/W) = Masked 1h (R/W) = Enabled
16	CTO_SIGEN	R/W	0h	Command timeout error signal status enable 0h (R/W) = Masked 1h (R/W) = Enabled
15	NULL	R	0h	Fixed to 0. The host driver shall control error interrupts using the error interrupt signal enable register. Writes to this bit are ignored.
14-11	RESERVED	R	0h	
10	BSR_SIGEN	R/W	0h	Boot Status signal status enable. A write to this register when SD_CON[BOOT] is cleared to 0 is ignored 0h (R/W) = Masked 1h (R/W) = Enabled
9	OBI_SIGEN	R/W	0h	Out-of-band interrupt signal status enable. A write to this register when SD_CON[14] OBIE is cleared to 0 is ignored. 0h (R/W) = Masked 1h (R/W) = Enabled
8	CIRQ_SIGEN	R/W	0h	Card interrupt signal status enable. A clear of this bit also clears the corresponding status bit. During 1-bit mode, if the interrupt routine does not remove the source of a card interrupt in the SDIO card, the status bit is reasserted when this bit is set to 1. This bit must be set to 1 when entering in smart idle mode to enable system to identity wake-up event and to allow controller to clear internal wake-up source. 0h (R/W) = Masked 1h (R/W) = Enabled
7	CREM_SIGEN	R/W	0h	Card Removal signal status enable This bit must be set to 1 when entering in smart idle mode to enable system to identity wake-up event and to allow controller to clear internal wake-up source. 0h (R/W) = Masked 1h (R/W) = Enabled
6	CINS_SIGEN	R/W	0h	Card Insertion signal status enable. This bit must be set to 1 when entering in smart idle mode to enable system to identity wake-up event and to allow controller to clear internal wake-up source. 0h (R/W) = Masked 1h (R/W) = Enabled
5	BRR_SIGEN	R/W	0h	Buffer read ready signal status enable 0h (R/W) = Masked 1h (R/W) = Enabled
4	BWR_SIGEN	R/W	0h	Buffer write ready signal status enable 0h (R/W) = Masked 1h (R/W) = Enabled
3	DMA_SIGEN	R/W	0h	DMA signal status enable 0h (R/W) = Masked 1h (R/W) = Enabled

Table 17-40. SD_ISE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	BGE_SIGEN	R/W	0h	Block gap event signal status enable 0h (R/W) = Masked 1h (R/W) = Enabled
1	TC_SIGEN	R/W	0h	Transfer completed signal status enable 0h (R/W) = Masked 1h (R/W) = Enabled
0	CC_SIGEN	R/W	0h	Command completed signal status enable 0h (R/W) = Masked 1h (R/W) = Enabled

17.5.1.22 SD_AC12 Register (Offset = 23Ch) [reset = 0h]

SD_AC12 is shown in Figure 17-58 and described in Table 17-41.

Return to Summary Table.

The host driver may determine which of the errors cases related to Auto CMD12 has occurred by checking this SD_AC12 register when an auto CMD12 error interrupt occurs. This register is valid only when auto CMD12 is enabled (SD_CMD[2] ACEN bit) and auto CMD12Error (SD_STAT[24] ACE bit) is set to 1. These bits are automatically reset when starting a new adtc command with data.

Figure 17-58. SD_AC12 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
CNI	RESERVED		ACIE	ACEB	ACCE	ACTO	ACNE
R-0h	R-0h		R-0h	R-0h	R-0h	R-0h	R-0h

Table 17-41. SD_AC12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	CNI	R	0h	Command not issue by auto CMD12 error. If this bit is set to 1, it means that pending command is not executed due to auto CMD12 error ACEB, ACCE, ACTO, or ACNE. 0h (R/W) = Not error 1h (R/W) = Command not issued
6-5	RESERVED	R	0h	
4	ACIE	R	0h	Auto CMD12 index error. This bit is a set to 1 when response index differs from corresponding command auto CMD12 index previously emitted. This bit depends on the command index check enable (SD_CMD[20] CICE bit). 0h (R/W) = No error 1h (R/W) = Auto CMD12 index error
3	ACEB	R	0h	Auto CMD12 end bit error. This bit is set to 1 when detecting a 0 at the end bit position of auto CMD12 command response. 0h (R/W) = No error 1h (R/W) = AutoCMD12 end bit error
2	ACCE	R	0h	Auto CMD12 CRC error. This bit is automatically set to 1 when a CRC7 error is detected in the auto CMD12 command response depending on the enable in the SD_CMD[19] CCCE bit. 0h (R/W) = No error 1h (R/W) = Auto CMD12 CRC error
1	ACTO	R	0h	Auto CMD12 timeout error. This bit is set to 1 if no response is received within 64 clock cycles from the end bit of the auto CMD12 command. 0h (R/W) = No error 1h (R/W) = Auto CMD12 time out

Table 17-41. SD_AC12 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	ACNE	R	0h	Auto CMD12 not executed. This bit is set to 1 if multiple block data transfer command has started and if an error occurs in command before auto CMD12 starts. 0h (R/W) = Auto CMD12 executed 1h (R/W) = Auto CMD12 not executed

17.5.1.23 SD_CAPA Register (Offset = 240h) [reset = 0h]

SD_CAPA is shown in Figure 17-59 and described in Table 17-42.

Return to Summary Table.

This register lists the capabilities of the MMC/SD/SDIO host controller.

Figure 17-59. SD_CAPA Register

31	30	29	28	27	26	25	24
RESERVED			BUS_64BIT	RESERVED	VS18	VS30	VS33
R-0h			R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SRS	DS	HSS	RESERVED	AD2S	RESERVED	MBL	
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	

15	14	13	12	11	10	9	8
RESERVED		BCF					
R-0h		R-0h					

7	6	5	4	3	2	1	0
TCU	RESERVED	TCF					
R-0h	R-0h	R-0h					

Table 17-42. SD_CAPA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	BUS_64BIT	R/W	0h	64 Bit System Bus Support. Setting 1 to this bit indicates that the Host Controller supports 64-bit address descriptor mode and is connected to 64-bit address system bus. 0h (R/W) = 32-bit System bus address 1h (R/W) = 64-bit System bus address
27	RESERVED	R	0h	
26	VS18	R/W	0h	Voltage support 1.8 V. Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initialization. This register is only reinitialized by a hard reset (via mmc_RESET signal). 0h (W) = 1.8 V not supported 0h (R) = 1.8 V not supported 1h (R) = 1.8 V supported 1h (W) = 1.8 V supported
25	VS30	R/W	0h	Voltage support 3.0V. Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initialization. This register is only reinitialized by a hard reset (via mmc_RESET signal). 0h (W) = 3.0 V not supported 0h (R) = 3.0 V not supported 1h (R) = 3.0 V supported 1h (W) = 3.0 V supported

Table 17-42. SD_CAPA Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	VS33	R/W	0h	Voltage support 3.3V. Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initialization. This register is only reinitialized by a hard reset (via mmc_RESET signal). 0h (W) = 3.3 V not supported 0h (R) = 3.3 V not supported 1h (R) = 3.3 V supported 1h (W) = 3.3 V supported
23	SRS	R	0h	Suspend/resume support (SDIO cards only). This bit indicates whether the host controller supports Suspend/Resume functionality. 0h (R/W) = The Host controller does not suspend/resume functionality. 1h (R/W) = The Host controller supports suspend/resume functionality.
22	DS	R	0h	DMA support. This bit indicates that the Host controller is able to use DMA to transfer data between system memory and the Host controller directly. 0h (R/W) = DMA not supported 1h (R/W) = DMA supported
21	HSS	R	0h	High-speed support. This bit indicates that the host controller supports high speed operations and can supply an up-to-52 MHz clock to the card. 0h (R/W) = DMA not supported 1h (R/W) = DMA supported
20	RESERVED	R	0h	
19	AD2S	R	0h	This bit indicates whether the Host Controller is capable of using ADMA2. It depends on setting of generic parameter MADMA_EN. 0h (R/W) = ADMA2 supported 1h (R/W) = ADMA2 not supported
18	RESERVED	R	0h	
17-16	MBL	R	0h	Maximum block length. This value indicates the maximum block size that the host driver can read and write to the buffer in the host controller. The host controller supports 512 bytes and 1024 bytes block transfers. 0h (R/W) = 512 bytes 1h (R/W) = 1024 bytes 2h (R/W) = 2048 bytes
15-14	RESERVED	R	0h	
13-8	BCF	R	0h	Base clock frequency for clock provided to the card. ARRAY(0x1bfe1b0)
7	TCU	R	0h	Timeout clock unit. This bit shows the unit of base clock frequency used to detect Data Timeout Error (SD_STAT[20] DTO bit). 0h (R/W) = kHz 1h (R/W) = MHz
6	RESERVED	R	0h	
5-0	TCF	R	0h	Timeout clock frequency. The timeout clock frequency is used to detect Data Timeout Error (SD_STAT[20] DTO bit). 0h (R/W) = The timeout clock frequency depends on the frequency of the clock provided to the card. The value of the timeout clock frequency is not available in this register.

17.5.1.24 SD_CUR_CAPA Register (Offset = 248h) [reset = 0h]

SD_CUR_CAPA is shown in Figure 17-60 and described in Table 17-43.

Return to Summary Table.

This register indicates the maximum current capability for each voltage. The value is meaningful if the voltage support is set in the capabilities register (SD_CAPA). Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initialization. This register is only reinitialized by a hard reset (via mmc_RESET signal).

Figure 17-60. SD_CUR_CAPA Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">CUR_1V8</td><td colspan="8">CUR_3V0</td><td colspan="8">CUR_3V3</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="8">R/W-0h</td><td colspan="8">R/W-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 17-43. SD_CUR_CAPA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-16	CUR_1V8	R/W	0h	Maximum current for 1.8 V 0h (R/W) = The maximum current capability for this voltage is not available. Feature not implemented.
15-8	CUR_3V0	R/W	0h	Maximum current for 3.0 V 0h (R/W) = The maximum current capability for this voltage is not available. Feature not implemented.
7-0	CUR_3V3	R/W	0h	Maximum current for 3.3 V 0h (R/W) = The maximum current capability for this voltage is not available. Feature not implemented.

17.5.1.25 SD_FE Register (Offset = 250h) [reset = 0h]

SD_FE is shown in Figure 17-61 and described in Table 17-44.

Return to Summary Table.

The Force Event register is not a physically implemented register. Rather, it is an address at which the Error Interrupt Status register can be written. The effect of a write to this address will be reflected in the Error Interrupt Status Register, if corresponding bit of the Error Interrupt Status Enable Register is set.

Figure 17-61. SD_FE Register

31	30	29	28	27	26	25	24
RESERVED		FE_BADA	FE_CERR	RESERVED		FE_ADMAE	FE_ACE
R-0h		W-0h	W-0h	R-0h		W-0h	W-0h
23	22	21	20	19	18	17	16
RESERVED	FE_DEB	FE_DCRC	FE_DTO	FE_CIE	FE_CEB	FE_CCRC	FE_CTO
R-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
FE_CNI	RESERVED		FE_ACIE	FE_ACEB	FE_ACCE	FE_ACTO	FE_ACNE
W-0h	R-0h		W-0h	W-0h	W-0h	W-0h	W-0h

Table 17-44. SD_FE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	
29	FE_BADA	W	0h	Force Event Bad access to data space. 0h (R/W) = No effect; no interrupt 1h (R/W) = Interrupt forced.
28	FE_CERR	W	0h	Force Event Card error 0h (R/W) = No effect; no interrupt 1h (R/W) = Interrupt forced.
27-26	RESERVED	R	0h	
25	FE_ADMAE	W	0h	Force Event ADMA error 0h (R/W) = No effect; no interrupt 1h (R/W) = Interrupt forced.
24	FE_ACE	W	0h	Force Event Auto CMD12 error. 0h (R/W) = No effect; no interrupt 1h (R/W) = Interrupt forced.
23	RESERVED	R	0h	
22	FE_DEB	W	0h	Force Event Data End Bit error. 0h (R/W) = No effect; no interrupt 1h (R/W) = Interrupt forced.
21	FE_DCRC	W	0h	Force Event Data CRC error 0h (R/W) = No effect; no interrupt 1h (R/W) = Interrupt forced.
20	FE_DTO	W	0h	Force Event Data timeout error 0h (R/W) = No effect; no interrupt 1h (R/W) = Interrupt forced.
19	FE_CIE	W	0h	Force Event Command index error 0h (R/W) = No effect; no interrupt 1h (R/W) = Interrupt forced.

Table 17-44. SD_FE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	FE_CEB	W	0h	Force Event Command end bit error 0h (R/W) = No effect; no interrupt 1h (R/W) = Interrupt forced.
17	FE_CCRC	W	0h	Force Event Comemand CRC error 0h (R/W) = No effect; no interrupt 1h (R/W) = Interrupt forced.
16	FE_CTO	W	0h	Force Event Command Timeout error 0h (R/W) = No effect; no interrupt 1h (R/W) = Interrupt forced.
15-8	RESERVED	R	0h	
7	FE_CNI	W	0h	Force Event Command not issue by Auto CMD12 error 0h (R/W) = No effect; no interrupt 1h (R/W) = Interrupt forced.
6-5	RESERVED	R	0h	
4	FE_ACIE	W	0h	Force Event Auto CMD12 index error 0h (R/W) = No effect; no interrupt 1h (R/W) = Interrupt forced.
3	FE_ACEB	W	0h	Force Event Auto CMD12 end bit error 0h (R/W) = No effect; no interrupt 1h (R/W) = Interrupt forced.
2	FE_ACCE	W	0h	Force Event Auto CMD12 CRC error 0h (R/W) = No effect; no interrupt 1h (R/W) = Interrupt forced.
1	FE_ACTO	W	0h	Force Event Auto CMD12 timeout error 0h (R/W) = No effect; no interrupt 1h (R/W) = Interrupt forced.
0	FE_ACNE	W	0h	Force Event Auto CMD12 not executed. 0h (R/W) = No effect; no interrupt. 1h (R/W) = Interrupt forced.

17.5.1.26 SD_ADMAES Register (Offset = 254h) [reset = 0h]

SD_ADMAES is shown in Figure 17-62 and described in Table 17-45.

Return to Summary Table.

When an ADMA Error Interrupt has occurred, the ADMA Error States field in this register holds the ADMA state and the ADMA System Address Register holds the address around the error descriptor. For recovering the error, the Host Driver requires the ADMA state to identify the error descriptor address as follows: ST_STOP: Previous location set in the ADMA System Address register is the error descriptor address. ST_FDS: Current location set in the ADMA System Address register is the error descriptor address. ST_CADR: This sate is never set because do not generate ADMA error in this state. ST_TFR: Previous location set in the ADMA System Address register is the error descriptor address. In the case of a write operation, the Host Driver should use ACMD22 to get the number of written block rather than using this information, since unwritten data may exist in the Host Controller. The Host Controller generates the ADMA Error Interrupt when it detects invalid descriptor data (Valid = 0) at the ST_FDS state. In this case, ADMA Error State indicates that an error occurs at ST_FDS state. The Host Driver may find that the Valid bit is not set in the error descriptor.

Figure 17-62. SD_ADMAES Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													LME	AES	
R-0h													W-0h	R/W-0h	

Table 17-45. SD_ADMAES Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	LME	W	0h	ADMA Length Mismatch Error: While Block Count Enable is being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length. Total data length cannot be divided by the block length. 0h (R/W) = No error 1h (R/W) = Error
1-0	AES	R/W	0h	ADMA Error State. This field indicates the state of ADMA when an error occurred during an ADMA data transfer. This field never indicates "10" because ADMA never stops in this state. 0h (R) = ST_STOP (Stop DMA). Contents of the SYS_SDR register 1h (W) = ST_STOP (Stop DMA). Points to the error descriptor. 2h (R) = Never set this state. (Not used) 3h (W) = ST_TFR (Transfer Data). Points to the 'next' of the error descriptor.

17.5.1.27 SD_ADMASAL Register (Offset = 258h) [reset = 0h]

SD_ADMASAL is shown in Figure 17-63 and described in Table 17-46.

Return to Summary Table.

This register holds the byte address of the executing command of the Descriptor table. The 32-bit Address Descriptor uses the lower 32 bits of this register. At the start of ADMA, the Host Driver shall set the start address of the Descriptor table.

Figure 17-63. SD_ADMASAL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">ADMA_A32B</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 17-46. SD_ADMASAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADMA_A32B	R/W	0h	The ADMA increments this register address, which points to the next line, whenever fetching a Descriptor line. When the ADMA Error Interrupt is generated, this register holds the valid Descriptor address depending on the ADMA state. The Host Driver shall program the Descriptor Table on a 32-bit boundary and set the 32-bit boundary address to this register. ADMA2 ignores the lower 2 bits of this register and assumes it to be 00b.

17.5.1.28 SD_ADMASAH Register (Offset = 25Ch) [reset = 0h]

SD_ADMASAH is shown in Figure 17-64 and described in Table 17-47.

Return to Summary Table.

Figure 17-64. SD_ADMASAH Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">ADMA_A32B</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 17-47. SD_ADMASAH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADMA_A32B	R/W	0h	ADMA_A32B.

17.5.1.29 SD_REV Register (Offset = 2FCh) [reset = 0h]

SD_REV is shown in Figure 17-65 and described in Table 17-48.

Return to Summary Table.

This register contains the hard coded RTL vendor revision number, the version number of SD specification compliancy and a slot status bit. SD_REV[31:16] = Host Controller Version. SD_REV[15:0] = Slot Interrupt Status.

Figure 17-65. SD_REV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VREV								SREV							
R-0h								R-0h							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															SIS
R-0h															R-0h

Table 17-48. SD_REV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	VREV	R	0h	Vendor Version Number. Bits [7:4] is the major revision, bits [3:0] is the minor revision. Examples: 10h for 1.0 21h for 2.1
23-16	SREV	R	0h	Specification Version Number. This status indicates the Standard SD Host Controller Specification Version. The upper and lower 4-bits indicate the version. 0h (R/W) = SD Host Specification Version 1.0
15-1	RESERVED	R	0h	
0	SIS	R	0h	Slot Interrupt Status. This status bit indicates the inverted state of interrupt signal for the module. By a power on reset or by setting a software reset for all (SD_SYSCTL[24] SRA), the interrupt signal shall be deasserted and this status shall read 0. 0h = Interrupt signal deasserted. 1h = Interrupt signal asserted.

Interprocessor Communication

This chapter describes the interprocessor communication of the device.

18.1 Mailbox

18.1.1 Introduction

18.1.1.1 Features

Global features of the Mailbox module are:

- OCP slave interface (L4) supports:
  - 32-bit data bus width
  - 8/16/32 bit access supported
  - 9-bit address bus width
  - Burst not supported
- 8 mailbox sub-modules
- Each mailbox sub module allows 1-way communication between 2 initiators
- Flexible mailbox/initiators assignment scheme
- 4 messages per mailbox sub-module
- 32-bit message width
- Support of 16/32-bit addressing scheme
- Non-intrusive emulation
- 4 interrupts (one per user: 1 to MPU Subsystem, 2 to PRU-ICSS, and 1 to Wakeup Processor)

18.1.1.2 Unsupported Features

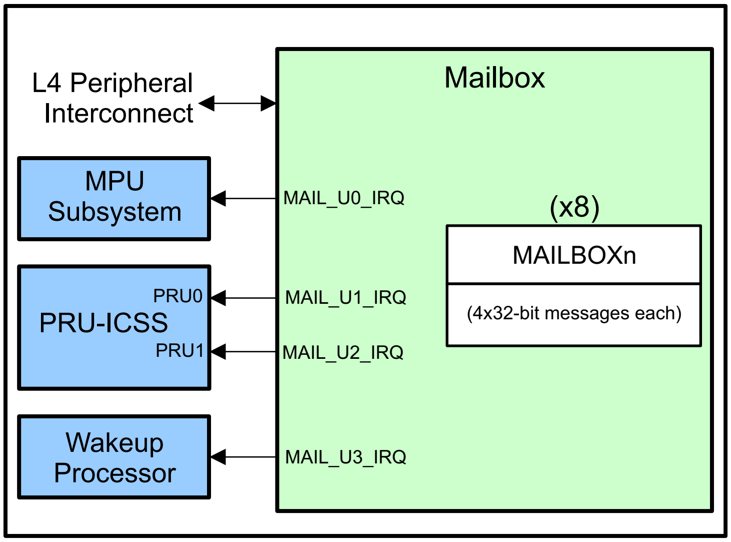
There are no unsupported features for Mailbox on this device.

18.1.2 Integration

This device contains a single instantiation of the Mailbox module at the system level. The mailbox function is made of eight sub-module mailboxes each supporting a 1-way communication between two initiators. The communication protocol from the sender to the receiver is implemented with mailbox registers using interrupts. The sender sends information to the receiver by writing to the mailbox. Interrupt signaling is used to notify the receiver a message has been queued or the sender for overflow situation.

The eight mailboxes are enough to handle communications between the MPU Subsystem, PRU-ICSS PRUs, and Wakeup Processor. Note that because the Wakeup Processor has access to only L4_Wakeup peripherals it does not have access to the Mailbox registers. A mailbox interrupt can still be sent to the Wakeup Processor to trigger message notification. The actual message payload must be placed in either the internal memory of the Wakeup Processor or in the Control Module Interprocessor Message registers (IPC_MSG_REG{0-15}).

Figure 18-1. Mailbox Integration



18.1.2.1 Mailbox Connectivity Attributes

The general connectivity for the Mailbox is shown in Table 18-1.

Table 18-1. Mailbox Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	L4PER_L4LS_GCLK
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	4 Interrupts mail_u0 (MBINT0) – to MPU Subsystem mail_u1 – to PRU-ICSS (PRU0) mail_u2 – to PRU-ICSS (PRU1) mail_u3 – to Wakeup Processor
DMA Requests	None
Physical Address	L4 Peripheral slave port

18.1.2.2 Mailbox Clock and Reset Management

The mailbox function operates from the L4 interface clock.

Table 18-2. Mailbox Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
Functional / Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk From PRCM

18.1.2.3 Mailbox Pin List

The Mailbox module does not include any external interface pins.

18.1.3 Functional Description

This device has the following mailbox instances:

- System mailbox

Table 18-3 shows Mailbox Implementation in this device, where u is the user number and m is the mailbox number.

Table 18-3. Mailbox Implementation

Mailbox Type	User Number(u)	Mailbox Number(m)	Messages per Mailbox
System mailbox	0 to 3	0 to 7	4

The mailbox module provides a means of communication through message queues among the users (depending on the mailbox module instance). The individual mailbox modules (8 for the system mailbox instance), or FIFOs, can associate (or de-associate) with any of the processors using the MAILBOX_IRQENABLE_SET_u (or MAILBOX_IRQENABLE_CLR_u) register.

The system mailbox module includes the following user subsystems:

- User 0: MPU Subsystem (u = 0)
- User 1: PRU_ICSS PRU0 (u = 1)
- User 2: PRU_ICSS PRU1 (u = 2)
- User 3: Wakeup Processor (u = 3)

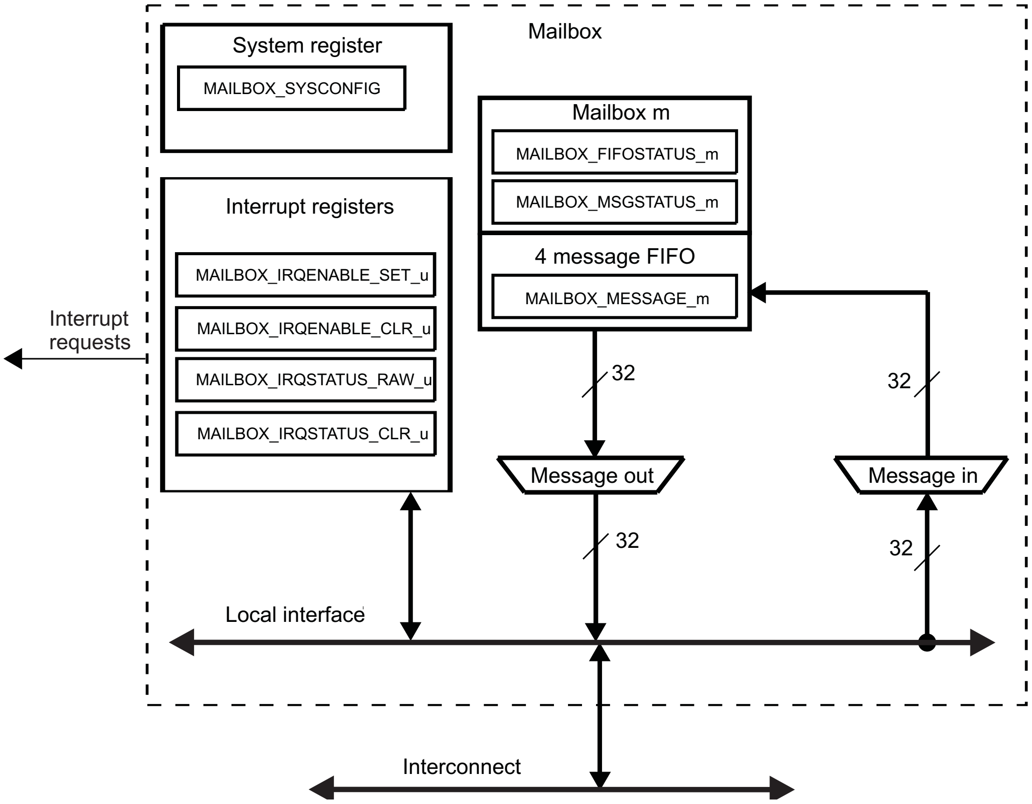
Each user has a dedicated interrupt signal from the corresponding mailbox module instance and dedicated interrupt enabling and status registers. Each MAILBOX_IRQSTATUS_RAW_u/MAILBOX_IRQSTATUS_CLR_u interrupt status register corresponds to a particular user.

For the system mailbox instance, a user can query its interrupt status register through the L4_STANDARD interconnect.

18.1.3.1 Mailbox Block Diagram

Figure 18-2 shows the mailbox block diagram.

Figure 18-2. Mailbox Block Diagram



18.1.3.2 Software Reset

The mailbox module supports a software reset through the MAILBOX_SYSCONFIG[0].SOFTRESET bit. Setting this bit to 1 enables an active software reset that is functionally equivalent to a hardware reset. Reading the MAILBOX_SYSCONFIG[0] SOFTRESET bit gives the status of the software reset:

- Read 1: the software reset is on-going
- Read 0: the software reset is complete

The software must ensure that the software reset completes before doing mailbox operations.

18.1.3.3 Power Management

Table 18-4 describes power-management features available for the mailbox module.

Table 18-4. Local Power Management Features

Feature	Registers	Description
Clock autogating	NA	Feature not available
Slave idle modes	MAILBOX_SYSCONFIG[3:2].SIDLEMODE	Force-idle, no-idle and smart-idle modes are available
Clock activity	NA	Feature not available
Master standby modes	NA	Feature not available
Global wake-up enable	NA	Feature not available
Wake-up sources enable	NA	Feature not available

The mailbox module can be configured using the MAILBOX_SYSCONFIG[3:2] SIDLEMODE bit field to one of the following acknowledgment modes:

- Force-idle mode (SIDLEMODE = 0x0): The mailbox module immediately enters the idle state on receiving a low-power-mode request from the PRCM module. In this mode, the software must ensure that there are no asserted output interrupts before requesting this mode to go into the idle state.
- No-idle mode (SIDLEMODE = 0x1): The mailbox module never enters the idle state.
- Smart-idle mode (SIDLEMODE = 0x2): After receiving a low-power-mode request from the PRCM module, the mailbox module enters the idle state only after all asserted output interrupts are acknowledged.

18.1.3.4 Interrupt Requests

An interrupt request allows the user of the mailbox to be notified when a message is received or when the message queue is not full. There is one interrupt per user. Table 18-5 lists the event flags, and their mask, that can cause module interrupts.

Table 18-5. Interrupt Events

Non-Maskable Event Flag[1]	Maskable Event Flag	Event Mask Bit	Event Unmask Bit	Description
MAILBOX_IRQSTATUS_RAW_u[0+m*2].NEWMSGSTATUSUUMBm	MAILBOX_IRQSTATUS_CLR_u[0+m*2].NEWMSGSTATUSUUMBm	MAILBOX_IRQENABLE_CLR_u[0+m*2].		
MAILBOX_IRQSTATUS_RAW_u[0+m*2].NEWMSGSTATUSUUMBm	MAILBOX_IRQSTATUS_CLR_u[0+m*2].NEWMSGSTATUSUUMBm	MAILBOX_IRQENABLE_CLR_u[0+m*2].NEWMSGSTATUSUUMBm	MAILBOX_IRQENABLE_SET_u[0+m*2].NEWMSGSTATUSUUMBm	Mailbox m receives a new message
MAILBOX_IRQSTATUS_RAW_u[1+m*2].NOTFULLSTATUSUMBm	MAILBOX_IRQSTATUS_CLR_u[1+m*2].NOTFULLSTATUSUMBm	MAILBOX_IRQENABLE_CLR_u[1+m*2].NOTFULLSTATUSUMBm	MAILBOX_IRQENABLE_SET_u[1+m*2].NOTFULLSTATUSUMBm	Mailbox m message queue is not full

(1) MAILBOX.MAILBOX_IRQSTATUS_RAW_u register is mostly used for debug purposes.

CAUTION

Once an event generating the interrupt request has been processed by the software, it must be cleared by writing a logical 1 in the corresponding bit of the MAILBOX_IRQSTATUS_CLR_u register. Writing a logical 1 in a bit of the MAILBOX_IRQSTATUS_CLR_u register will also clear to 0 the corresponding bit in the appropriate MAILBOX_IRQSTATUS_RAW_u register.

An event can generate an interrupt request when a logical 1 is written to the corresponding unmask bit in the MAILBOX_IRQENABLE_SET_u register. Events are reported in the appropriate MAILBOX_IRQSTATUS_CLR_u and MAILBOX_IRQSTATUS_RAW_u registers.

An event stops generating interrupt requests when a logical 1 is written to the corresponding mask bit in the MAILBOX_IRQENABLE_CLR_u register. Events are only reported in the appropriate MAILBOX_IRQSTATUS_RAW_u register.

In case of the MAILBOX_IRQSTATUS_RAW_u register, the event is reported in the corresponding bit even if the interrupt request generation is disabled for this event.

18.1.3.5 Assignment

18.1.3.5.1 Description

To assign a receiver to a mailbox, set the new message interrupt enable bit corresponding to the desired mailbox in the MAILBOX_IRQENABLE_SET_u register. The receiver reads the MAILBOX_MESSAGE_m register to retrieve a message from the mailbox.

An alternate method for the receiver that does not use the interrupts is to poll the MAILBOX_FIFOSTATUS_m and/or MAILBOX_MSGSTATUS_m registers to know when to send or retrieve a message to or from the mailbox. This method does not require assigning a receiver to a mailbox. Because this method does not include the explicit assignment of the mailbox, the software must avoid having multiple receivers use the same mailbox, which can result in incoherency.

To assign a sender to a mailbox, set the queue-not-full interrupt enable bit of the desired mailbox in the MAILBOX_IRQENABLE_SET_u register, where u is the number of the sending user. However, direct allocation of a mailbox to a sender is not recommended because it can cause the sending processor to be constantly interrupted.

It is recommended that register polling be used to:

- Check the status of either the MAILBOX_FIFOSTATUS_m or MAILBOX_MSGSTATUS_m registers
- Write the message to the corresponding MAILBOX_MESSAGE_m register, if space is available

The sender might use the queue-not-full interrupt when the initial mailbox status check indicates the mailbox is full. In this case, the sender can enable the queue-not-full interrupt for its mailbox in the appropriate MAILBOX_IRQENABLE_SET_u register. This allows the sender to be notified by interrupt only when a FIFO queue has at least one available entry.

Reading the MAILBOX_IRQSTATUS_CLR_u register determines the status of the new message and the queue-not-full interrupts for a particular user. Writing 1 to the corresponding bit in the MAILBOX_IRQSTATUS_CLR_u register acknowledges, and subsequently clears, an interrupt.

CAUTION

Assigning multiple senders or multiple receivers to the same mailbox is not recommended.

18.1.3.6 Sending and Receiving Messages

18.1.3.6.1 Description

When a 32-bit message is written to the MAILBOX_MESSAGE_m register, the message is appended into the FIFO queue. This queue holds four messages. If the queue is full, the message is discarded. Queue overflow can be avoided by first reading the MAILBOX_FIFOSTATUS_m register to check that the mailbox message queue is not full before writing a new message to it. Reading the MAILBOX_MESSAGE_m register returns the message at the beginning of the FIFO queue and removes it from the queue. If the FIFO queue is empty when the MAILBOX_MESSAGE_m register is read, the value 0 is returned. The new message interrupt is asserted when at least one message is in the mailbox message FIFO queue. To determine the number of messages in the mailbox message FIFO queue, read the MAILBOX_MSGSTATUS_m register.

18.1.3.7 16-Bit Register Access

18.1.3.7.1 Description

So that 16-bit processors can access the mailbox module, the module allows 16-bit register read and write access, with restrictions for the MAILBOX_MESSAGE_m registers. The 16-bit half-words are organized in little endian fashion; that is, the least-significant 16 bits are at the low address and the most-significant 16 bits are at the high address (low address + 0x02). All mailbox module registers can be read or written to directly using individual 16-bit accesses with no restriction on interleaving, except the MAILBOX_MESSAGE_m registers, which must always be accessed by either single 32-bit accesses or two consecutive 16-bit accesses.

CAUTION

When using 16-bit accesses to the MAILBOX_MESSAGE_m registers, the order of access must be the least-significant half-word first (low address) and the most-significant half-word last (high address). This requirement is because of the update operation by the message FIFO of the MAILBOX_MSGSTATUS_m registers. The update of the FIFO queue contents and the associated status registers and possible interrupt generation occurs only when the most-significant 16 bits of a MAILBOX_MESSAGE_m are accessed.

18.1.4 Programming Guide

18.1.4.1 Low-level Programming Models

This section covers the low-level hardware programming sequences for configuration and usage of the mailbox module.

18.1.4.1.1 Global Initialization

18.1.4.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the mailbox module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration of the mailbox.

See Section 18.1.2 for further information.

Table 18-6. Global Initialization of Surrounding Modules for System Mailbox

Surrounding Modules	Comments
PRCM	Mailbox functional/interface clock must be enabled.
Interrupt Controllers	MPU interrupt controller must be configured to enable the interrupt request generation to the MPU.

18.1.4.1.1.2 Mailbox Global Initialization

18.1.4.1.1.2.1 Main Sequence - Mailbox Global Initialization

This procedure initializes the mailbox module after a power-on or software reset.

Table 18-7. Mailbox Global Initialization

	Register/Bitfield/Programming Model	Value
Perform a software reset	MAILBOX_SYSCONFIG[0].SOFTRESET	1
Wait until reset is complete	MAILBOX_SYSCONFIG[0].SOFTRESET	0
Set idle mode configuration	MAILBOX_SYSCONFIG[3:2].SIDLEMODE	0x-

18.1.4.1.2 Operational Modes Configuration

18.1.4.1.2.1 Main Sequence - Sending a Message (Polling Method)

Table 18-8. Sending a Message (Polling Method)

Step	Register/Bitfield/Programming Model	Value
IF : Is FIFO full ?	MAILBOX_FIFOSTATUS_m[0].FIFOFULLMB	=1h
Wait until at least one message slot is available	MAILBOX_FIFOSTATUS_m[0].FIFOFULLMB	=0h
ELSE		
Write message	MAILBOX_MESSAGE_m[31:0].MESSAGEVALUEMBM	----h
ENDIF		

18.1.4.1.2.2 Main Sequence - Sending a Message (Interrupt Method)

Table 18-9. Sending a Message (Interrupt Method)

Step	Register/Bitfield/Programming Model	Value
IF : Is FIFO full ?	MAILBOX_FIFOSTATUS_m[0].FIFOFULLMB	=1h
Enable interrupt event	MAILBOX_IRQENABLE_SET_u[1+ m*2]	1h
User(processor) can perform anothr task until interrupt occurs		
ELSE		
Write message	MAILBOX_MESSAGE_m[31:0].MESSAGEVALUEMBM	----h
ENDIF		

18.1.4.1.2.3 Main Sequence - Receiving a Message (Polling Method)

Table 18-10. Receiving a Message (Polling Method)

Step	Register/Bitfield/Programming Model	Value
IF : Number of messages is not equal to 0	MAILBOX_MSGSTATUS_m[2:0].NBOFMSGMB	!=0h
Read message	MAILBOX_MESSAGE_m[31:0].MESSAGEVALUEMBM	----h
ENDIF		

18.1.4.1.2.4 Main Sequence - Receiving a Message (Interrupt Method)

Table 18-11. Receiving a Message (Interrupt Method)

Step	Register/Bitfield/Programming Model	Value
Enable interrupt event	MAILBOX_IRQENABLE_SET_u[0 + m*2]	1h
User(processor) can perform anothr task until interrupt occurs		

18.1.4.1.3 Events Servicing

18.1.4.1.3.1 Sending Mode

Table 18-12 describes the events servicing in sending mode.

Table 18-12. Events Servicing in Sending Mode

Step	Register/Bitfield/Programming Model	Value
Read interrupt status bit	MAILBOX_IRQSTATUS_CLR_u[1 + m*2]	1
Write message	MAILBOX_MESSAGE_m[31:0].MESSAGEVALUEMBM	----h
Write 1 to acknowledge interrupt	MAILBOX_IRQSTATUS_CLR_u[1 + m*2]	1

18.1.4.1.3.2 Receiving Mode

Table 18-13 describes the events servicing in receiving mode.

Table 18-13. Events Servicing in Receiving Mode

Step	Register/Bitfield/Programming Model	Value
Read interrupt status bit	MAILBOX_IRQSTATUS_CLR_u[0 + m*2]	1
IF : Number of messages is not equal to 0 ?	MAILBOX_MSGSTATUS_m[2:0].NBOFMSGMB	!=0h
Read message	MAILBOX_MESSAGE_m[31:0].MESSAGEVALUEMBM	----h
ELSE		
Write 1 to acknowledge interrupt	MAILBOX_IRQSTATUS_CLR_u[0 + m*2]	1
ENDIF		

18.1.5 MAILBOX Registers

Table 18-14 lists the memory-mapped registers for the MAILBOX. All register offset addresses not listed in Table 18-14 should be considered as reserved locations and the register contents should not be modified.

Table 18-14. MAILBOX REGISTERS

Offset	Acronym	Register Name	Section
0h	MLB_REVISION		Section 18.1.5.1
10h	MLB_SYSCONFIG		Section 18.1.5.2
40h	MLB_MESSAGE_0		Section 18.1.5.3
44h	MLB_MESSAGE_1		Section 18.1.5.4
48h	MLB_MESSAGE_2		Section 18.1.5.5
4Ch	MLB_MESSAGE_3		Section 18.1.5.6
50h	MLB_MESSAGE_4		Section 18.1.5.7
54h	MLB_MESSAGE_5		Section 18.1.5.8
58h	MLB_MESSAGE_6		Section 18.1.5.9
5Ch	MLB_MESSAGE_7		Section 18.1.5.10
80h	MLB_FIFOSTS_0		Section 18.1.5.11
84h	MLB_FIFOSTS_1		Section 18.1.5.12
88h	MLB_FIFOSTS_2		Section 18.1.5.13
8Ch	MLB_FIFOSTS_3		Section 18.1.5.14
90h	MLB_FIFOSTS_4		Section 18.1.5.15
94h	MLB_FIFOSTS_5		Section 18.1.5.16
98h	MLB_FIFOSTS_6		Section 18.1.5.17
9Ch	MLB_FIFOSTS_7		Section 18.1.5.18
C0h	MLB_MSGSTS_0		Section 18.1.5.19
C4h	MLB_MSGSTS_1		Section 18.1.5.20
C8h	MLB_MSGSTS_2		Section 18.1.5.21
CCh	MLB_MSGSTS_3		Section 18.1.5.22
D0h	MLB_MSGSTS_4		Section 18.1.5.23
D4h	MLB_MSGSTS_5		Section 18.1.5.24
D8h	MLB_MSGSTS_6		Section 18.1.5.25
DCh	MLB_MSGSTS_7		Section 18.1.5.26
100h	MLB_IRQSTS_RAW_0		Section 18.1.5.27
104h	MLB_IRQSTS_CLR_0		Section 18.1.5.28
108h	MLB_IRQEN_SET_0		Section 18.1.5.29
10Ch	MLB_IRQEN_CLR_0		Section 18.1.5.30
110h	MLB_IRQSTS_RAW_1		Section 18.1.5.31
114h	MLB_IRQSTS_CLR_1		Section 18.1.5.32
118h	MLB_IRQEN_SET_1		Section 18.1.5.33
11Ch	MLB_IRQEN_CLR_1		Section 18.1.5.34
120h	MLB_IRQSTS_RAW_2		Section 18.1.5.35
124h	MLB_IRQSTS_CLR_2		Section 18.1.5.36
128h	MLB_IRQEN_SET_2		Section 18.1.5.37
12Ch	MLB_IRQEN_CLR_2		Section 18.1.5.38
130h	MLB_IRQSTS_RAW_3		Section 18.1.5.39
134h	MLB_IRQSTS_CLR_3		Section 18.1.5.40
138h	MLB_IRQEN_SET_3		Section 18.1.5.41
13Ch	MLB_IRQEN_CLR_3		Section 18.1.5.42

18.1.5.1 MLB_REVISION Register (offset = 0h) [reset = 400h]

Register mask: FFFFFFFFh

MLB_REVISION is shown in Figure 18-3 and described in Table 18-15.

This register contains the IP revision code

Figure 18-3. MLB_REVISION Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCHEME		RES		FUNC											
R-0h		R-0h		R-0h											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTL					MAJOR			CUSTOM		MINOR					
R-0h					R-4h			R-0h		R-0h					

Table 18-15. MLB_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	0h	Not defined yet
29-28	RES	R	0h	Reserved
27-16	FUNC	R	0h	Not defined yet
15-11	RTL	R	0h	Not defined yet
10-8	MAJOR	R	4h	IP-Major Revision
7-6	CUSTOM	R	0h	Not Defined Yet
5-0	MINOR	R	0h	IP-Minor Revision

18.1.5.2 MLB_SYSCONFIG Register (offset = 10h) [reset = 8h]

Register mask: FFFFFFFFh

MLB_SYSCONFIG is shown in Figure 18-4 and described in Table 18-16.

This register controls the various parameters of the OCP interface

Figure 18-4. MLB_SYSCONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							

23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED				SIDLEMODE		RESERVED	SOFTRESET
R/W-0h				R/W-2h		R/W-0h	R/W-0h

Table 18-16. MLB_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	Write 0's for future compatibility Reads returns 0
3-2	SIDLEMODE	R/W	2h	
1	RESERVED	R/W	0h	Write 0's for future compatibility Read returns 0
0	SOFTRESET	R/W	0h	Software reset. This bit is automatically reset by the hardware. During reads, it always return 0 0h = Normal mode 1h = The module is reset

18.1.5.3 MLB_MESSAGE_0 Register (offset = 40h) [reset = 0h]

MLB_MESSAGE_0 is shown in Figure 18-5 and described in Table 18-17.

The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue

Figure 18-5. MLB_MESSAGE_0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">MESSAGEVALUEMBM</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 18-17. MLB_MESSAGE_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MESSAGEVALUEMBM	R/W	0h	Message in Mailbox. The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue.

18.1.5.4 MLB_MESSAGE_1 Register (offset = 44h) [reset = 0h]

MLB_MESSAGE_1 is shown in Figure 18-6 and described in Table 18-18.

The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue

Figure 18-6. MLB_MESSAGE_1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">MESSAGEVALUEMBM</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 18-18. MLB_MESSAGE_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MESSAGEVALUEMBM	R/W	0h	Message in Mailbox. The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue.

18.1.5.5 MLB_MESSAGE_2 Register (offset = 48h) [reset = 0h]

MLB_MESSAGE_2 is shown in Figure 18-7 and described in Table 18-19.

The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue

Figure 18-7. MLB_MESSAGE_2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MESSAGEVALUEMBM																															
R/W-0h																															

Table 18-19. MLB_MESSAGE_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MESSAGEVALUEMBM	R/W	0h	Message in Mailbox. The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue.

18.1.5.6 MLB_MESSAGE_3 Register (offset = 4Ch) [reset = 0h]

MLB_MESSAGE_3 is shown in Figure 18-8 and described in Table 18-20.

The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue

Figure 18-8. MLB_MESSAGE_3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MESSAGEVALUEMBM																															
R/W-0h																															

Table 18-20. MLB_MESSAGE_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MESSAGEVALUEMBM	R/W	0h	Message in Mailbox. The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue.

18.1.5.7 MLB_MESSAGE_4 Register (offset = 50h) [reset = 0h]

MLB_MESSAGE_4 is shown in Figure 18-9 and described in Table 18-21.

The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue

Figure 18-9. MLB_MESSAGE_4 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">MESSAGEVALUEMBM</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 18-21. MLB_MESSAGE_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MESSAGEVALUEMBM	R/W	0h	Message in Mailbox. The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue.

18.1.5.8 MLB_MESSAGE_5 Register (offset = 54h) [reset = 0h]

MLB_MESSAGE_5 is shown in Figure 18-10 and described in Table 18-22.

The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue

Figure 18-10. MLB_MESSAGE_5 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">MESSAGEVALUEMBM</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 18-22. MLB_MESSAGE_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MESSAGEVALUEMBM	R/W	0h	Message in Mailbox. The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue.

18.1.5.9 MLB_MESSAGE_6 Register (offset = 58h) [reset = 0h]

MLB_MESSAGE_6 is shown in Figure 18-11 and described in Table 18-23.

The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue

Figure 18-11. MLB_MESSAGE_6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MESSAGEVALUEMBM																															
R/W-0h																															

Table 18-23. MLB_MESSAGE_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MESSAGEVALUEMBM	R/W	0h	Message in Mailbox. The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue.

18.1.5.10 MLB_MESSAGE_7 Register (offset = 5Ch) [reset = 0h]

MLB_MESSAGE_7 is shown in Figure 18-12 and described in Table 18-24.

The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue

Figure 18-12. MLB_MESSAGE_7 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">MESSAGEVALUEMBM</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 18-24. MLB_MESSAGE_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MESSAGEVALUEMBM	R/W	0h	Message in Mailbox. The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue.

18.1.5.11 MLB_FIFOSTS_0 Register (offset = 80h) [reset = 0h]

MLB_FIFOSTS_0 is shown in Figure 18-13 and described in Table 18-25.

The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue

Figure 18-13. MLB_FIFOSTS_0 Register

31	30	29	28	27	26	25	24
MESSAGEVALUEMBM							
R/W-0h							

23	22	21	20	19	18	17	16
MESSAGEVALUEMBM							
R/W-0h							

15	14	13	12	11	10	9	8
MESSAGEVALUEMBM							
R/W-0h							

7	6	5	4	3	2	1	0
MESSAGEVALUEMBM							FIFOFULLMBM
R/W-0h							R-0h

Table 18-25. MLB_FIFOSTS_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	MESSAGEVALUEMBM	R/W	0h	Message in Mailbox. The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue.
0	FIFOFULLMBM	R	0h	Full flag for Mailbox 0h = Mailbox FIFO is not full 1h = Mailbox FIFO is full

18.1.5.12 MLB_FIFOSTS_1 Register (offset = 84h) [reset = 0h]

MLB_FIFOSTS_1 is shown in Figure 18-14 and described in Table 18-26.

The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue

Figure 18-14. MLB_FIFOSTS_1 Register

31	30	29	28	27	26	25	24
MESSAGEVALUEMBM							
R/W-0h							

23	22	21	20	19	18	17	16
MESSAGEVALUEMBM							
R/W-0h							

15	14	13	12	11	10	9	8
MESSAGEVALUEMBM							
R/W-0h							

7	6	5	4	3	2	1	0
MESSAGEVALUEMBM							FIFOFULLMBM
R/W-0h							R-0h

Table 18-26. MLB_FIFOSTS_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	MESSAGEVALUEMBM	R/W	0h	Message in Mailbox. The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue.
0	FIFOFULLMBM	R	0h	Full flag for Mailbox 0h = Mailbox FIFO is not full 1h = Mailbox FIFO is full

18.1.5.13 MLB_FIFOSTS_2 Register (offset = 88h) [reset = 0h]

MLB_FIFOSTS_2 is shown in Figure 18-15 and described in Table 18-27.

The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue

Figure 18-15. MLB_FIFOSTS_2 Register

31	30	29	28	27	26	25	24
MESSAGEVALUEMBM							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
MESSAGEVALUEMBM							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MESSAGEVALUEMBM							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MESSAGEVALUEMBM							FIFOFULLMBM
R/W-0h							R-0h

Table 18-27. MLB_FIFOSTS_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	MESSAGEVALUEMBM	R/W	0h	Message in Mailbox. The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue.
0	FIFOFULLMBM	R	0h	Full flag for Mailbox 0h = Mailbox FIFO is not full 1h = Mailbox FIFO is full

18.1.5.14 MLB_FIFOSTS_3 Register (offset = 8Ch) [reset = 0h]

MLB_FIFOSTS_3 is shown in Figure 18-16 and described in Table 18-28.

The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue

Figure 18-16. MLB_FIFOSTS_3 Register

31	30	29	28	27	26	25	24
MESSAGEVALUEMBM							
R/W-0h							

23	22	21	20	19	18	17	16
MESSAGEVALUEMBM							
R/W-0h							

15	14	13	12	11	10	9	8
MESSAGEVALUEMBM							
R/W-0h							

7	6	5	4	3	2	1	0
MESSAGEVALUEMBM							FIFOFULLMBM
R/W-0h							R-0h

Table 18-28. MLB_FIFOSTS_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	MESSAGEVALUEMBM	R/W	0h	Message in Mailbox. The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue.
0	FIFOFULLMBM	R	0h	Full flag for Mailbox 0h = Mailbox FIFO is not full 1h = Mailbox FIFO is full

18.1.5.15 MLB_FIFOSTS_4 Register (offset = 90h) [reset = 0h]

MLB_FIFOSTS_4 is shown in Figure 18-17 and described in Table 18-29.

The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue

Figure 18-17. MLB_FIFOSTS_4 Register

31	30	29	28	27	26	25	24
MESSAGEVALUEMBM							
R/W-0h							

23	22	21	20	19	18	17	16
MESSAGEVALUEMBM							
R/W-0h							

15	14	13	12	11	10	9	8
MESSAGEVALUEMBM							
R/W-0h							

7	6	5	4	3	2	1	0
MESSAGEVALUEMBM							FIFOFULLMBM
R/W-0h							R-0h

Table 18-29. MLB_FIFOSTS_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	MESSAGEVALUEMBM	R/W	0h	Message in Mailbox. The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue.
0	FIFOFULLMBM	R	0h	Full flag for Mailbox 0h = Mailbox FIFO is not full 1h = Mailbox FIFO is full

18.1.5.16 MLB_FIFOSTS_5 Register (offset = 94h) [reset = 0h]

MLB_FIFOSTS_5 is shown in Figure 18-18 and described in Table 18-30.

The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue

Figure 18-18. MLB_FIFOSTS_5 Register

31	30	29	28	27	26	25	24
MESSAGEVALUEMBM							
R/W-0h							

23	22	21	20	19	18	17	16
MESSAGEVALUEMBM							
R/W-0h							

15	14	13	12	11	10	9	8
MESSAGEVALUEMBM							
R/W-0h							

7	6	5	4	3	2	1	0
MESSAGEVALUEMBM							FIFOFULLMBM
R/W-0h							R-0h

Table 18-30. MLB_FIFOSTS_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	MESSAGEVALUEMBM	R/W	0h	Message in Mailbox. The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue.
0	FIFOFULLMBM	R	0h	Full flag for Mailbox 0h = Mailbox FIFO is not full 1h = Mailbox FIFO is full

18.1.5.17 MLB_FIFOSTS_6 Register (offset = 98h) [reset = 0h]

MLB_FIFOSTS_6 is shown in Figure 18-19 and described in Table 18-31.

The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue

Figure 18-19. MLB_FIFOSTS_6 Register

31	30	29	28	27	26	25	24
MESSAGEVALUEMBM							
R/W-0h							

23	22	21	20	19	18	17	16
MESSAGEVALUEMBM							
R/W-0h							

15	14	13	12	11	10	9	8
MESSAGEVALUEMBM							
R/W-0h							

7	6	5	4	3	2	1	0
MESSAGEVALUEMBM							FIFOFULLMBM
R/W-0h							R-0h

Table 18-31. MLB_FIFOSTS_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	MESSAGEVALUEMBM	R/W	0h	Message in Mailbox. The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue.
0	FIFOFULLMBM	R	0h	Full flag for Mailbox 0h = Mailbox FIFO is not full 1h = Mailbox FIFO is full

18.1.5.18 MLB_FIFOSTS_7 Register (offset = 9Ch) [reset = 0h]

MLB_FIFOSTS_7 is shown in Figure 18-20 and described in Table 18-32.

The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue

Figure 18-20. MLB_FIFOSTS_7 Register

31	30	29	28	27	26	25	24
MESSAGEVALUEMBM							
R/W-0h							

23	22	21	20	19	18	17	16
MESSAGEVALUEMBM							
R/W-0h							

15	14	13	12	11	10	9	8
MESSAGEVALUEMBM							
R/W-0h							

7	6	5	4	3	2	1	0
MESSAGEVALUEMBM							FIFOFULLMBM
R/W-0h							R-0h

Table 18-32. MLB_FIFOSTS_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	MESSAGEVALUEMBM	R/W	0h	Message in Mailbox. The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue.
0	FIFOFULLMBM	R	0h	Full flag for Mailbox 0h = Mailbox FIFO is not full 1h = Mailbox FIFO is full

18.1.5.19 MLB_MSGSTS_0 Register (offset = C0h) [reset = 0h]

MLB_MSGSTS_0 is shown in Figure 18-21 and described in Table 18-33.

The message status register has the status of the messages in the mailbox

Figure 18-21. MLB_MSGSTS_0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="29">RESERVED</td><td colspan="3">NBOFMSG MBM</td></tr>
<tr><td colspan="29">R-</td><td colspan="3">R-0h</td></tr>
</table>

Table 18-33. MLB_MSGSTS_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R		
2-0	NBOFMSGMBM	R	0h	Number of unread messages in Mailbox. Limited to four messages per mailbox.

18.1.5.20 MLB_MSGSTS_1 Register (offset = C4h) [reset = 0h]

MLB_MSGSTS_1 is shown in Figure 18-22 and described in Table 18-34.

The message status register has the status of the messages in the mailbox

Figure 18-22. MLB_MSGSTS_1 Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3	2 1 0
RESERVED	NBOFMSG MBM
R-	R-0h

Table 18-34. MLB_MSGSTS_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R		
2-0	NBOFMSGMBM	R	0h	Number of unread messages in Mailbox. Limited to four messages per mailbox.

18.1.5.21 MLB_MSGSTS_2 Register (offset = C8h) [reset = 0h]

MLB_MSGSTS_2 is shown in Figure 18-23 and described in Table 18-35.

The message status register has the status of the messages in the mailbox

Figure 18-23. MLB_MSGSTS_2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													NBOFMSGMBM		
R-																													R-0h		

Table 18-35. MLB_MSGSTS_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R		
2-0	NBOFMSGMBM	R	0h	Number of unread messages in Mailbox. Limited to four messages per mailbox.

18.1.5.22 MLB_MSGSTS_3 Register (offset = CCh) [reset = 0h]

MLB_MSGSTS_3 is shown in Figure 18-24 and described in Table 18-36.

The message status register has the status of the messages in the mailbox

Figure 18-24. MLB_MSGSTS_3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													NBOFMSGMBM		
R-																													R-0h		

Table 18-36. MLB_MSGSTS_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R		
2-0	NBOFMSGMBM	R	0h	Number of unread messages in Mailbox. Limited to four messages per mailbox.

18.1.5.23 MLB_MSGSTS_4 Register (offset = D0h) [reset = 0h]

MLB_MSGSTS_4 is shown in Figure 18-25 and described in Table 18-37.

The message status register has the status of the messages in the mailbox

Figure 18-25. MLB_MSGSTS_4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													NBOFMSGMBM		
R-																													R-0h		

Table 18-37. MLB_MSGSTS_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R		
2-0	NBOFMSGMBM	R	0h	Number of unread messages in Mailbox. Limited to four messages per mailbox.

18.1.5.24 MLB_MSGSTS_5 Register (offset = D4h) [reset = 0h]

MLB_MSGSTS_5 is shown in Figure 18-26 and described in Table 18-38.

The message status register has the status of the messages in the mailbox

Figure 18-26. MLB_MSGSTS_5 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="29">RESERVED</td><td colspan="3">NBOFMSGMBM</td></tr>
<tr><td colspan="29">R-</td><td colspan="3">R-0h</td></tr>
</table>

Table 18-38. MLB_MSGSTS_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R		
2-0	NBOFMSGMBM	R	0h	Number of unread messages in Mailbox. Limited to four messages per mailbox.

18.1.5.25 MLB_MSGSTS_6 Register (offset = D8h) [reset = 0h]

MLB_MSGSTS_6 is shown in Figure 18-27 and described in Table 18-39.

The message status register has the status of the messages in the mailbox

Figure 18-27. MLB_MSGSTS_6 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="29">RESERVED</td><td colspan="3">NBOFMSGMBM</td></tr>
<tr><td colspan="29">R-</td><td colspan="3">R-0h</td></tr>
</table>

Table 18-39. MLB_MSGSTS_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R		
2-0	NBOFMSGMBM	R	0h	Number of unread messages in Mailbox. Limited to four messages per mailbox.

18.1.5.26 MLB_MSGSTS_7 Register (offset = DCh) [reset = 0h]

MLB_MSGSTS_7 is shown in Figure 18-28 and described in Table 18-40.

The message status register has the status of the messages in the mailbox

Figure 18-28. MLB_MSGSTS_7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													NBOFMSG MBM		
R-																													R-0h		

Table 18-40. MLB_MSGSTS_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R		
2-0	NBOFMSGMBM	R	0h	Number of unread messages in Mailbox. Limited to four messages per mailbox.

18.1.5.27 MLB_IRQSTS_RAW_0 Register (offset = 100h) [reset = 0h]

Register mask: FFFFFFFFh

MLB_IRQSTS_RAW_0 is shown in Figure 18-29 and described in Table 18-41.

The interrupt status register has the status for each event that may be responsible for the generation of an interrupt to the corresponding user - write 1 to a given bit resets this bit. This register is mainly used for debug purpose.

Figure 18-29. MLB_IRQSTS_RAW_0 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
NOTFULLSTS UUMB7	NEWMSGSTS UUMB7	NOTFULLSTS UUMB6	NEWMSGSTS UUMB6	NOTFULLSTS UUMB5	NEWMSGSTS UUMB5	NOTFULLSTS UUMB4	NEWMSGSTS UUMB4
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
NOTFULLSTS UUMB3	NEWMSGSTS UUMB3	NOTFULLSTS UUMB2	NEWMSGSTS UUMB2	NOTFULLSTS UUMB1	NEWMSGSTS UUMB1	NOTFULLSTS UUMB0	NEWMSGSTS UUMB0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-41. MLB_IRQSTS_RAW_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NOTFULLSTSUUMB7	R/W	0h	Not Full Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
14	NEWMSGSTSUUMB7	R/W	0h	New Message Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
13	NOTFULLSTSUUMB6	R/W	0h	Not Full Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
12	NEWMSGSTSUUMB6	R/W	0h	New Message Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
11	NOTFULLSTSUUMB5	R/W	0h	Not Full Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
10	NEWMSGSTSUUMB5	R/W	0h	New Message Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
9	NOTFULLSTSUUMB4	R/W	0h	Not Full Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)
8	NEWMSGSTSUUMB4	R/W	0h	New Message Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)

Table 18-41. MLB_IRQSTS_RAW_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	NOTFULLSTSUUMB3	R/W	0h	Not Full Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
6	NEWMSGSTSUUMB3	R/W	0h	New Message Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
5	NOTFULLSTSUUMB2	R/W	0h	Not Full Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
4	NEWMSGSTSUUMB2	R/W	0h	New Message Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
3	NOTFULLSTSUUMB1	R/W	0h	Not Full Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
2	NEWMSGSTSUUMB1	R/W	0h	New Message Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
1	NOTFULLSTSUUMB0	R/W	0h	Not Full Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)
0	NEWMSGSTSUUMB0	R/W	0h	New Message Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)

18.1.5.28 MLB_IRQSTS_CLR_0 Register (offset = 104h) [reset = 0h]

Register mask: FFFFFFFFh

MLB_IRQSTS_CLR_0 is shown in Figure 18-30 and described in Table 18-42.

The interrupt status register has the status combined with irq-enable for each event that may be responsible for the generation of an interrupt to the corresponding user - write 1 to a given bit resets this bit.

Figure 18-30. MLB_IRQSTS_CLR_0 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
NOTFULLSTSUUMB7	NEWMSGSTSUUMB7	NOTFULLSTSUUMB6	NEWMSGSTSUUMB6	NOTFULLSTSUUMB5	NEWMSGSTSUUMB5	NOTFULLSTSUUMB4	NEWMSGSTSUUMB4
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
NOTFULLSTSUUMB3	NEWMSGSTSUUMB3	NOTFULLSTSUUMB2	NEWMSGSTSUUMB2	NOTFULLSTSUUMB1	NEWMSGSTSUUMB1	NOTFULLSTSUUMB0	NEWMSGSTSUUMB0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-42. MLB_IRQSTS_CLR_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NOTFULLSTSUUMB7	R/W	0h	Not Full Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
14	NEWMSGSTSUUMB7	R/W	0h	New Message Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
13	NOTFULLSTSUUMB6	R/W	0h	Not Full Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
12	NEWMSGSTSUUMB6	R/W	0h	New Message Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
11	NOTFULLSTSUUMB5	R/W	0h	Not Full Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
10	NEWMSGSTSUUMB5	R/W	0h	New Message Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
9	NOTFULLSTSUUMB4	R/W	0h	Not Full Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)
8	NEWMSGSTSUUMB4	R/W	0h	New Message Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)

Table 18-42. MLB_IRQSTS_CLR_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	NOTFULLSTSUUMB3	R/W	0h	Not Full Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
6	NEWMSGSTSUUMB3	R/W	0h	New Message Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
5	NOTFULLSTSUUMB2	R/W	0h	Not Full Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
4	NEWMSGSTSUUMB2	R/W	0h	New Message Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
3	NOTFULLSTSUUMB1	R/W	0h	Not Full Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
2	NEWMSGSTSUUMB1	R/W	0h	New Message Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
1	NOTFULLSTSUUMB0	R/W	0h	Not Full Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)
0	NEWMSGSTSUUMB0	R/W	0h	New Message Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)

18.1.5.29 MLB_IRQEN_SET_0 Register (offset = 108h) [reset = 0h]

Register mask: FFFFFFFFh

MLB_IRQEN_SET_0 is shown in Figure 18-31 and described in Table 18-43.

The interrupt enable register enables to unmask the module internal source of interrupt to the corresponding user. This register is write 1 to set.

Figure 18-31. MLB_IRQEN_SET_0 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
NOTFULLSTS UUMB7	NEWMSGSTS UUMB7	NOTFULLSTS UUMB6	NEWMSGSTS UUMB6	NOTFULLSTS UUMB5	NEWMSGSTS UUMB5	NOTFULLSTS UUMB4	NEWMSGSTS UUMB4
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
NOTFULLSTS UUMB3	NEWMSGSTS UUMB3	NOTFULLSTS UUMB2	NEWMSGSTS UUMB2	NOTFULLSTS UUMB1	NEWMSGSTS UUMB1	NOTFULLSTS UUMB0	NEWMSGSTS UUMB0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-43. MLB_IRQEN_SET_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NOTFULLSTSUUMB7	R/W	0h	Not Full Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
14	NEWMSGSTSUUMB7	R/W	0h	New Message Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
13	NOTFULLSTSUUMB6	R/W	0h	Not Full Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
12	NEWMSGSTSUUMB6	R/W	0h	New Message Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
11	NOTFULLSTSUUMB5	R/W	0h	Not Full Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
10	NEWMSGSTSUUMB5	R/W	0h	New Message Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
9	NOTFULLSTSUUMB4	R/W	0h	Not Full Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)
8	NEWMSGSTSUUMB4	R/W	0h	New Message Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)

Table 18-43. MLB_IRQEN_SET_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	NOTFULLSTSUUMB3	R/W	0h	Not Full Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
6	NEWMSGSTSUUMB3	R/W	0h	New Message Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
5	NOTFULLSTSUUMB2	R/W	0h	Not Full Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
4	NEWMSGSTSUUMB2	R/W	0h	New Message Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
3	NOTFULLSTSUUMB1	R/W	0h	Not Full Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
2	NEWMSGSTSUUMB1	R/W	0h	New Message Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
1	NOTFULLSTSUUMB0	R/W	0h	Not Full Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)
0	NEWMSGSTSUUMB0	R/W	0h	New Message Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)

18.1.5.30 MLB_IRQEN_CLR_0 Register (offset = 10Ch) [reset = 0h]

Register mask: FFFFFFFFh

MLB_IRQEN_CLR_0 is shown in Figure 18-32 and described in Table 18-44.

The interrupt enable register enables to mask the module internal source of interrupt to the corresponding user. This register is write 1 to clear.

Figure 18-32. MLB_IRQEN_CLR_0 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
NOTFULLSTSUUMB7	NEWMSGSTSUUMB7	NOTFULLSTSUUMB6	NEWMSGSTSUUMB6	NOTFULLSTSUUMB5	NEWMSGSTSUUMB5	NOTFULLSTSUUMB4	NEWMSGSTSUUMB4
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
NOTFULLSTSUUMB3	NEWMSGSTSUUMB3	NOTFULLSTSUUMB2	NEWMSGSTSUUMB2	NOTFULLSTSUUMB1	NEWMSGSTSUUMB1	NOTFULLSTSUUMB0	NEWMSGSTSUUMB0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-44. MLB_IRQEN_CLR_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NOTFULLSTSUUMB7	R/W	0h	Not Full Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
14	NEWMSGSTSUUMB7	R/W	0h	New Message Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
13	NOTFULLSTSUUMB6	R/W	0h	Not Full Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
12	NEWMSGSTSUUMB6	R/W	0h	New Message Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
11	NOTFULLSTSUUMB5	R/W	0h	Not Full Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
10	NEWMSGSTSUUMB5	R/W	0h	New Message Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
9	NOTFULLSTSUUMB4	R/W	0h	Not Full Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)
8	NEWMSGSTSUUMB4	R/W	0h	New Message Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)

Table 18-44. MLB_IRQEN_CLR_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	NOTFULLSTSUUMB3	R/W	0h	Not Full Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
6	NEWMSGSTSUUMB3	R/W	0h	New Message Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
5	NOTFULLSTSUUMB2	R/W	0h	Not Full Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
4	NEWMSGSTSUUMB2	R/W	0h	New Message Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
3	NOTFULLSTSUUMB1	R/W	0h	Not Full Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
2	NEWMSGSTSUUMB1	R/W	0h	New Message Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
1	NOTFULLSTSUUMB0	R/W	0h	Not Full Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)
0	NEWMSGSTSUUMB0	R/W	0h	New Message Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)

18.1.5.31 MLB_IRQSTS_RAW_1 Register (offset = 110h) [reset = 0h]

Register mask: FFFFFFFFh

MLB_IRQSTS_RAW_1 is shown in Figure 18-33 and described in Table 18-45.

The interrupt status register has the status for each event that may be responsible for the generation of an interrupt to the corresponding user - write 1 to a given bit resets this bit. This register is mainly used for debug purpose.

Figure 18-33. MLB_IRQSTS_RAW_1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
NOTFULLSTS UUMB7	NEWMSGSTS UUMB7	NOTFULLSTS UUMB6	NEWMSGSTS UUMB6	NOTFULLSTS UUMB5	NEWMSGSTS UUMB5	NOTFULLSTS UUMB4	NEWMSGSTS UUMB4
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
NOTFULLSTS UUMB3	NEWMSGSTS UUMB3	NOTFULLSTS UUMB2	NEWMSGSTS UUMB2	NOTFULLSTS UUMB1	NEWMSGSTS UUMB1	NOTFULLSTS UUMB0	NEWMSGSTS UUMB0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-45. MLB_IRQSTS_RAW_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NOTFULLSTSUUMB7	R/W	0h	Not Full Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
14	NEWMSGSTSUUMB7	R/W	0h	New Message Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
13	NOTFULLSTSUUMB6	R/W	0h	Not Full Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
12	NEWMSGSTSUUMB6	R/W	0h	New Message Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
11	NOTFULLSTSUUMB5	R/W	0h	Not Full Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
10	NEWMSGSTSUUMB5	R/W	0h	New Message Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
9	NOTFULLSTSUUMB4	R/W	0h	Not Full Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)
8	NEWMSGSTSUUMB4	R/W	0h	New Message Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)

Table 18-45. MLB_IRQSTS_RAW_1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	NOTFULLSTSUUMB3	R/W	0h	Not Full Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
6	NEWMSGSTSUUMB3	R/W	0h	New Message Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
5	NOTFULLSTSUUMB2	R/W	0h	Not Full Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
4	NEWMSGSTSUUMB2	R/W	0h	New Message Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
3	NOTFULLSTSUUMB1	R/W	0h	Not Full Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
2	NEWMSGSTSUUMB1	R/W	0h	New Message Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
1	NOTFULLSTSUUMB0	R/W	0h	Not Full Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)
0	NEWMSGSTSUUMB0	R/W	0h	New Message Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)

18.1.5.32 MLB_IRQSTS_CLR_1 Register (offset = 114h) [reset = 0h]

Register mask: FFFFFFFFh

MLB_IRQSTS_CLR_1 is shown in Figure 18-34 and described in Table 18-46.

The interrupt status register has the status combined with irq-enable for each event that may be responsible for the generation of an interrupt to the corresponding user - write 1 to a given bit resets this bit.

Figure 18-34. MLB_IRQSTS_CLR_1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
NOTFULLSTS UUMB7	NEWMSGSTS UUMB7	NOTFULLSTS UUMB6	NEWMSGSTS UUMB6	NOTFULLSTS UUMB5	NEWMSGSTS UUMB5	NOTFULLSTS UUMB4	NEWMSGSTS UUMB4
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
NOTFULLSTS UUMB3	NEWMSGSTS UUMB3	NOTFULLSTS UUMB2	NEWMSGSTS UUMB2	NOTFULLSTS UUMB1	NEWMSGSTS UUMB1	NOTFULLSTS UUMB0	NEWMSGSTS UUMB0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-46. MLB_IRQSTS_CLR_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NOTFULLSTSUUMB7	R/W	0h	Not Full Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
14	NEWMSGSTSUUMB7	R/W	0h	New Message Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
13	NOTFULLSTSUUMB6	R/W	0h	Not Full Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
12	NEWMSGSTSUUMB6	R/W	0h	New Message Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
11	NOTFULLSTSUUMB5	R/W	0h	Not Full Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
10	NEWMSGSTSUUMB5	R/W	0h	New Message Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
9	NOTFULLSTSUUMB4	R/W	0h	Not Full Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)
8	NEWMSGSTSUUMB4	R/W	0h	New Message Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)

Table 18-46. MLB_IRQSTS_CLR_1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	NOTFULLSTSUUMB3	R/W	0h	Not Full Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
6	NEWMSGSTSUUMB3	R/W	0h	New Message Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
5	NOTFULLSTSUUMB2	R/W	0h	Not Full Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
4	NEWMSGSTSUUMB2	R/W	0h	New Message Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
3	NOTFULLSTSUUMB1	R/W	0h	Not Full Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
2	NEWMSGSTSUUMB1	R/W	0h	New Message Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
1	NOTFULLSTSUUMB0	R/W	0h	Not Full Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)
0	NEWMSGSTSUUMB0	R/W	0h	New Message Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)

18.1.5.33 MLB_IRQEN_SET_1 Register (offset = 118h) [reset = 0h]

Register mask: FFFFFFFFh

MLB_IRQEN_SET_1 is shown in Figure 18-35 and described in Table 18-47.

The interrupt enable register enables to unmask the module internal source of interrupt to the corresponding user. This register is write 1 to set.

Figure 18-35. MLB_IRQEN_SET_1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
NOTFULLSTS UUMB7	NEWMSGSTS UUMB7	NOTFULLSTS UUMB6	NEWMSGSTS UUMB6	NOTFULLSTS UUMB5	NEWMSGSTS UUMB5	NOTFULLSTS UUMB4	NEWMSGSTS UUMB4
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
NOTFULLSTS UUMB3	NEWMSGSTS UUMB3	NOTFULLSTS UUMB2	NEWMSGSTS UUMB2	NOTFULLSTS UUMB1	NEWMSGSTS UUMB1	NOTFULLSTS UUMB0	NEWMSGSTS UUMB0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-47. MLB_IRQEN_SET_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NOTFULLSTSUUMB7	R/W	0h	Not Full Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
14	NEWMSGSTSUUMB7	R/W	0h	New Message Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
13	NOTFULLSTSUUMB6	R/W	0h	Not Full Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
12	NEWMSGSTSUUMB6	R/W	0h	New Message Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
11	NOTFULLSTSUUMB5	R/W	0h	Not Full Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
10	NEWMSGSTSUUMB5	R/W	0h	New Message Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
9	NOTFULLSTSUUMB4	R/W	0h	Not Full Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)
8	NEWMSGSTSUUMB4	R/W	0h	New Message Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)

Table 18-47. MLB_IRQEN_SET_1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	NOTFULLSTSUUMB3	R/W	0h	Not Full Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
6	NEWMSGSTSUUMB3	R/W	0h	New Message Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
5	NOTFULLSTSUUMB2	R/W	0h	Not Full Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
4	NEWMSGSTSUUMB2	R/W	0h	New Message Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
3	NOTFULLSTSUUMB1	R/W	0h	Not Full Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
2	NEWMSGSTSUUMB1	R/W	0h	New Message Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
1	NOTFULLSTSUUMB0	R/W	0h	Not Full Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)
0	NEWMSGSTSUUMB0	R/W	0h	New Message Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)

18.1.5.34 MLB_IRQEN_CLR_1 Register (offset = 11Ch) [reset = 0h]

Register mask: FFFFFFFFh

MLB_IRQEN_CLR_1 is shown in Figure 18-36 and described in Table 18-48.

The interrupt enable register enables to mask the module internal source of interrupt to the corresponding user. This register is write 1 to clear.

Figure 18-36. MLB_IRQEN_CLR_1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
NOTFULLSTS UUMB7	NEWMSGSTS UUMB7	NOTFULLSTS UUMB6	NEWMSGSTS UUMB6	NOTFULLSTS UUMB5	NEWMSGSTS UUMB5	NOTFULLSTS UUMB4	NEWMSGSTS UUMB4
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
NOTFULLSTS UUMB3	NEWMSGSTS UUMB3	NOTFULLSTS UUMB2	NEWMSGSTS UUMB2	NOTFULLSTS UUMB1	NEWMSGSTS UUMB1	NOTFULLSTS UUMB0	NEWMSGSTS UUMB0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-48. MLB_IRQEN_CLR_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NOTFULLSTSUUMB7	R/W	0h	Not Full Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
14	NEWMSGSTSUUMB7	R/W	0h	New Message Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
13	NOTFULLSTSUUMB6	R/W	0h	Not Full Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
12	NEWMSGSTSUUMB6	R/W	0h	New Message Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
11	NOTFULLSTSUUMB5	R/W	0h	Not Full Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
10	NEWMSGSTSUUMB5	R/W	0h	New Message Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
9	NOTFULLSTSUUMB4	R/W	0h	Not Full Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)
8	NEWMSGSTSUUMB4	R/W	0h	New Message Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)

Table 18-48. MLB_IRQEN_CLR_1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	NOTFULLSTSUUMB3	R/W	0h	Not Full Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
6	NEWMSGSTSUUMB3	R/W	0h	New Message Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
5	NOTFULLSTSUUMB2	R/W	0h	Not Full Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
4	NEWMSGSTSUUMB2	R/W	0h	New Message Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
3	NOTFULLSTSUUMB1	R/W	0h	Not Full Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
2	NEWMSGSTSUUMB1	R/W	0h	New Message Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
1	NOTFULLSTSUUMB0	R/W	0h	Not Full Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)
0	NEWMSGSTSUUMB0	R/W	0h	New Message Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)

18.1.5.35 MLB_IRQSTS_RAW_2 Register (offset = 120h) [reset = 0h]

Register mask: FFFFFFFFh

MLB_IRQSTS_RAW_2 is shown in Figure 18-37 and described in Table 18-49.

The interrupt status register has the status for each event that may be responsible for the generation of an interrupt to the corresponding user - write 1 to a given bit resets this bit. This register is mainly used for debug purpose.

Figure 18-37. MLB_IRQSTS_RAW_2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
NOTFULLSTS UUMB7	NEWMSGSTS UUMB7	NOTFULLSTS UUMB6	NEWMSGSTS UUMB6	NOTFULLSTS UUMB5	NEWMSGSTS UUMB5	NOTFULLSTS UUMB4	NEWMSGSTS UUMB4
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
NOTFULLSTS UUMB3	NEWMSGSTS UUMB3	NOTFULLSTS UUMB2	NEWMSGSTS UUMB2	NOTFULLSTS UUMB1	NEWMSGSTS UUMB1	NOTFULLSTS UUMB0	NEWMSGSTS UUMB0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-49. MLB_IRQSTS_RAW_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NOTFULLSTSUUMB7	R/W	0h	Not Full Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
14	NEWMSGSTSUUMB7	R/W	0h	New Message Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
13	NOTFULLSTSUUMB6	R/W	0h	Not Full Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
12	NEWMSGSTSUUMB6	R/W	0h	New Message Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
11	NOTFULLSTSUUMB5	R/W	0h	Not Full Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
10	NEWMSGSTSUUMB5	R/W	0h	New Message Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
9	NOTFULLSTSUUMB4	R/W	0h	Not Full Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)
8	NEWMSGSTSUUMB4	R/W	0h	New Message Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)

Table 18-49. MLB_IRQSTS_RAW_2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	NOTFULLSTSUUMB3	R/W	0h	Not Full Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
6	NEWMSGSTSUUMB3	R/W	0h	New Message Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
5	NOTFULLSTSUUMB2	R/W	0h	Not Full Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
4	NEWMSGSTSUUMB2	R/W	0h	New Message Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
3	NOTFULLSTSUUMB1	R/W	0h	Not Full Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
2	NEWMSGSTSUUMB1	R/W	0h	New Message Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
1	NOTFULLSTSUUMB0	R/W	0h	Not Full Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)
0	NEWMSGSTSUUMB0	R/W	0h	New Message Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)

18.1.5.36 MLB_IRQSTS_CLR_2 Register (offset = 124h) [reset = 0h]

Register mask: FFFFFFFFh

MLB_IRQSTS_CLR_2 is shown in Figure 18-38 and described in Table 18-50.

The interrupt status register has the status combined with irq-enable for each event that may be responsible for the generation of an interrupt to the corresponding user - write 1 to a given bit resets this bit.

Figure 18-38. MLB_IRQSTS_CLR_2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
NOTFULLSTS UUMB7	NEWMSGSTS UUMB7	NOTFULLSTS UUMB6	NEWMSGSTS UUMB6	NOTFULLSTS UUMB5	NEWMSGSTS UUMB5	NOTFULLSTS UUMB4	NEWMSGSTS UUMB4
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
NOTFULLSTS UUMB3	NEWMSGSTS UUMB3	NOTFULLSTS UUMB2	NEWMSGSTS UUMB2	NOTFULLSTS UUMB1	NEWMSGSTS UUMB1	NOTFULLSTS UUMB0	NEWMSGSTS UUMB0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-50. MLB_IRQSTS_CLR_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NOTFULLSTSUUMB7	R/W	0h	Not Full Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
14	NEWMSGSTSUUMB7	R/W	0h	New Message Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
13	NOTFULLSTSUUMB6	R/W	0h	Not Full Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
12	NEWMSGSTSUUMB6	R/W	0h	New Message Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
11	NOTFULLSTSUUMB5	R/W	0h	Not Full Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
10	NEWMSGSTSUUMB5	R/W	0h	New Message Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
9	NOTFULLSTSUUMB4	R/W	0h	Not Full Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)
8	NEWMSGSTSUUMB4	R/W	0h	New Message Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)

Table 18-50. MLB_IRQSTS_CLR_2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	NOTFULLSTSUUMB3	R/W	0h	Not Full Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
6	NEWMSGSTSUUMB3	R/W	0h	New Message Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
5	NOTFULLSTSUUMB2	R/W	0h	Not Full Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
4	NEWMSGSTSUUMB2	R/W	0h	New Message Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
3	NOTFULLSTSUUMB1	R/W	0h	Not Full Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
2	NEWMSGSTSUUMB1	R/W	0h	New Message Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
1	NOTFULLSTSUUMB0	R/W	0h	Not Full Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)
0	NEWMSGSTSUUMB0	R/W	0h	New Message Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)

18.1.5.37 MLB_IRQEN_SET_2 Register (offset = 128h) [reset = 0h]

Register mask: FFFFFFFFh

MLB_IRQEN_SET_2 is shown in Figure 18-39 and described in Table 18-51.

The interrupt enable register enables to unmask the module internal source of interrupt to the corresponding user. This register is write 1 to set.

Figure 18-39. MLB_IRQEN_SET_2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
NOTFULLSTS UUMB7	NEWMSGSTS UUMB7	NOTFULLSTS UUMB6	NEWMSGSTS UUMB6	NOTFULLSTS UUMB5	NEWMSGSTS UUMB5	NOTFULLSTS UUMB4	NEWMSGSTS UUMB4
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
NOTFULLSTS UUMB3	NEWMSGSTS UUMB3	NOTFULLSTS UUMB2	NEWMSGSTS UUMB2	NOTFULLSTS UUMB1	NEWMSGSTS UUMB1	NOTFULLSTS UUMB0	NEWMSGSTS UUMB0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-51. MLB_IRQEN_SET_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NOTFULLSTSUUMB7	R/W	0h	Not Full Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
14	NEWMSGSTSUUMB7	R/W	0h	New Message Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
13	NOTFULLSTSUUMB6	R/W	0h	Not Full Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
12	NEWMSGSTSUUMB6	R/W	0h	New Message Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
11	NOTFULLSTSUUMB5	R/W	0h	Not Full Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
10	NEWMSGSTSUUMB5	R/W	0h	New Message Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
9	NOTFULLSTSUUMB4	R/W	0h	Not Full Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)
8	NEWMSGSTSUUMB4	R/W	0h	New Message Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)

Table 18-51. MLB_IRQEN_SET_2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	NOTFULLSTSUUMB3	R/W	0h	Not Full Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
6	NEWMSGSTSUUMB3	R/W	0h	New Message Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
5	NOTFULLSTSUUMB2	R/W	0h	Not Full Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
4	NEWMSGSTSUUMB2	R/W	0h	New Message Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
3	NOTFULLSTSUUMB1	R/W	0h	Not Full Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
2	NEWMSGSTSUUMB1	R/W	0h	New Message Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
1	NOTFULLSTSUUMB0	R/W	0h	Not Full Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)
0	NEWMSGSTSUUMB0	R/W	0h	New Message Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)

18.1.5.38 MLB_IRQEN_CLR_2 Register (offset = 12Ch) [reset = 0h]

Register mask: FFFFFFFFh

MLB_IRQEN_CLR_2 is shown in Figure 18-40 and described in Table 18-52.

The interrupt enable register enables to mask the module internal source of interrupt to the corresponding user. This register is write 1 to clear.

Figure 18-40. MLB_IRQEN_CLR_2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
NOTFULLSTS UUMB7	NEWMSGSTS UUMB7	NOTFULLSTS UUMB6	NEWMSGSTS UUMB6	NOTFULLSTS UUMB5	NEWMSGSTS UUMB5	NOTFULLSTS UUMB4	NEWMSGSTS UUMB4
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
NOTFULLSTS UUMB3	NEWMSGSTS UUMB3	NOTFULLSTS UUMB2	NEWMSGSTS UUMB2	NOTFULLSTS UUMB1	NEWMSGSTS UUMB1	NOTFULLSTS UUMB0	NEWMSGSTS UUMB0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-52. MLB_IRQEN_CLR_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NOTFULLSTSUUMB7	R/W	0h	Not Full Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
14	NEWMSGSTSUUMB7	R/W	0h	New Message Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
13	NOTFULLSTSUUMB6	R/W	0h	Not Full Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
12	NEWMSGSTSUUMB6	R/W	0h	New Message Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
11	NOTFULLSTSUUMB5	R/W	0h	Not Full Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
10	NEWMSGSTSUUMB5	R/W	0h	New Message Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
9	NOTFULLSTSUUMB4	R/W	0h	Not Full Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)
8	NEWMSGSTSUUMB4	R/W	0h	New Message Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)

Table 18-52. MLB_IRQEN_CLR_2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	NOTFULLSTSUUMB3	R/W	0h	Not Full Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
6	NEWMSGSTSUUMB3	R/W	0h	New Message Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
5	NOTFULLSTSUUMB2	R/W	0h	Not Full Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
4	NEWMSGSTSUUMB2	R/W	0h	New Message Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
3	NOTFULLSTSUUMB1	R/W	0h	Not Full Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
2	NEWMSGSTSUUMB1	R/W	0h	New Message Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
1	NOTFULLSTSUUMB0	R/W	0h	Not Full Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)
0	NEWMSGSTSUUMB0	R/W	0h	New Message Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)

18.1.5.39 MLB_IRQSTS_RAW_3 Register (offset = 130h) [reset = 0h]

Register mask: FFFFFFFFh

MLB_IRQSTS_RAW_3 is shown in Figure 18-41 and described in Table 18-53.

The interrupt status register has the status for each event that may be responsible for the generation of an interrupt to the corresponding user - write 1 to a given bit resets this bit. This register is mainly used for debug purpose.

Figure 18-41. MLB_IRQSTS_RAW_3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
NOTFULLSTS UUMB7	NEWMSGSTS UUMB7	NOTFULLSTS UUMB6	NEWMSGSTS UUMB6	NOTFULLSTS UUMB5	NEWMSGSTS UUMB5	NOTFULLSTS UUMB4	NEWMSGSTS UUMB4
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
NOTFULLSTS UUMB3	NEWMSGSTS UUMB3	NOTFULLSTS UUMB2	NEWMSGSTS UUMB2	NOTFULLSTS UUMB1	NEWMSGSTS UUMB1	NOTFULLSTS UUMB0	NEWMSGSTS UUMB0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-53. MLB_IRQSTS_RAW_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NOTFULLSTSUUMB7	R/W	0h	Not Full Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
14	NEWMSGSTSUUMB7	R/W	0h	New Message Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
13	NOTFULLSTSUUMB6	R/W	0h	Not Full Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
12	NEWMSGSTSUUMB6	R/W	0h	New Message Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
11	NOTFULLSTSUUMB5	R/W	0h	Not Full Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
10	NEWMSGSTSUUMB5	R/W	0h	New Message Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
9	NOTFULLSTSUUMB4	R/W	0h	Not Full Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)
8	NEWMSGSTSUUMB4	R/W	0h	New Message Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)

Table 18-53. MLB_IRQSTS_RAW_3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	NOTFULLSTSUUMB3	R/W	0h	Not Full Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
6	NEWMSGSTSUUMB3	R/W	0h	New Message Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
5	NOTFULLSTSUUMB2	R/W	0h	Not Full Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
4	NEWMSGSTSUUMB2	R/W	0h	New Message Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
3	NOTFULLSTSUUMB1	R/W	0h	Not Full Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
2	NEWMSGSTSUUMB1	R/W	0h	New Message Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
1	NOTFULLSTSUUMB0	R/W	0h	Not Full Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)
0	NEWMSGSTSUUMB0	R/W	0h	New Message Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)

18.1.5.40 MLB_IRQSTS_CLR_3 Register (offset = 134h) [reset = 0h]

Register mask: FFFFFFFFh

MLB_IRQSTS_CLR_3 is shown in Figure 18-42 and described in Table 18-54.

The interrupt status register has the status combined with irq-enable for each event that may be responsible for the generation of an interrupt to the corresponding user - write 1 to a given bit resets this bit.

Figure 18-42. MLB_IRQSTS_CLR_3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
NOTFULLSTS UUMB7	NEWMSGSTS UUMB7	NOTFULLSTS UUMB6	NEWMSGSTS UUMB6	NOTFULLSTS UUMB5	NEWMSGSTS UUMB5	NOTFULLSTS UUMB4	NEWMSGSTS UUMB4
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
NOTFULLSTS UUMB3	NEWMSGSTS UUMB3	NOTFULLSTS UUMB2	NEWMSGSTS UUMB2	NOTFULLSTS UUMB1	NEWMSGSTS UUMB1	NOTFULLSTS UUMB0	NEWMSGSTS UUMB0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-54. MLB_IRQSTS_CLR_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NOTFULLSTSUUMB7	R/W	0h	Not Full Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
14	NEWMSGSTSUUMB7	R/W	0h	New Message Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
13	NOTFULLSTSUUMB6	R/W	0h	Not Full Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
12	NEWMSGSTSUUMB6	R/W	0h	New Message Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
11	NOTFULLSTSUUMB5	R/W	0h	Not Full Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
10	NEWMSGSTSUUMB5	R/W	0h	New Message Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
9	NOTFULLSTSUUMB4	R/W	0h	Not Full Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)
8	NEWMSGSTSUUMB4	R/W	0h	New Message Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)

Table 18-54. MLB_IRQSTS_CLR_3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	NOTFULLSTSUUMB3	R/W	0h	Not Full Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
6	NEWMSGSTSUUMB3	R/W	0h	New Message Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
5	NOTFULLSTSUUMB2	R/W	0h	Not Full Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
4	NEWMSGSTSUUMB2	R/W	0h	New Message Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
3	NOTFULLSTSUUMB1	R/W	0h	Not Full Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
2	NEWMSGSTSUUMB1	R/W	0h	New Message Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
1	NOTFULLSTSUUMB0	R/W	0h	Not Full Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)
0	NEWMSGSTSUUMB0	R/W	0h	New Message Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)

18.1.5.41 MLB_IRQEN_SET_3 Register (offset = 138h) [reset = 0h]

Register mask: FFFFFFFFh

MLB_IRQEN_SET_3 is shown in Figure 18-43 and described in Table 18-55.

The interrupt enable register enables to unmask the module internal source of interrupt to the corresponding user. This register is write 1 to set.

Figure 18-43. MLB_IRQEN_SET_3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
NOTFULLSTSUUMB7	NEWMSGSTSUUMB7	NOTFULLSTSUUMB6	NEWMSGSTSUUMB6	NOTFULLSTSUUMB5	NEWMSGSTSUUMB5	NOTFULLSTSUUMB4	NEWMSGSTSUUMB4
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
NOTFULLSTSUUMB3	NEWMSGSTSUUMB3	NOTFULLSTSUUMB2	NEWMSGSTSUUMB2	NOTFULLSTSUUMB1	NEWMSGSTSUUMB1	NOTFULLSTSUUMB0	NEWMSGSTSUUMB0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-55. MLB_IRQEN_SET_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NOTFULLSTSUUMB7	R/W	0h	Not Full Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
14	NEWMSGSTSUUMB7	R/W	0h	New Message Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
13	NOTFULLSTSUUMB6	R/W	0h	Not Full Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
12	NEWMSGSTSUUMB6	R/W	0h	New Message Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
11	NOTFULLSTSUUMB5	R/W	0h	Not Full Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
10	NEWMSGSTSUUMB5	R/W	0h	New Message Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
9	NOTFULLSTSUUMB4	R/W	0h	Not Full Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)
8	NEWMSGSTSUUMB4	R/W	0h	New Message Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)

Table 18-55. MLB_IRQEN_SET_3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	NOTFULLSTSUUMB3	R/W	0h	Not Full Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
6	NEWMSGSTSUUMB3	R/W	0h	New Message Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
5	NOTFULLSTSUUMB2	R/W	0h	Not Full Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
4	NEWMSGSTSUUMB2	R/W	0h	New Message Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
3	NOTFULLSTSUUMB1	R/W	0h	Not Full Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
2	NEWMSGSTSUUMB1	R/W	0h	New Message Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
1	NOTFULLSTSUUMB0	R/W	0h	Not Full Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)
0	NEWMSGSTSUUMB0	R/W	0h	New Message Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)

18.1.5.42 MLB_IRQEN_CLR_3 Register (offset = 13Ch) [reset = 0h]

Register mask: FFFFFFFFh

MLB_IRQEN_CLR_3 is shown in Figure 18-44 and described in Table 18-56.

The interrupt enable register enables to mask the module internal source of interrupt to the corresponding user. This register is write 1 to clear.

Figure 18-44. MLB_IRQEN_CLR_3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
NOTFULLSTS UUMB7	NEWMSGSTS UUMB7	NOTFULLSTS UUMB6	NEWMSGSTS UUMB6	NOTFULLSTS UUMB5	NEWMSGSTS UUMB5	NOTFULLSTS UUMB4	NEWMSGSTS UUMB4
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
NOTFULLSTS UUMB3	NEWMSGSTS UUMB3	NOTFULLSTS UUMB2	NEWMSGSTS UUMB2	NOTFULLSTS UUMB1	NEWMSGSTS UUMB1	NOTFULLSTS UUMB0	NEWMSGSTS UUMB0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-56. MLB_IRQEN_CLR_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NOTFULLSTSUUMB7	R/W	0h	Not Full Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
14	NEWMSGSTSUUMB7	R/W	0h	New Message Status bit for User u, Mailbox 7 0h = No action 1h = Set the event (for debug)
13	NOTFULLSTSUUMB6	R/W	0h	Not Full Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
12	NEWMSGSTSUUMB6	R/W	0h	New Message Status bit for User u, Mailbox 6 0h = No action 1h = Set the event (for debug)
11	NOTFULLSTSUUMB5	R/W	0h	Not Full Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
10	NEWMSGSTSUUMB5	R/W	0h	New Message Status bit for User u, Mailbox 5 0h = No action 1h = Set the event (for debug)
9	NOTFULLSTSUUMB4	R/W	0h	Not Full Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)
8	NEWMSGSTSUUMB4	R/W	0h	New Message Status bit for User u, Mailbox 4 0h = No action 1h = Set the event (for debug)

Table 18-56. MLB_IRQEN_CLR_3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	NOTFULLSTSUUMB3	R/W	0h	Not Full Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
6	NEWMSGSTSUUMB3	R/W	0h	New Message Status bit for User u, Mailbox 3 0h = No action 1h = Set the event (for debug)
5	NOTFULLSTSUUMB2	R/W	0h	Not Full Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
4	NEWMSGSTSUUMB2	R/W	0h	New Message Status bit for User u, Mailbox 2 0h = No action 1h = Set the event (for debug)
3	NOTFULLSTSUUMB1	R/W	0h	Not Full Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
2	NEWMSGSTSUUMB1	R/W	0h	New Message Status bit for User u, Mailbox 1 0h = No action 1h = Set the event (for debug)
1	NOTFULLSTSUUMB0	R/W	0h	Not Full Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)
0	NEWMSGSTSUUMB0	R/W	0h	New Message Status bit for User u, Mailbox 0 0h = No action 1h = Set the event (for debug)

18.2 Spinlock

18.2.1 SPINLOCK Registers

Table 18-57 lists the memory-mapped registers for the SPINLOCK. All register offset addresses not listed in Table 18-57 should be considered as reserved locations and the register contents should not be modified.

Table 18-57. SPINLOCK REGISTERS

Offset	Acronym	Register Name	Section
0h	SPINLOCK_REV		Section 18.2.1.1
10h	SPINLOCK_SYSCONFIG		Section 18.2.1.2
14h	SPINLOCK_SYSTS		Section 18.2.1.3
800h	SPINLOCK_REG_0		Section 18.2.1.4
804h	SPINLOCK_REG_1		Section 18.2.1.5
808h	SPINLOCK_REG_2		Section 18.2.1.6
80Ch	SPINLOCK_REG_3		Section 18.2.1.7
810h	SPINLOCK_REG_4		Section 18.2.1.8
814h	SPINLOCK_REG_5		Section 18.2.1.9
818h	SPINLOCK_REG_6		Section 18.2.1.10
81Ch	SPINLOCK_REG_7		Section 18.2.1.11
820h	SPINLOCK_REG_8		Section 18.2.1.12
824h	SPINLOCK_REG_9		Section 18.2.1.13
828h	SPINLOCK_REG_10		Section 18.2.1.14
82Ch	SPINLOCK_REG_11		Section 18.2.1.15
830h	SPINLOCK_REG_12		Section 18.2.1.16
834h	SPINLOCK_REG_13		Section 18.2.1.17
838h	SPINLOCK_REG_14		Section 18.2.1.18
83Ch	SPINLOCK_REG_15		Section 18.2.1.19
840h	SPINLOCK_REG_16		Section 18.2.1.20
844h	SPINLOCK_REG_17		Section 18.2.1.21
848h	SPINLOCK_REG_18		Section 18.2.1.22
84Ch	SPINLOCK_REG_19		Section 18.2.1.23
850h	SPINLOCK_REG_20		Section 18.2.1.24
854h	SPINLOCK_REG_21		Section 18.2.1.25
858h	SPINLOCK_REG_22		Section 18.2.1.26
85Ch	SPINLOCK_REG_23		Section 18.2.1.27
860h	SPINLOCK_REG_24		Section 18.2.1.28
864h	SPINLOCK_REG_25		Section 18.2.1.29
868h	SPINLOCK_REG_26		Section 18.2.1.30
86Ch	SPINLOCK_REG_27		Section 18.2.1.31
870h	SPINLOCK_REG_28		Section 18.2.1.32
874h	SPINLOCK_REG_29		Section 18.2.1.33
878h	SPINLOCK_REG_30		Section 18.2.1.34
87Ch	SPINLOCK_REG_31		Section 18.2.1.35

18.2.1.1 SPINLOCK_REV Register (offset = 0h) [reset = 50020000h]

Register mask: FFFFFFFFh

SPINLOCK_REV is shown in Figure 18-45 and described in Table 18-58.

Read-only IP revision identifier (X.Y.R) used by software to determine features, bugs and compatibility of an instance of this the Spin Lock module.

Figure 18-45. SPINLOCK_REV Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">REV</td></tr>
<tr><td colspan="32">R-50020000h</td></tr>
</table>

Table 18-58. SPINLOCK_REV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REV	R	50020000h	IP Revision Code.

18.2.1.2 SPINLOCK_SYSCONFIG Register (offset = 10h) [reset = 11h]

Register mask: FFFFFFFFh

SPINLOCK_SYSCONFIG is shown in Figure 18-46 and described in Table 18-59.

This register controls the various parameters of the OCP interface. Note that several fields are present by read-only.

Figure 18-46. SPINLOCK_SYSCONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							CLOCKACTIVITY
R-0h							R-0h

7	6	5	4	3	2	1	0
RESERVED			SIDLEMODE		ENWAKEUP	SOFTRESET	AUTOGATING
R-0h			R-2h		R-0h	W-0h	R-1h

Table 18-59. SPINLOCK_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	CLOCKACTIVITY	R	0h	Indicates whether the module requires the OCP when in IDLE mode. 0h = OCP clock is not required by the module during IDLE mode and may be switched off. 1h = OCP clock is required by the module, even during idle mode.
7-5	RESERVED	R	0h	
4-3	SIDLEMODE	R	2h	Control of the slave interface power management IDLE request acknowledgement. 0h = IDLE request is acknowledged unconditionally and immediately. 1h = IDLE request is never acknowledged. 2h = IDLE request acknowledgement is based on the internal module activity. 3h = Reserved. Do not use.
2	ENWAKEUP	R	0h	Asynchronous wakeup gereration. 0h = Wakeup generation is disabled. 1h = Enable wakeup generation.
1	SOFTRESET	W	0h	Module software reset. 0h = No Description 1h = Start a soft reset sequence of the Spin Lock module.
0	AUTOGATING	R	1h	Internal OCP clock gating strategy. 0h = OCP clock is not gated when OCP interface is idle. 1h = Automatic internal OCP clock gating strategy is applied, based on the OCP interface activity.

18.2.1.3 SPINLOCK_SYSTS Register (offset = 14h) [reset = 1000001h]

Register mask: FFFFFFFFh

SPINLOCK_SYSTS is shown in Figure 18-47 and described in Table 18-60.

This register provides status information about this instance of the Spin Lock module.

Figure 18-47. SPINLOCK_SYSTS Register

31	30	29	28	27	26	25	24
NUMLOCKS							
R-1h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
IU7	IU6	IU5	IU4	IU3	IU2	IU1	IU0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

7	6	5	4	3	2	1	0
RESERVED							RESETDONE
R-0h							R-1h

Table 18-60. SPINLOCK_SYSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NUMLOCKS	R	1h	
23-16	RESERVED	R	0h	
15	IU7	R	0h	
14	IU6	R	0h	
13	IU5	R	0h	
12	IU4	R	0h	
11	IU3	R	0h	
10	IU2	R	0h	
9	IU1	R	0h	In-Use flag 1, covering lock registers 32 - 63. Reads as one only if one or more lock registers in this range are TAKEN. If no lock registers are implemented in this range, then this flag always reads as 0.
8	IU0	R	0h	In-Use flag 0, covering lock registers 0 - 31. Reads as one only if one or more lock registers in this range are TAKEN.
7-1	RESERVED	R	0h	reserved
0	RESETDONE	R	1h	0: Reset in progress. 1: Reset is completed.

18.2.1.4 SPINLOCK_REG_0 Register (offset = 800h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_0 is shown in Figure 18-48 and described in Table 18-61.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-48. SPINLOCK_REG_0 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-61. SPINLOCK_REG_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.5 SPINLOCK_REG_1 Register (offset = 804h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_1 is shown in Figure 18-49 and described in Table 18-62.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-49. SPINLOCK_REG_1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-62. SPINLOCK_REG_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.6 SPINLOCK_REG_2 Register (offset = 808h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_2 is shown in Figure 18-50 and described in Table 18-63.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-50. SPINLOCK_REG_2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-63. SPINLOCK_REG_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.7 SPINLOCK_REG_3 Register (offset = 80Ch) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_3 is shown in Figure 18-51 and described in Table 18-64.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-51. SPINLOCK_REG_3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-64. SPINLOCK_REG_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.8 SPINLOCK_REG_4 Register (offset = 810h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_4 is shown in Figure 18-52 and described in Table 18-65.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-52. SPINLOCK_REG_4 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-65. SPINLOCK_REG_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.9 SPINLOCK_REG_5 Register (offset = 814h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_5 is shown in Figure 18-53 and described in Table 18-66.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-53. SPINLOCK_REG_5 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-66. SPINLOCK_REG_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.10 SPINLOCK_REG_6 Register (offset = 818h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_6 is shown in Figure 18-54 and described in Table 18-67.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-54. SPINLOCK_REG_6 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="7">RESERVED</td><td>TAKEN</td></tr>
<tr><td colspan="7">R-0h</td><td>R/W-0h</td></tr>
</table>

Table 18-67. SPINLOCK_REG_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.11 SPINLOCK_REG_7 Register (offset = 81Ch) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_7 is shown in Figure 18-55 and described in Table 18-68.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-55. SPINLOCK_REG_7 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-68. SPINLOCK_REG_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.12 SPINLOCK_REG_8 Register (offset = 820h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_8 is shown in Figure 18-56 and described in Table 18-69.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-56. SPINLOCK_REG_8 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-69. SPINLOCK_REG_8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.13 SPINLOCK_REG_9 Register (offset = 824h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_9 is shown in Figure 18-57 and described in Table 18-70.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-57. SPINLOCK_REG_9 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="7">RESERVED</td><td>TAKEN</td></tr>
<tr><td colspan="7">R-0h</td><td>R/W-0h</td></tr>
</table>

Table 18-70. SPINLOCK_REG_9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.14 SPINLOCK_REG_10 Register (offset = 828h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_10 is shown in Figure 18-58 and described in Table 18-71.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-58. SPINLOCK_REG_10 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-71. SPINLOCK_REG_10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.15 SPINLOCK_REG_11 Register (offset = 82Ch) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_11 is shown in Figure 18-59 and described in Table 18-72.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-59. SPINLOCK_REG_11 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="7">RESERVED</td><td>TAKEN</td></tr>
<tr><td colspan="7">R-0h</td><td>R/W-0h</td></tr>
</table>

Table 18-72. SPINLOCK_REG_11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.16 SPINLOCK_REG_12 Register (offset = 830h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_12 is shown in Figure 18-60 and described in Table 18-73.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-60. SPINLOCK_REG_12 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-73. SPINLOCK_REG_12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.17 SPINLOCK_REG_13 Register (offset = 834h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_13 is shown in Figure 18-61 and described in Table 18-74.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-61. SPINLOCK_REG_13 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-74. SPINLOCK_REG_13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.18 SPINLOCK_REG_14 Register (offset = 838h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_14 is shown in Figure 18-62 and described in Table 18-75.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-62. SPINLOCK_REG_14 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-75. SPINLOCK_REG_14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.19 SPINLOCK_REG_15 Register (offset = 83Ch) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_15 is shown in Figure 18-63 and described in Table 18-76.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-63. SPINLOCK_REG_15 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-76. SPINLOCK_REG_15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.20 SPINLOCK_REG_16 Register (offset = 840h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_16 is shown in Figure 18-64 and described in Table 18-77.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-64. SPINLOCK_REG_16 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-77. SPINLOCK_REG_16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.21 SPINLOCK_REG_17 Register (offset = 844h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_17 is shown in Figure 18-65 and described in Table 18-78.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-65. SPINLOCK_REG_17 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-78. SPINLOCK_REG_17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.22 SPINLOCK_REG_18 Register (offset = 848h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_18 is shown in Figure 18-66 and described in Table 18-79.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-66. SPINLOCK_REG_18 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-79. SPINLOCK_REG_18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.23 SPINLOCK_REG_19 Register (offset = 84Ch) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_19 is shown in Figure 18-67 and described in Table 18-80.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-67. SPINLOCK_REG_19 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-80. SPINLOCK_REG_19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.24 SPINLOCK_REG_20 Register (offset = 850h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_20 is shown in Figure 18-68 and described in Table 18-81.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-68. SPINLOCK_REG_20 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-81. SPINLOCK_REG_20 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.25 SPINLOCK_REG_21 Register (offset = 854h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_21 is shown in Figure 18-69 and described in Table 18-82.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-69. SPINLOCK_REG_21 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-82. SPINLOCK_REG_21 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.26 SPINLOCK_REG_22 Register (offset = 858h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_22 is shown in Figure 18-70 and described in Table 18-83.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-70. SPINLOCK_REG_22 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-83. SPINLOCK_REG_22 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.27 SPINLOCK_REG_23 Register (offset = 85Ch) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_23 is shown in Figure 18-71 and described in Table 18-84.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-71. SPINLOCK_REG_23 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-84. SPINLOCK_REG_23 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.28 SPINLOCK_REG_24 Register (offset = 860h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_24 is shown in Figure 18-72 and described in Table 18-85.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-72. SPINLOCK_REG_24 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-85. SPINLOCK_REG_24 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.29 SPINLOCK_REG_25 Register (offset = 864h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_25 is shown in Figure 18-73 and described in Table 18-86.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-73. SPINLOCK_REG_25 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-86. SPINLOCK_REG_25 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.30 SPINLOCK_REG_26 Register (offset = 868h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_26 is shown in Figure 18-74 and described in Table 18-87.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-74. SPINLOCK_REG_26 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-87. SPINLOCK_REG_26 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.31 SPINLOCK_REG_27 Register (offset = 86Ch) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_27 is shown in Figure 18-75 and described in Table 18-88.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-75. SPINLOCK_REG_27 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-88. SPINLOCK_REG_27 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.32 SPINLOCK_REG_28 Register (offset = 870h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_28 is shown in Figure 18-76 and described in Table 18-89.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-76. SPINLOCK_REG_28 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-89. SPINLOCK_REG_28 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.33 SPINLOCK_REG_29 Register (offset = 874h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_29 is shown in Figure 18-77 and described in Table 18-90.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-77. SPINLOCK_REG_29 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-90. SPINLOCK_REG_29 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.34 SPINLOCK_REG_30 Register (offset = 878h) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_30 is shown in Figure 18-78 and described in Table 18-91.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-78. SPINLOCK_REG_30 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-91. SPINLOCK_REG_30 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

18.2.1.35 SPINLOCK_REG_31 Register (offset = 87Ch) [reset = 0h]

Register mask: FFFFFFFFh

SPINLOCK_REG_31 is shown in Figure 18-79 and described in Table 18-92.

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

Figure 18-79. SPINLOCK_REG_31 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							TAKEN
R-0h							R/W-0h

Table 18-92. SPINLOCK_REG_31 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TAKEN	R/W	0h	

Timers

This chapter describes the timers for the device.

Topic **Page**

19.1 DMTimer

19.1.1 Introduction

The timer module contains a free running upward counter with auto reload capability on overflow. The timer counter can be read and written in real-time (while counting). The timer module includes compare logic to allow an interrupt event on a programmable counter matching value.

A dedicated output signal can be pulsed or toggled on overflow and match event. This output offers a timing stamp trigger signal or PWM (pulse-width modulation) signal sources. A dedicated output signal can be used for general purpose PORGPOCFG (directly driven by bit 14 of the TCLR register). A dedicated input signal is used to trigger automatic timer counter capture and interrupt event, on programmable input signal transition type. A programmable clock divider (prescaler) allows reduction of the timer input clock frequency. All internal timer interrupt sources are merged in one module interrupt line and one wake-up line. Each internal interrupt source can be independently enabled/disabled.

This module is controllable through the OCP peripheral bus.

As two clock domains are managed inside this module, resynchronization is done by special logic between the OCP clock domain and the Timer clock domain. At reset, synchronization logic allows utilization of all ratios between the OCP clock and the Timer clock. A drawback of this mode is that full-resynchronization path is used with access latency performance impact in terms of OCP clock cycles. In order to improve module access latency, and under restricted conditions on clocks ratios, write-posted mode can be used by setting the POSTED bit of the System Control Register (TSCR). Under this mode, write posted mode is enabled, meaning that OCP write command is granted before the write process completes in the timer clock domain. This mode allows software to do concurrent writes on Dual Mode timer registers and to observe write process completion (synchronization) at the software level by reading independent write posted status bits in the Write Posted Status Register (TWPS).

19.1.1.1 Features

The timer consists of the following features:

- Counter timer with compare and capture modes
- Auto-reload mode
- Start-stop mode
- Programmable divider clock source
- 16-32 bit addressing
- “On the fly” read/write registers
- Interrupts generated on overflow, compare and capture
- Interrupt enable
- Wake-up enable (only for Timer0)
- Write posted mode
- Dedicated input trigger for capture mode and dedicated output trigger/PWM signal
- Dedicated output signal for general purpose use PORGPOCFG
- OCP interface compatible

The Timer resolution and interrupt period are dependent on the selected input clock and clock prescaler value. Example resolutions for common clock values are shown in Table 19-1.

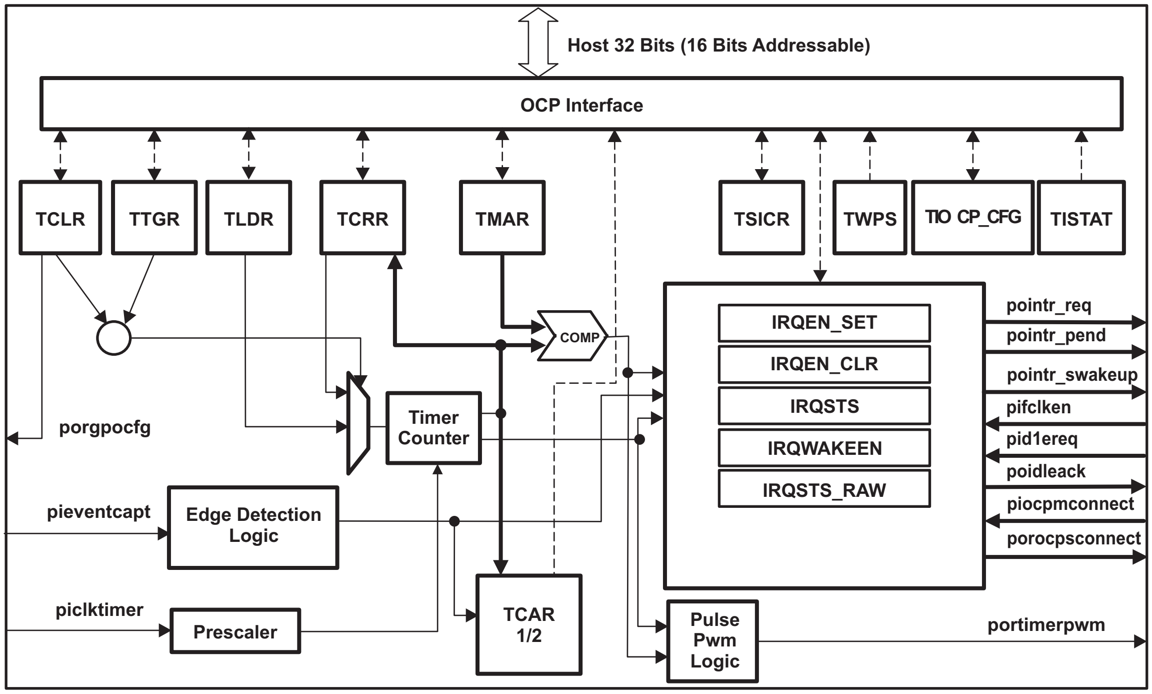
Table 19-1. Timer Resolution and Maximum Range

Clock	Prescaler	Resolution	Interrupt Period Range
32.768 KHz	1 (min)	31.25 us	31.25 us to ~36h 35m
	256 (max)	8 ms	8 ms to ~391d 22h 48m
25 MHz	1 (min)	40 ns	40 ns to ~171.8s
	256 (max)	10.24 us	~20.5 us to ~24h 32m

19.1.1.2 Functional Block Diagram

Figure 19-1 shows a block diagram of the timer.

Figure 19-1. Timer Block Diagram



19.1.2 Integration

The integration for each timer is shown in the following figures:

- Timer0: Figure 19-2
- Timer2–7: Figure 19-3
- Timer8–11: Figure 19-4

Figure 19-2. Timer0 Integration

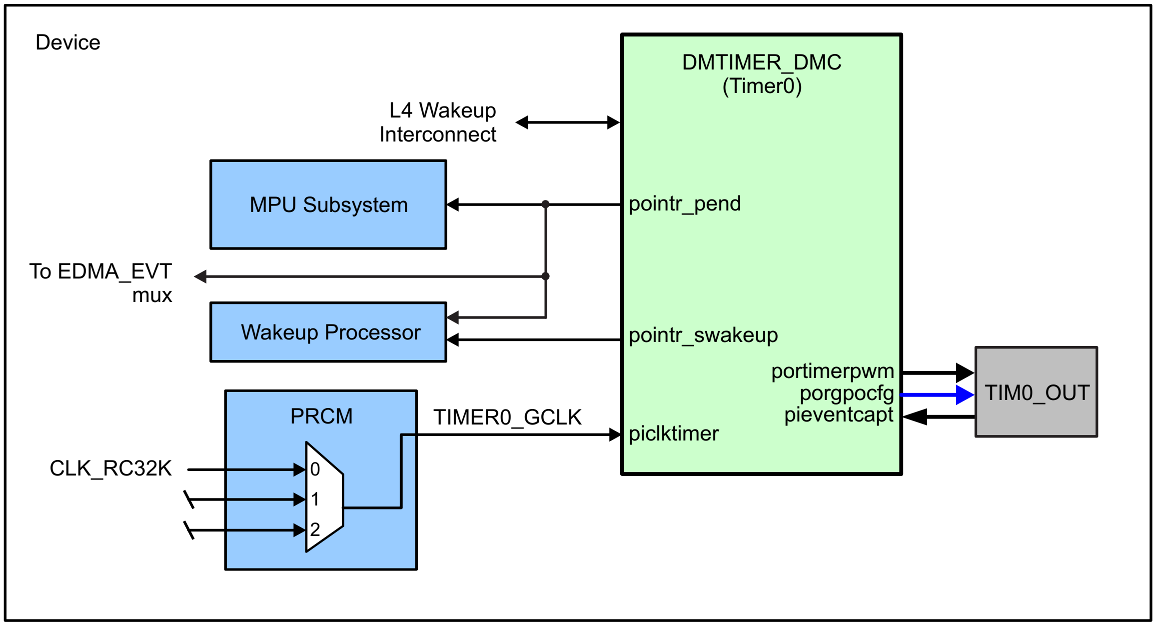


Figure 19-3. Timer2-7 Integration

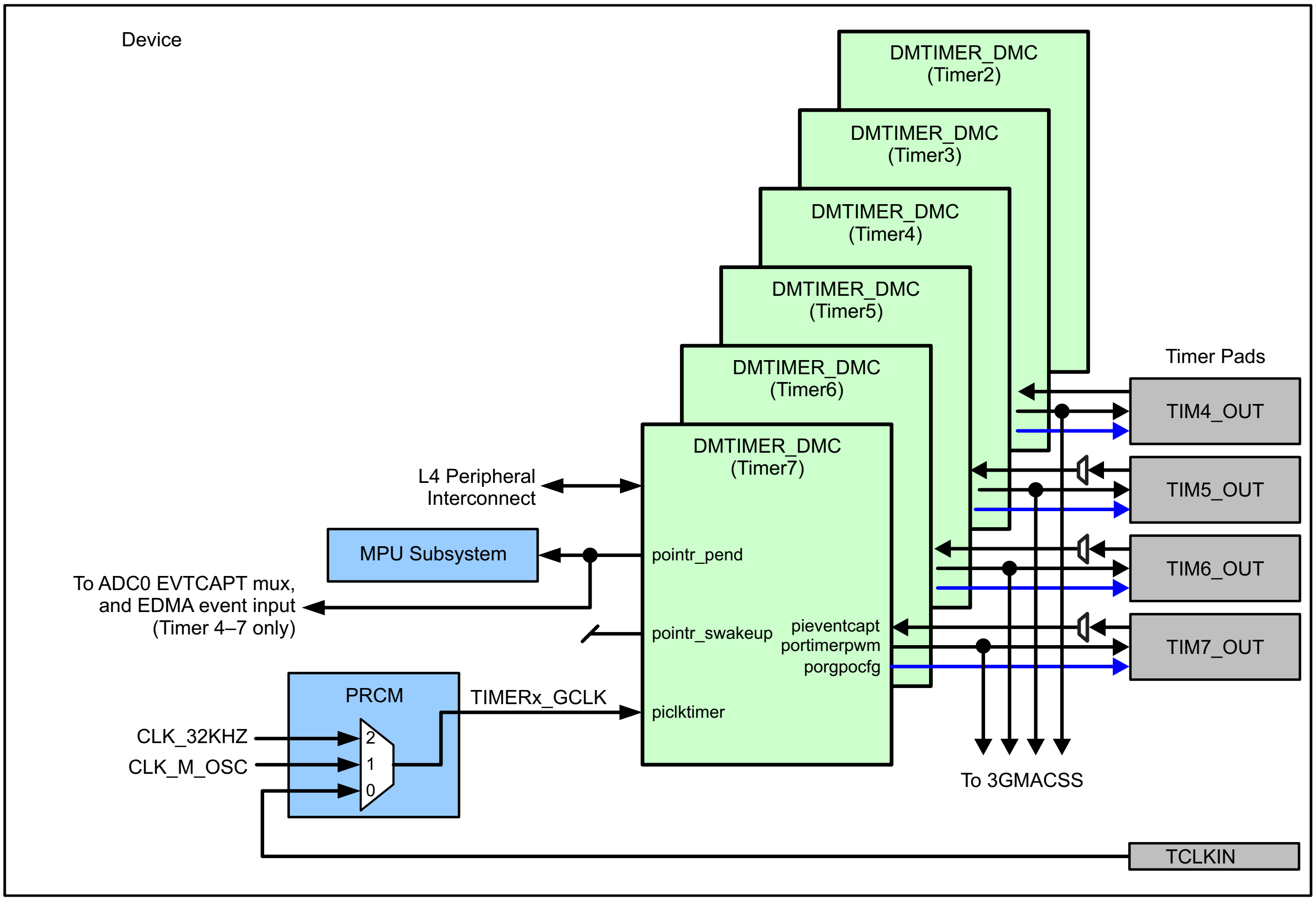
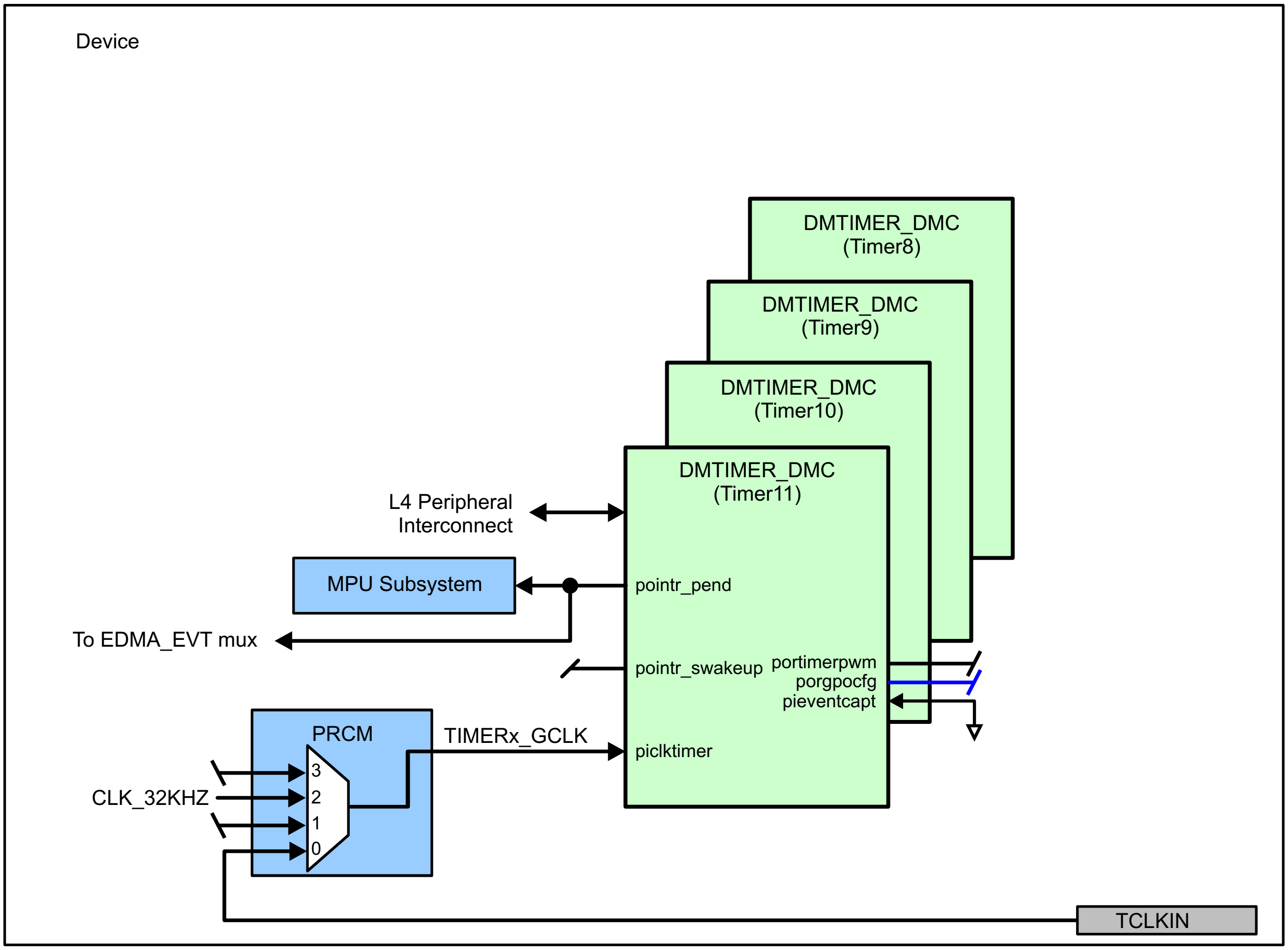


Figure 19-4. Timer8-11 Integration



19.1.2.1 Timer Connectivity Attributes

Table 19-2. Timer[0] Connectivity Attributes

Attributes	Type
Power domain	Wakeup domain
Clock Domain	PD_WKUP_L4_WKUP_GCLK (Interface/OCP) PD_WKUP_TIMER0_GCLK (Func)
Reset Signals	WKUP_DOM_RST_N
Idle/Wakeup Signals	Smart Idle Slave Wakeup
Interrupt Requests	1 to MPU Subsystem (TINT0), Wakeup Processor, EDMA SWAKEUP to Wakeup Processor
DMA Requests	None
Physical Address	L4 Wakeup slave port

Table 19-3. Timer[2–11] Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L4LS_GCLK (OCP) Functional Clocks: PD_PER_TIMER2_GCLK (Timer 2) PD_PER_TIMER3_GCLK (Timer 3) PD_PER_TIMER4_GCLK (Timer 4) PD_PER_TIMER5_GCLK (Timer 5) PD_PER_TIMER6_GCLK (Timer 6) PD_PER_TIMER7_GCLK (Timer 7) PD_PER_TIMER8_GCLK (Timer 8) PD_PER_TIMER9_GCLK (Timer 9) PD_PER_TIMER10_GCLK (Timer 10) PD_PER_TIMER11_GCLK (Timer 11)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle (No wakeup capabilities)
Interrupt Requests	1 per timer module to MPU Subsystem (TINT2 - TINT7) Also to EDMA for Timer 2–Timer 7 Also to TSC_ADC event capture mux for Timer 4–Timer7
DMA Requests	Interrupt requests are redirected as DMA requests: 1 per instance (TINTx)
Physical Address	L4 Peripheral slave port

19.1.2.2 Timer Clock and Reset Management

The DMTimer0 functional clock is sourced from the on-chip ~32.768 kHz oscillator (CLK_RC32K).

Each DMTimer[2–7] functional clock is selected within the PRCM using the associated CLKSEL_TIMERx_CLK register from the following possible sources:

- The 24-MHz (typ) system clock (CLK_M_OSC)
- The PER PLL generated 32.768 KHz clock (CLK_32KHZ)
- The TCLKIN external timer input clock.

Each DMTimer[8-11] functional clock is selected within the PRCM using the associated CLKSEL_TIMERx_CLK register from the following possible sources:

- The 24-MHz (typ) system clock (CLK_M_OSC)
- The PER PLL generated 32.768 KHz clock (CLK_32KHZ)
- The TCLKIN external timer input clock

19.1.2.3 Timer Clock Signals

Table 19-4. Timer[0, 2-7] Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
		Timer[0] Clock Signals	
PICLKOCP Interface clock	26 MHz	CLK_M_OSC	pd_wkup_l4_wkup_gclk from PRCM
PICLKTIMER Functional clock	26 MHz	CLK_RC32K	pd_wkup_timer0_gclk from PRCM
		Timer[2–7] Clock Signals	
PICLKOCP Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk from PRCM
PICLKTIMER Functional clock	26 MHz	CLK_M_OSC CLK_32KHZ (PER_CLKOUTM2 / 5859.375) TCLKIN	pd_per_timer2_gclk pd_per_timer3_gclk pd_per_timer4_gclk pd_per_timer5_gclk pd_per_timer6_gclk pd_per_timer7_gclk from PRCM

Table 19-5. Timer[8-11] Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
PICLKOCP Interface clock	100 MHz	CORE_CLKOUTM4/2	pd_perl4ls_gclk from PRCM
PICLKTIMER Functional clock	26 MHz	CLK_M_OSC CLK_32KHZ (PER_CLKOUTM2 / 5859.375) TCLKIN	pd_per_timer8_gclk pd_per_timer9_gclk pd_per_timer10_gclk pd_per_timer11_gclk

19.1.2.4 Timer Pin List

The timer PIEVENTCAPT input and PORTIMERPWM output signals are muxed onto a single TIMER I/O pad. The pad direction (and hence the pin function) are controlled from within the DMTimer module using the PORGPOCFG signal as an output enable.

Table 19-6. Timer Pin List

Pin	Type	Description
TCLKIN	I	External timer clock source
TIMER0	I/O	Timer 0 trigger input or PWM output
TIMER4	I/O	Timer 4 trigger input or PWM output
TIMER5	I/O	Timer 5 trigger input or PWM output
TIMER6	I/O	Timer 6 trigger input or PWM output
TIMER7	I/O	Timer 7 trigger input or PWM output

19.1.3 Functional Description

The general-purpose timer is an upward counter. It supports 3 functional modes:

- Timer mode
- Capture mode
- Compare mode

By default, after core reset, the capture and compare modes are disabled.

19.1.3.1 Timer Mode Functionality

The timer is an upward counter that can be started and stopped at any time through the Timer Control Register (TCLR ST bit). The Timer Counter Register (TCRR) can be loaded when stopped or on the fly (while counting). TCRR can be loaded directly by a TCRR Write access with the new timer value. TCRR can also be loaded with the value held in the Timer Load Register (TLDR) by a trigger register (TTGR) Write access. The TCRR loading is done regardless of the value written to TTGR. The value of the timer counter register (TCRR) can be read when stopped or captured on the fly by a TCRR Read access. The timer is stopped and the counter value is cleared to “0” when the module’s reset is asserted. The timer is maintained in stop after reset is released. When the timer is stopped, TCRR is frozen. The timer can be restarted from the frozen value unless TCRR has been reloaded with a new value.

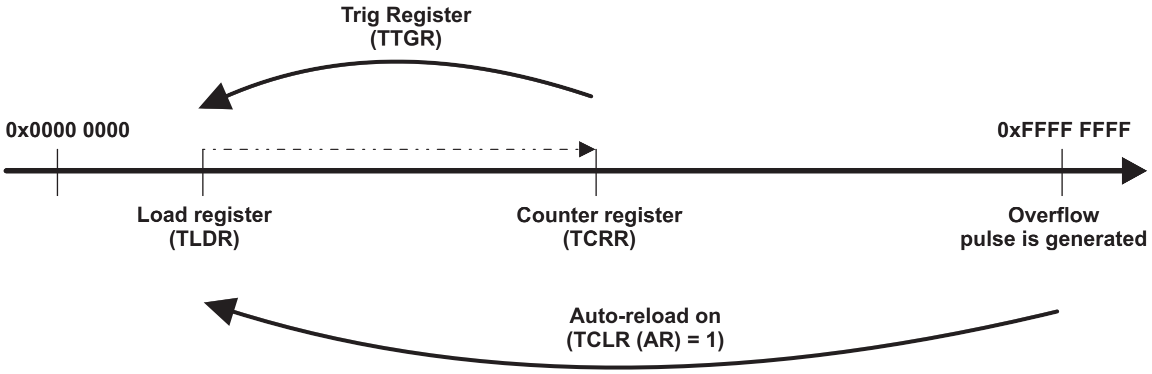
In the one shot mode (TCLR AR bit = 0), the counter is stopped after counting overflow (counter value remains at zero).

When the auto-reload mode is enabled (TCLR AR bit = 1), the TCRR is reloaded with the Timer Load Register (TLDR) value after a counting overflow.

It is not recommended to put the overflow value (FFFF FFFFh) in TLDR because it can lead to undesired results.

An interrupt can be issued on overflow if the overflow interrupt enable bit is set in the timer Interrupt Enable Register (IRQEN_SET OVF_IT_FLAG bit = 1). A dedicated output pin (PORTIMERPWM) is programmed through TCLR (TRG and PT bits) to generate one positive pulse (prescaler duration) or to invert the current value (toggle mode) when an overflow occurs.

Figure 19-5. TCRR Timing Value



19.1.3.2 Capture Mode Functionality

The timer value in TCRR can be captured and saved in TCAR1 or TCAR2 function of the mode selected in TCLR through the field CAPT_MODE when a transition is detected on the module input pin (PIEVENTCAPT). The edge detection circuitry monitors transitions on the input pin (PIEVENTCAPT).

Rising transition, falling transition or both can be selected in TCLR (TCM bit) to trigger the timer counter capture. The module sets the IRQSTS (TCAR_IT_FLAG bit) when an active transition is detected and at the same time the counter value TCRR is stored in one of the timer capture registers TCAR1 or TCAR2 as follows:

- If TCLR's CAPT_MODE field is 0 then, on the first enabled capture event, the value of the counter register is saved in TCAR1 register and all the next events are ignored (no update on TCAR1 and no interrupt triggering) until the detection logic is reset or the interrupt status register is cleared on TCAR's position writing a 1 in it.
- If TCLR's CAPT_MODE field is 1 then, on the first enabled captured event, the counter value is saved in TCAR1 register and, on the second enabled capture event, the value of the counter register is saved in TCAR2 register. All the other events are ignored (no update on TCAR1/2 and no interrupt triggering) until the detection logic is reset or the interrupt status register is cleared on TCAR's position writing a 1 in it. This mechanism is useful for period calculation of a clock if that clock is connected to the PIEVENTCAPT input pin.

The edge detection logic is reset (a new capture is enabled) when the active capture interrupt is served. The TCAR_IT_FLAG bit of IRQSTS (previously 1) is cleared. The timer functional clock (input to prescaler) is used to sample the input pin (PIEVENTCAPT). Negative or positive pulses can be detected when the pulse time exceeds the functional clock period. An interrupt can be issued on transition detection if the capture interrupt enable bit is set in the Timer Interrupt Enable Register IRQEN_SET (TCAR_IT_FLAG bit).

In Figure 19-6, the TCM value is 01 and CAPT_MODE is 0 - only rising edge of the PIEVENTCAPT will trigger a capture in TCAR and only TCAR1 will update.

In Figure 19-7, the TCM value is 01 and CAPT_MODE is 1 - only rising edge of the PIEVENTCAPT will trigger a capture in TCAR1 on first enabled event and TCAR2 will update on the second enabled event.

Figure 19-6. Capture Wave Example for CAPT_MODE = 0

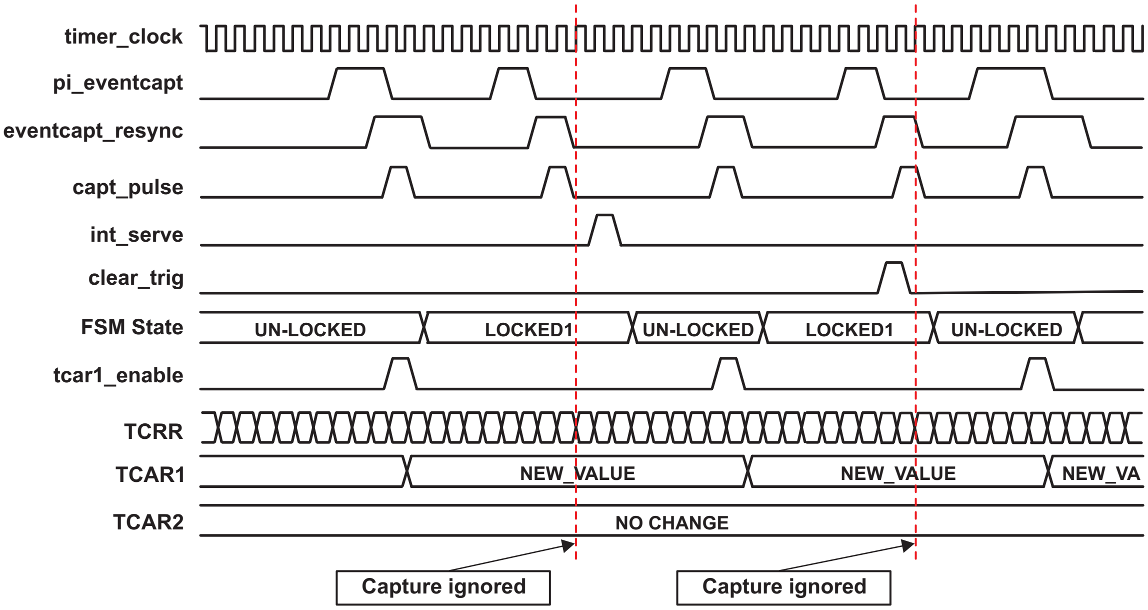
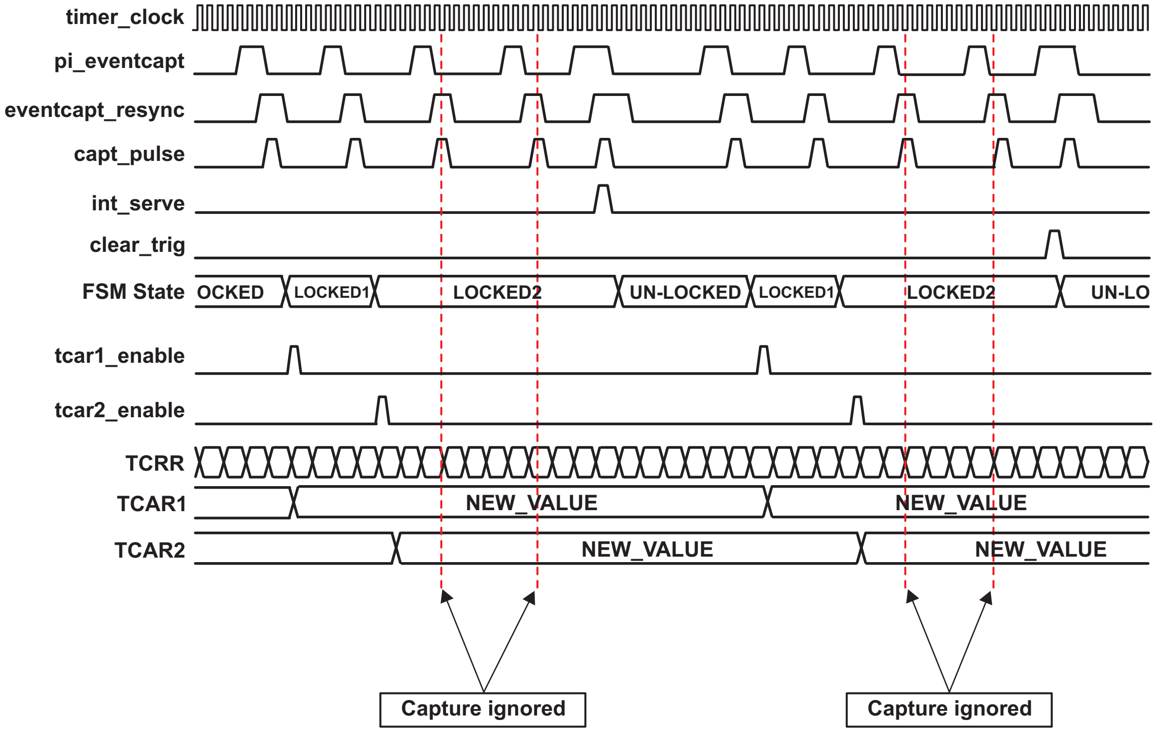


Figure 19-7. Capture Wave Example for CAPT_MODE = 1



19.1.3.3 Compare Mode Functionality

When Compare Enable TCLR (CE bit) is set to 1, the timer value (TCRR) is permanently compared to the value held in timer match register (TMAR). TMAR value can be loaded at any time (timer counting or stop). When the TCRR and the TMAR values match, an interrupt can be issued if the IRQEN_SET (MAT_EN_FLAG bit) is set. The correct implementation is to write a compare value in TMAR register before setting TCLR (CE bit) to avoid any unwanted interrupts due to a reset value matching effect.

The dedicated output pin (PORTIMERPWM) can be programmed through TCLR (TRG and PT bits) to generate one positive pulse (TIMER clock duration) or to invert the current value (toggle mode) when an overflow and a match occur.

19.1.3.4 Prescaler Functionality

A prescaler counter can be used to divide the timer counter input clock frequency. The prescaler is enabled when TCLR bit 5 is set (PRE). The 2n division ratio value (PTV) can be configured in the TCLR register. The prescaler counter is reset when the timer counter is stopped or reloaded on the fly.

Table 19-7. Prescaler Functionality

Contexts	Prescaler Counter	Timer Counter
Overflow (when Auto-reload on)	Reset	TLDR
TCRR Write	Reset	TCRR
TTGR Write	Reset	TLDR
Stop	Reset	Frozen

19.1.3.5 Pulse-Width Modulation

The timer can be configured to provide a programmable pulse-width modulation (PORTIMERPWM) output. The PORTIMERPWM output pin can be configured to toggle on a specified event. TCLR (TRG bits) determines on which register value the PORTIMERPWM pin toggles. Either overflow or match can be used to toggle the PORTIMERPWM pin, when a compare condition occurs.

In case of overflow and match mode, the match event will be ignored from the moment the mode was set-up until the first overflow event occurs (see Figure 19-7).

The TCLR (SCPWM bit) can be programmed to set or clear the PORTIMERPWM output signal while the counter is stopped or the triggering is off only. This allows fixing a deterministic state of the output pin when modulation is stopped. The modulation is synchronously stopped when the TRG bit is cleared and an overflow has occurred.

In the following timing diagram, the internal overflow pulse is set each time (FFFF FFFFFh – TLDR + 1) value is reached, and the internal match pulse is set when the counter reaches TMAR register value. According to TCLR (TRG and PT bits) programming value, the timer provides pulse or PWM on the output pin (PORTIMERPWM).

The TLDR and TMAR registers must keep values smaller than the overflow value (FFFF FFFFh) with at least 2 units. In case the PWM trigger events are both overflow and match, the difference between the values kept in TMAR register and the value in TLDR must be at least 2 units. When match event is used, the compare mode TCLR (CE) must be set.

In Figure 19-8, TCLR (SCPWM bit) is cleared to 0. In Figure 19-9, TCLR (SCPWM bit) is set to 1.

Figure 19-8. Timing Diagram of Pulse-Width Modulation with SCPWM = 0

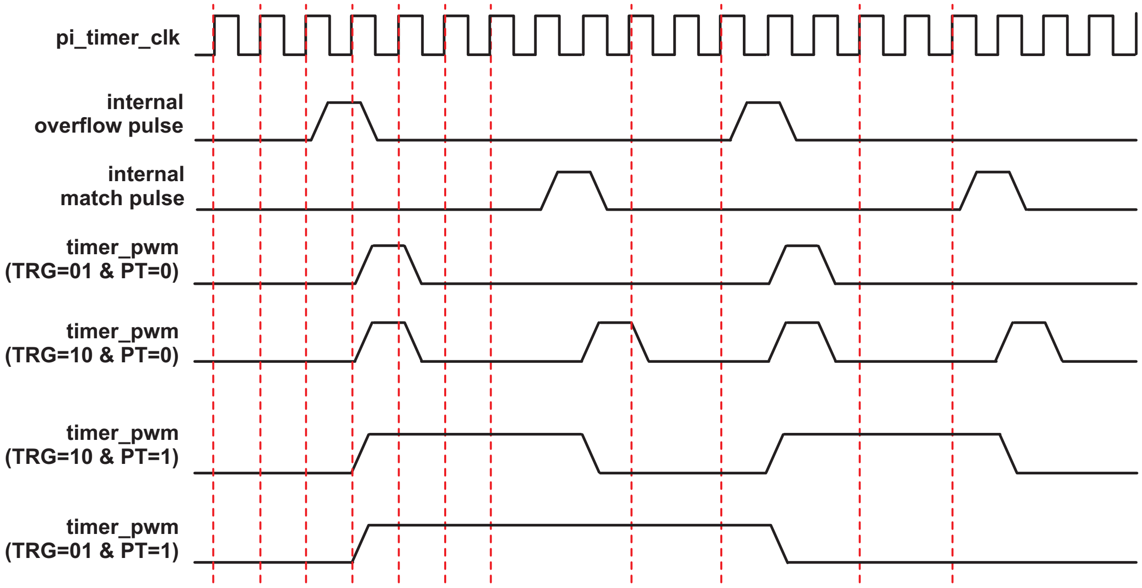
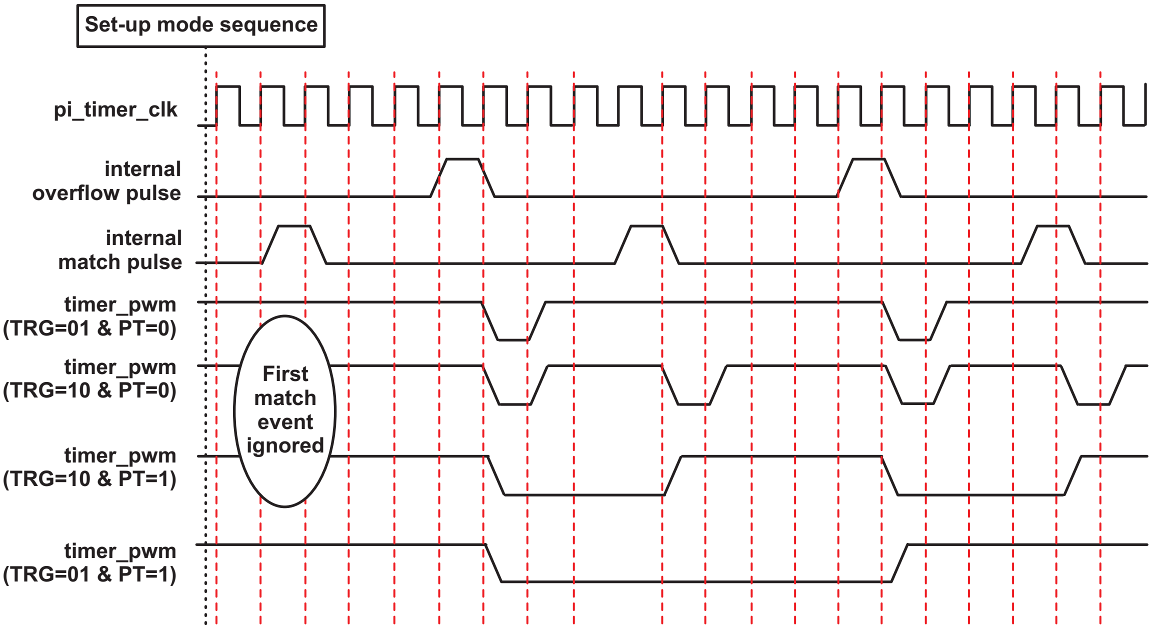


Figure 19-9. Timing Diagram of Pulse-Width Modulation with SCPWM = 1



19.1.3.6 Timer Counting Rate

The timer counter is composed of a prescaler stage and a timer counter. Prescaler stage is clocked with the timer clock and acts as a clock divider for the timer counter stage. The ratio can be managed by accessing the ratio definition field of the control register (PTV and PRE of TCLR). See Table 19-8.

The timer rate is defined by:

- The value of the prescaler fields (PRE and PTV of TCLR register)
- The value loaded into the Timer Load Register (TLDR).

Table 19-8. Prescaler Clock Ratios Value

PRE	PTV	Divisor (PS)
0	X	1
1	0	2
1	1	4
1	2	8
1	3	16
1	4	32
1	5	64
1	6	128
1	7	256

The timer rate equation is as follows:

(FFFF FFFFh – TLDR + 1) × timer Clock period × Clock Divider (PS)

With timer Clock period = 1/ timer Clock frequency and PS = 2(PTV + 1).

As an example, if we consider a timer clock input of 32 kHz, with a PRE field equal to 0, the timer output period is:

Table 19-9. Value and Corresponding Interrupt Period

TLDR	Interrupt period
0000 0000h	37 h
FFFF 0000h	2 s
FFFF FFF0h	500 us
FFFF FFFEh	62.5 us

19.1.3.7 Dual Mode Timer Under Emulation

To configure the Timer to stop during emulation suspend events (for example, debugger breakpoints), set up the Timer and the Debug Subsystem:

1. Set TIOCP_CFG.EMUFREE=0. This will allow the Suspend_Control signal from the Debug Subsystem (Chapter 31) to stop and start the Timer. Note that if EMUFREE=1, the Suspend_Control signal is ignored and the Timer is free running regardless of any debug suspend event. This EMUFREE bit gives local control from a module perspective to gate the suspend signal coming from the Debug Subsystem.
2. Set the appropriate xxx_Suspend_Control register = 0x9, as described in Section 31.1.1.1, *Debug Suspend Support for Peripherals*. Choose the register appropriate to the peripheral you want to suspend during a suspend event.

19.1.3.8 Accessing Registers

All registers are 32-bit wide, accessible via OCP interface with 16-bit or 32-bit OCP access (Read/Write). The 32-bit registers write update in 16 bits access must be LSB16 first and the second write access must be MSB16. For the write operation, the module allows skipping the MSB access if the user does not need to update the 16 MSB bits of the register, but only for the OCP registers (TIDR, TIOCP_CFG, IRQSTS_RAW, IRQSTS, IRQEN_SET, IRQEN_CLR, IRQWAKEEN and TSICR). The write operation on any functional register (TCLR, TCRR, TLDR, TTGR and TMAR) must be complete (the MSB must be written even if the MSB data is not used).

19.1.3.8.1 Programming the Timer Registers

The TLDR, TCRR, TCLR, TIOCP_CFG, IRQSTS, IRQEN_SET, IRQEN_CLR, IRQWAKEEN, TTGR, TSICR and TMAR registers write is done synchronously with OCP clock, by the host, using the OCP bus protocol.

19.1.3.8.2 Reading the Timer Registers

The counter register (TCRR) is a 32-bit “atomic datum” and 16-bit capture is done on the 16-bit LSB first to allow atomic LSB16 + MSB16 capture. Atomic capture is also performed for the TCAR1 and TCAR2 registers as they may change due to internal processes.

19.1.3.8.3 OCP Error Generation

The timer module responds with error indication in the following cases:

Error on write transactions

- Assert the PORSRESP = ERR signal in the same cycle as PORSCMDACCEPTED.
- Use the ERR code for PORSRESP during the response phase.

Error on read transactions

- Assert the PORSRESP = ERR signal in the same cycle as PORSCMDACCEPTED.
- Use the ERR code for PORSRESP during the response phase. PORSDATA in this case is not valid.

Table 19-10. OCP Error Reporting

Error Type	Response: SRESP = ERR
Unsupported PIOCPMCMD command	Yes
Address error: Read or write to a non-existing internal address	No
Read to write-only registers and write to read-only registers	No
Unaligned address (PIOCPMADDR ≠ 00) on read/write transaction	Yes
Unsupported PIOCPMBYTEEN on read/write transaction	Yes

NOTE: Byte enable “0000” is a supported byte enable.

19.1.3.9 Posted Mode Selection

A choice between the two synchronization modes will be made taking into account the frequency ratio and the stall periods that can be supported by the system, without impacting the global performance.

The posted mode selection applies only to functional registers that require synchronization on/from timer clock domain. For write operation, the registers affected by this posted/non-posted selection are: TCLR, TLDR, TCRR, TTGR and TMAR. For read operation, the register affected by this posted/non-posted selection are: TCRR, TCAR1 and TCAR2.

The OCP clock domain synchronous registers TIDR, TIOCP_CFG, TISTAT, IRQSTS, IRQSTS_RAW, IRQEN_SET, IRQEN_CLR, IRQWAKEEN, TWPS and TSICR are not affected by the posted/non-posted mode selection; the write/read operation is effective and acknowledged (command accepted) after one OCP clock cycle from the command assertion.

19.1.3.10 Write Registers Access

19.1.3.10.1 Write Posted

This mode can be used only if the functional frequency range is freq (timer) < freq (OCP)/4.

This mode is used if TSICR (POSTED bit) is set to 1 in the timer control register.

This mode uses a posted-write scheme to update any internal register. The write transaction is immediately acknowledged on the OCP interface, although the effective write operation will occur later, due to a resynchronisation in the timer clock domain. This has the advantage of not stalling either the interconnect system, or the CPU that requested the write transaction. For each register, a status bit is provided, that is set if there is a pending write access to this register.

In this mode, it is mandatory that the CPU checks the status bit prior to any write access. In case a write is attempted to a register with a previous access pending, the previous access is discarded without notice (this can lead to unexpected results also).

There is one status bit per register, accessible in the Timer Write Posted Status Register. When the timer module operates in this mode, there is an automatic sampling of the current timer counter value, in an OCP-synchronized capture register. Consequently, any read access to the timer counter register does not add any re-synchronization latency; the current value is always available.

A register read following a write posted register (on the same register) is not ensured to read the previous write value if the write posted process is not completed. Software synchronization should be used to avoid a non-coherent read.

The drawback of this automatic update mechanism is that it assumes a given relationship between the OCP interface frequency and the timer functional frequency.

This posted period is defined as the interval between the posted write access request and the reset of the posted bit in TWPS register, and can be quantified:

T (reset posted max.) = 3 OCP clock + 2.5 TIMER clock

The time when the write accomplishes is:

T (write accomplish) = 1 OCP clock + 2.5 TIMER clock

19.1.3.10.2 Write Non-Posted

This mode is functional regardless of the ratio between the OCP interface frequency and the functional clock frequency. The recommended functional frequency range is freq (timer) >= freq (OCP)/4.

This mode is used if TSICR (POSTED bit) is cleared to 0 in the timer control register.

This mode uses a non posted-write scheme to update any internal register. That means the write transaction will not be acknowledged on the OCP interface, until the effective write operation occurs, after the resynchronisation in the timer clock domain. The drawback is that both the interconnect system and the CPU are stalled during this period.

- The latency of the interrupt serving is increased, as the interconnect system and CPU are stalled.
- An interconnect logic, including time-out logic to detect erroneous transactions, can generate an unwanted system abort event.

The stall period is defined as the interval between the non-posted write access request and the rise of the command accept signal and can be quantified:

T (stall max.) = 3 OCP clock + 2.5 TIMER clock

The time when the write accomplishes is:

T (write accomplish) = 1 OCP clock + 2.5 TIMER clock

A register read following a write to the same register is always coherent.

19.1.3.11 Read Registers Access

19.1.3.11.1 Read Posted

This mode can be used only if the functional frequency range is freq (timer) < freq (OCP)/4.

This mode is used if TSICR (POSTED bit) is set to 1 in the timer control register.

This mode uses a posted-read scheme, for reading any internal register. The read transaction is immediately acknowledged on the OCP interface, and the value to be read has been previously resynchronised. This has the advantage of not stalling either the interconnect system, or the CPU that requested the read transaction.

19.1.3.11.2 Read Non-Posted

This mode is functional whatever the ratio between the OCP interface frequency and the functional clock frequency. Recommended functional frequency range is freq (timer) >= freq (OCP)/4.

This mode is used if TSICR (POSTED bit) is cleared to 0 in the timer control register.

This mode uses a non posted-read scheme, for reading any internal register. The read transaction will not be acknowledged on the OCP interface, until the effective read operation occurs, after the resynchronisation in the timer clock domain. The drawback is that both the interconnect system and the CPU are stalled during this period.

- The latency of the interrupt serving is increased, as the interconnect system and the CPU are stalled.
- An interconnect system including time-out logic to detect erroneous transactions can generate an unwanted system abort event.

This mode applies only to three registers: TCRR, TCAR1 and TCAR2, which need resynchronisation from functional to OCP clock domains.

The stall period is defined as the interval between the non-posted read access request and the rise of the command accept signal and can be quantified:

T (stall max.) = 3 OCP clock + 2.5 TIMER clock

The time when the value is sampled is:

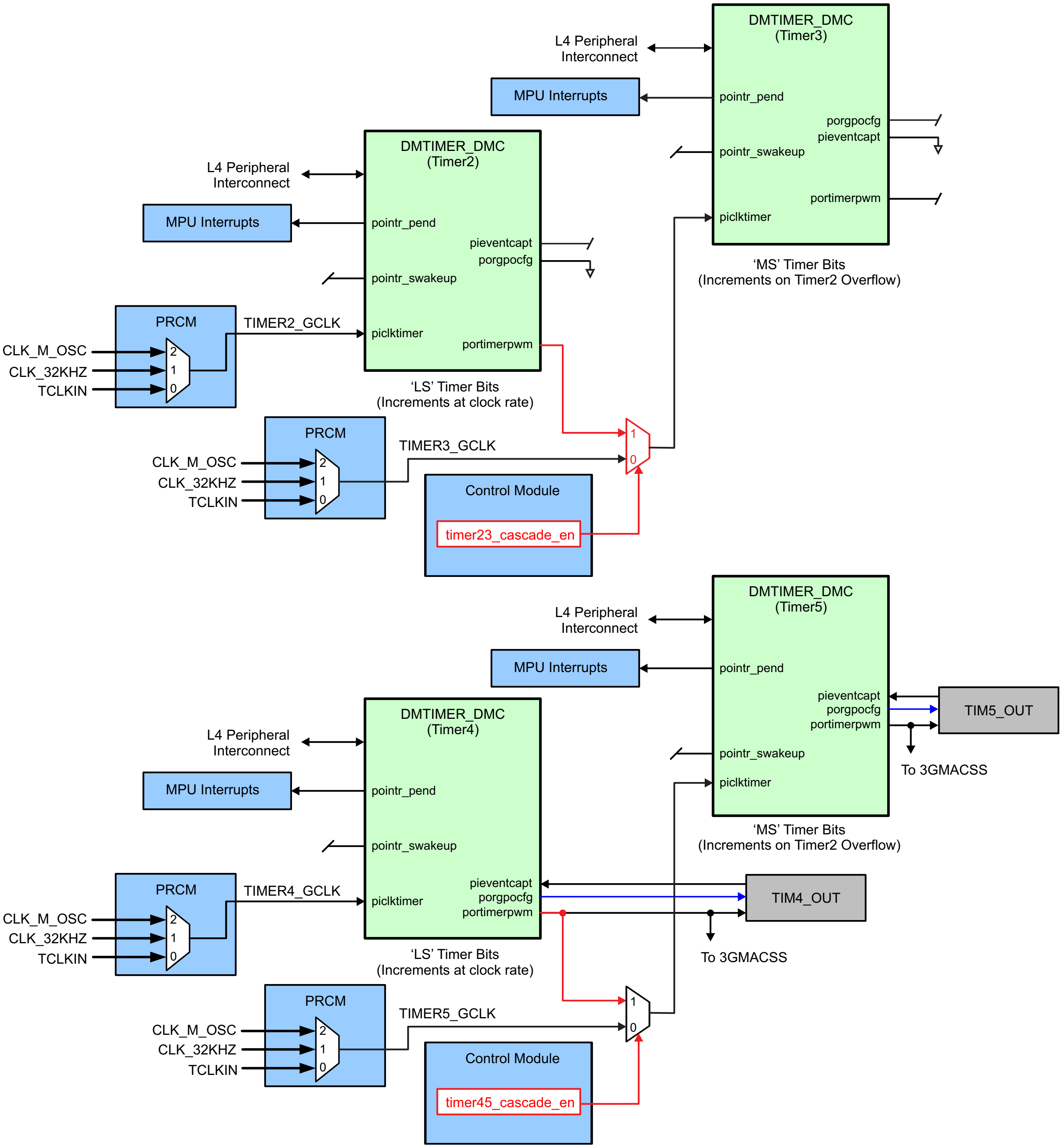
T (read sample) = 1 OCP clock + 2.5 TIMER clock

19.1.3.12 Timer Cascading

In order to provide a 64-bit timer option, the device includes logic to cascade two pairs of Timer modules: DMTIMER2-3 provide a 64-bit internal timer and DMTIMER4-5 provide a 64-bit timer with output capability. The cascade logic consists of an additional mux on the clock input of the second timer in each pair to allow it to be clocked with the output of the first timer in each pair. Cascade mode is selected through the TIMER_CASCADE_CTRL register in the Control Module.

For cascading to work, the LS module must be set to generate only one pulse on the output (PT=0) on overflow (TRG = 01). In this way, the MS Timer counter is incremented each time the LS Timer overflows. (Another configuration option is: PT=1 and TRG = 10).

Figure 19-10. Timer Cascading Details

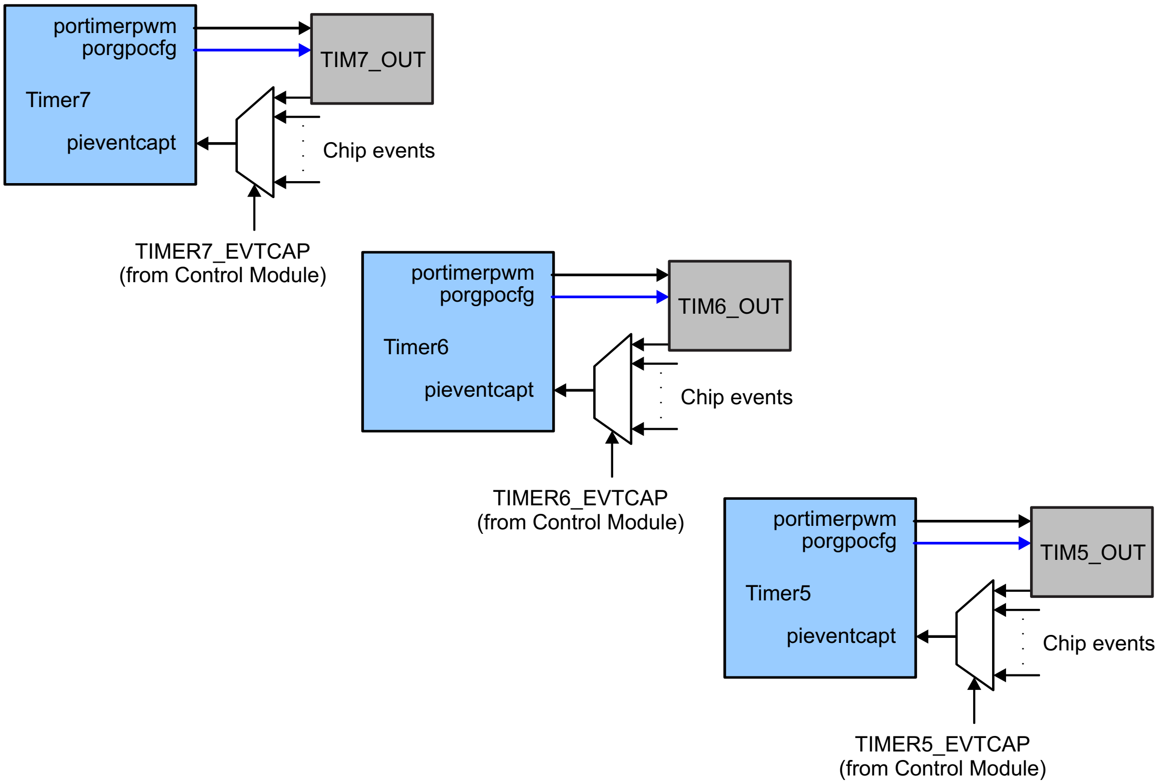


NOTE: Red lines indicate new logic.

19.1.3.13 Timer Synchronization

The timers provide synchronization signals to allow them to be synchronized to other modules or events. This feature is enabled only for Timer0-1 and Timer4-7. On Timer0, Timer1, and Timer4, only the external pin may be used for synchronization. On Timer5-7, the capture events may be selected from among 31 different pins or internal interrupt signals. The event is selected using the corresponding TIMERx_EVTCAPT field of the TIMER_EVT_CAP register in the Control Module.

Figure 19-11. Timer Sync Event Detail



19.1.4 DMTIMER Registers

Table 19-11 lists the memory-mapped registers for the DMTIMER. All register offset addresses not listed in Table 19-11 should be considered as reserved locations and the register contents should not be modified.

Table 19-11. DMTIMER Registers

Offset	Acronym	Register Name	Section
0h	DMTIMER_TIDR		Section 19.1.4.1
10h	DMTIMER_TIOCP_CFG		Section 19.1.4.2
20h	DMTIMER_IRQ_EOI		Section 19.1.4.3
24h	DMTIMER_IRQSTS_RAW		Section 19.1.4.4
28h	DMTIMER_IRQSTS		Section 19.1.4.5
2Ch	DMTIMER_IRQEN_SET		Section 19.1.4.6
30h	DMTIMER_IRQEN_CLR		Section 19.1.4.7
34h	DMTIMER_IRQWAKEEN		Section 19.1.4.8
38h	DMTIMER_TCLR		Section 19.1.4.9
3Ch	DMTIMER_TCRR		Section 19.1.4.10
40h	DMTIMER_TLDR		Section 19.1.4.11
44h	DMTIMER_TTGR		Section 19.1.4.12

Table 19-11. DMTIMER Registers (continued)

Offset	Acronym	Register Name	Section
48h	DMTIMER_TWPS		Section 19.1.4.13
4Ch	DMTIMER_TMAR		Section 19.1.4.14
50h	DMTIMER_TCAR1		Section 19.1.4.15
54h	DMTIMER_TSICR		Section 19.1.4.16
58h	DMTIMER_TCAR2		Section 19.1.4.17

19.1.4.1 DMTIMER_TIDR Register (Offset = 0h) [reset = 4FFF0301h]

Register mask: FFFFFFFFh

DMTIMER_TIDR is shown in Figure 19-12 and described in Table 19-12.

Return to Summary Table.

This read only register contains the revision number of the module. A write to this register has no effect. This Register is used by software to track features, bugs, and compatibility.

Figure 19-12. DMTIMER_TIDR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="2">SCHEME</td><td colspan="2">RESERVED</td><td colspan="4">FUNC</td></tr>
<tr><td colspan="2">R-1h</td><td colspan="2">R-0h</td><td colspan="4">R-FFFh</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">FUNC</td></tr>
<tr><td colspan="8">R-FFFh</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="5">R_RTL</td><td colspan="3">X_MAJOR</td></tr>
<tr><td colspan="5">R-0h</td><td colspan="3">R-3h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CUSTOM</td><td colspan="6">Y_MINOR</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="6">R-1h</td></tr>
</table>

Table 19-12. DMTIMER_TIDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	
29-28	RESERVED	R	0h	Reads return 0
27-16	FUNC	R	FFFh	
15-11	R_RTL	R	0h	
10-8	X_MAJOR	R	3h	
7-6	CUSTOM	R	0h	
5-0	Y_MINOR	R	1h	

19.1.4.2 DMTIMER_TIOCP_CFG Register (Offset = 10h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_TIOCP_CFG is shown in Figure 19-13 and described in Table 19-13.

Return to Summary Table.

This register controls the various parameters of the OCP interface

Figure 19-13. DMTIMER_TIOCP_CFG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				IDLEMODE		EMUFREE	SOFTRESET
R-0h				R/W-0h		R/W-0h	R/W-0h

Table 19-13. DMTIMER_TIOCP_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-2	IDLEMODE	R/W	0h	Power management, req/ack control 0h (R/W) = Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements. Backup mode, for debug only. 1h (R/W) = No-idle mode: local target never enters idle state. Backup mode, for debug only. 2h (R/W) = Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wakeup events. 3h (R/W) = Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state. Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented.
1	EMUFREE	R/W	0h	Sensitivity to emulation (debug) suspend event from Debug Subsystem. 0h (R/W) = The timer is frozen during a debug suspend event. 1h (R/W) = The timer runs free. Debug suspend event is ignored.
0	SOFTRESET	R/W	0h	Software reset. 0h (R/W) = Read 0 : reset done, no pending action Write 0 : no action 1h (R/W) = Read 1 : initiate software reset Write 1 : Reset ongoing

19.1.4.3 DMTIMER_IRQ_EOI Register (Offset = 20h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_IRQ_EOI is shown in Figure 19-14 and described in Table 19-14.

Return to Summary Table.

Software End-Of-Interrupt: Allows the generation of further pulses on the interrupt line, if an new interrupt event is pending, when using the pulsed output. Unused when using the level interrupt line (depending on module integration).

Figure 19-14. DMTIMER_IRQ_EOI Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							LINE_NUMBER
R-0h							R-0/W-0h

Table 19-14. DMTIMER_IRQ_EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	LINE_NUMBER	R-0/W	0h	Write the number of the interrupt line to apply a SW EOI to it. Note that there is only a single line (i.e. number 0). Read : Read always returns 0 Write 0 : SW EOI on interrupt line Write 1 : No action

19.1.4.4 DMTIMER_IRQSTS_RAW Register (Offset = 24h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_IRQSTS_RAW is shown in Figure 19-15 and described in Table 19-15.

Return to Summary Table.

Component interrupt request status. Check the corresponding secondary status register. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.

Figure 19-15. DMTIMER_IRQSTS_RAW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">RESERVED</td><td>TCAR_IT_FLAG</td><td>OVF_IT_FLAG</td><td>MAT_IT_FLAG</td></tr>
<tr><td colspan="5">R-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 19-15. DMTIMER_IRQSTS_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	TCAR_IT_FLAG	R/W	0h	IRQ status for Capture Read 0 : No event pending Write 0 : No action Read 1 : IRQ event pending Write 1 : Trigger IRQ event by software
1	OVF_IT_FLAG	R/W	0h	IRQ status for Overflow Read 0 : No event pending Write 0 : No action Read 1 : IRQ event pending Write 1 : Trigger IRQ event by software
0	MAT_IT_FLAG	R/W	0h	IRQ status for Match Read 0 : No event pending Write 0 : No action Read 1 : IRQ event pending Write 1 : Trigger IRQ event by software

19.1.4.5 DMTIMER_IRQSTS Register (Offset = 28h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_IRQSTS is shown in Figure 19-16 and described in Table 19-16.

Return to Summary Table.

Component interrupt request status. Check the corresponding secondary status register.Enabled status isn't set unless event is enabled.Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, i.e. even if not enabled).

Figure 19-16. DMTIMER_IRQSTS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">RESERVED</td><td>TCAR_IT_FLAG</td><td>OVF_IT_FLAG</td><td>MAT_IT_FLAG</td></tr>
<tr><td colspan="5">R-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 19-16. DMTIMER_IRQSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	TCAR_IT_FLAG	R/W	0h	IRQ status for Capture Read 0 : No event pending Write 0 : No action Read 1 : IRQ event pending Write 1 : Clear pending event, if any
1	OVF_IT_FLAG	R/W	0h	IRQ status for Overflow Read 0 : No event pending Write 0 : No action Read 1 : IRQ event pending Write 1 : Clear pending event, if any
0	MAT_IT_FLAG	R/W	0h	IRQ status for Match Read 0 : No event pending Write 0 : No action Read 1 : IRQ event pending Write 1 : Clear pending event, if any

19.1.4.6 DMTIMER_IRQEN_SET Register (Offset = 2Ch) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_IRQEN_SET is shown in Figure 19-17 and described in Table 19-17.

Return to Summary Table.

Component interrupt request enable Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.

Figure 19-17. DMTIMER_IRQEN_SET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					TCAR_EN_FLAG	OVF_EN_FLAG	MAT_EN_FLAG
R-0h					R/W-0h	R/W-0h	R/W-0h

Table 19-17. DMTIMER_IRQEN_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	TCAR_EN_FLAG	R/W	0h	IRQ enable for Compare Read 0 : IRQ event is disabled Write 0 : No action Read 1 : IRQ event is enabled Write 1 : Set IRQ enable
1	OVF_EN_FLAG	R/W	0h	IRQ enable for Overflow Read 0 : IRQ event is disabled Write 0 : No action Read 1 : IRQ event is enabled Write 1 : Set IRQ enable
0	MAT_EN_FLAG	R/W	0h	IRQ enable for Match Read 0 : IRQ event is disabled Write 0 : No action Read 1 : IRQ event is enabled Write 1 : Set IRQ enable

19.1.4.7 DMTIMER_IRQEN_CLR Register (Offset = 30h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_IRQEN_CLR is shown in Figure 19-18 and described in Table 19-18.

Return to Summary Table.

Component interrupt request enable Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.

Figure 19-18. DMTIMER_IRQEN_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					TCAR_EN_FLAG	OVF_EN_FLAG	MAT_EN_FLAG
R-0h					R/W-0h	R/W-0h	R/W-0h

Table 19-18. DMTIMER_IRQEN_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	TCAR_EN_FLAG	R/W	0h	IRQ enable for Compare Read 0 : IRQ event is disabled Write 0 : No action Read 1 : IRQ event is enabled Write 1 : Clear IRQ enable
1	OVF_EN_FLAG	R/W	0h	IRQ enable for Overflow Read 0 : IRQ event is disabled Write 0 : No action Read 1 : IRQ event is enabled Write 1 : Clear IRQ enable
0	MAT_EN_FLAG	R/W	0h	IRQ enable for Match Read 0 : IRQ event is disabled Write 0 : No action Read 1 : IRQ event is enabled Write 1 : Clear IRQ enable

19.1.4.8 DMTIMER_IRQWAKEEN Register (Offset = 34h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_IRQWAKEEN is shown in Figure 19-19 and described in Table 19-19.

Return to Summary Table.

Wakeup-enabled events taking place when module is idle shall generate an asynchronous wakeup.

Figure 19-19. DMTIMER_IRQWAKEEN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					TCAR_WUP_ENA	OVF_WUP_ENA	MAT_WUP_ENA
R-0h					R/W-0h	R/W-0h	R/W-0h

Table 19-19. DMTIMER_IRQWAKEEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	TCAR_WUP_ENA	R/W	0h	Wakeup generation for Compare 0h (R/W) = Wakeup disabled 1h (R/W) = Wakeup enabled
1	OVF_WUP_ENA	R/W	0h	Wakeup generation for Overflow 0h (R/W) = Wakeup disabled 1h (R/W) = Wakeup enabled
0	MAT_WUP_ENA	R/W	0h	Wakeup generation for Match 0h (R/W) = Wakeup disabled 1h (R/W) = Wakeup enabled

19.1.4.9 DMTIMER_TCLR Register (Offset = 38h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_TCLR is shown in Figure 19-20 and described in Table 19-20.

Return to Summary Table.

This register controls optional features specific to the timer functionality.

Figure 19-20. DMTIMER_TCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	GPO_CFG	CAPT_MODE	PT	TRG		TCM	
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
SCPWM	CE	PRE	PTV			AR	ST
R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	R/W-0h

Table 19-20. DMTIMER_TCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14	GPO_CFG	R/W	0h	General Purpose Output - This register directly drives the PORGPOCFG output pin. 0h (R/W) = PORGPOCFG drives 0 1h (R/W) = PORGPOCFG drives 1
13	CAPT_MODE	R/W	0h	Capture mode select bit (first/second) 0h (R/W) = single capture 1h (R/W) = capture on second event
12	PT	R/W	0h	Pulse or toggle mode on PORTIMERPWM output pin 0h (R/W) = pulse 1h (R/W) = toggle
11-10	TRG	R/W	0h	Trigger output mode on PORTIMERPWM output pin 0h (R/W) = no trigger 1h (R/W) = trigger on overflow 2h (R/W) = trigger on overflow and match 3h (R/W) = reserved
9-8	TCM	R/W	0h	Transition Capture Mode on PIEVENTCAPT input pin (When the TCM field passed from (00) to any other combination then the TCAR_IT_FLAG and the edge detection logic are cleared) 0h (R/W) = no capture 1h (R/W) = capture on low to high transition 2h (R/W) = capture on both edge transition 3h (R/W) = Capture on booth edges of PIEVETCAPT
7	SCPWM	R/W	0h	This bit should be set or clear while the timer is stopped or the trigger is off. 0h (R/W) = clear the PORTIMERPWM output pin and select positive pulse for pulse mode 1h (R/W) = set the PORTIMERPWM output pin and select negative pulse for pulse mode

Table 19-20. DMTIMER_TCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	CE	R/W	0h	Compare enable 0h (R/W) = compare mode is disable 1h (R/W) = compare mode is enable
5	PRE	R/W	0h	Prescaler enable 0h (R/W) = The TIMER clock input pin clocks the counter 1h (R/W) = The divided input pin clocks the counter
4-2	PTV	R/W	0h	Pre-scale clock Timer Value
1	AR	R/W	0h	Auto-reload mode 0h (R/W) = One shot timer 1h (R/W) = Auto-reload timer
0	ST	R/W	0h	Start/Stop timer control 0h (R/W) = Stop timer. Only the counter is frozen In case of one-shot mode selected (AR =0), this bit is automatically reset by internal logic when the counter is overflowed. 1h (R/W) = Start timer

19.1.4.10 DMTIMER_TCRR Register (Offset = 3Ch) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_TCRR is shown in Figure 19-21 and described in Table 19-21.

Return to Summary Table.

This register holds the value of the internal counter

Figure 19-21. DMTIMER_TCRR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER_CTR																															
R/W-0h																															

Table 19-21. DMTIMER_TCRR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TIMER_CTR	R/W	0h	Value of TIMER counter

19.1.4.11 DMTIMER_TLDR Register (Offset = 40h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_TLDR is shown in Figure 19-22 and described in Table 19-22.

Return to Summary Table.

This register holds the timer's load value

Figure 19-22. DMTIMER_TLDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOAD_VALUE																															
R/W-0h																															

Table 19-22. DMTIMER_TLDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LOAD_VALUE	R/W	0h	Timer counter value loaded on overflow in auto-reload mode or on TTGR write access LOAD_VALUE must be different than the timer overflow value (0xFFFFFFFF).

19.1.4.12 DMTIMER_TTGR Register (Offset = 44h) [reset = FFFFFFFFh]

Register mask: FFFFFFFFh

DMTIMER_TTGR is shown in Figure 19-23 and described in Table 19-23.

Return to Summary Table.

The read value of this register is always 0xFFFF FFFF.

Figure 19-23. DMTIMER_TTGR Register

31-0
TTGR_VALUE
R-1/W-FFFFFFFFh

Table 19-23. DMTIMER_TTGR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TTGR_VALUE	R-1/W	FFFFFFFFh	Writing in the TTGR register, TCRR will be loaded from TLDR and prescaler counter will be cleared Reload will be done regardless of the AR field value of TCLR register

19.1.4.13 DMTIMER_TWPS Register (Offset = 48h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_TWPS is shown in Figure 19-24 and described in Table 19-24.

Return to Summary Table.

This register contains the write posting bits for all writ-able functional registers

Figure 19-24. DMTIMER_TWPS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">RESERVED</td><td>W_PEND_TMAR</td><td>W_PEND_TTGR</td><td>W_PEND_TLDR</td><td>W_PEND_TCRR</td><td>W_PEND_TCLR</td></tr>
<tr><td colspan="3">R-0h</td><td>R-0h</td><td>R-0h</td><td>R-0h</td><td>R-0h</td><td>R-0h</td></tr>
</table>

Table 19-24. DMTIMER_TWPS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4	W_PEND_TMAR	R	0h	When equal to 1, a write is pending to the TMAR register
3	W_PEND_TTGR	R	0h	When equal to 1, a write is pending to the TTGR register
2	W_PEND_TLDR	R	0h	When equal to 1, a write is pending to the TLDR register
1	W_PEND_TCRR	R	0h	When equal to 1, a write is pending to the TCRR register
0	W_PEND_TCLR	R	0h	When equal to 1, a write is pending to the TCLR register

19.1.4.14 DMTIMER_TMAR Register (Offset = 4Ch) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_TMAR is shown in Figure 19-25 and described in Table 19-25.

Return to Summary Table.

The compare logic consists of a 32-bit wide, read/write data TMAR register and logic to compare counter

Figure 19-25. DMTIMER_TMAR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMPARE_VALUE																															
R/W-0h																															

Table 19-25. DMTIMER_TMAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMPARE_VALUE	R/W	0h	Value to be compared to the timer counter

19.1.4.15 DMTIMER_TCAR1 Register (Offset = 50h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_TCAR1 is shown in Figure 19-26 and described in Table 19-26.

Return to Summary Table.

Figure 19-26. DMTIMER_TCAR1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">CAPTURE_VALUE1</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 19-26. DMTIMER_TCAR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPTURE_VALUE1	R	0h	Timer counter value captured on an external event trigger

19.1.4.16 DMTIMER_TSICR Register (Offset = 54h) [reset = X]

DMTIMER_TSICR is shown in Figure 19-27 and described in Table 19-27.

Return to Summary Table.

Timer Synchronous Interface Control Register

Figure 19-27. DMTIMER_TSICR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					POSTED	SFT	RESERVED
R-0h					R/W-X	R-0/W-0h	R-0h

Table 19-27. DMTIMER_TSICR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	POSTED	R/W	X	Reset value of POSTED depends on hardware integration module at design time. Software must read POSTED field to get the hardware module configuration. 0h (R/W) = posted mode inactive: will delay the command accept output signal. 1h (R/W) = posted mode active (clocks ratio needs to fit freq (timer) < freq (OCP)/4 frequency requirement)
1	SFT	R-0/W	0h	This bit reset all the functional part of the module 0h (R/W) = software reset is disabled 1h (R/W) = software reset is enabled
0	RESERVED	R	0h	

19.1.4.17 DMTIMER_TCAR2 Register (Offset = 58h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_TCAR2 is shown in Figure 19-28 and described in Table 19-28.

Return to Summary Table.

Figure 19-28. DMTIMER_TCAR2 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">CAPTURE_VALUE2</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 19-28. DMTIMER_TCAR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPTURE_VALUE2	R	0h	Timer counter value captured on an external event trigger

19.2 DMTimer 1ms

19.2.1 Introduction

This peripheral is a 32-bit timer offering:

- Counter timer with compare and capture modes
- Auto-reload mode
- Start-stop mode
- Generate 1 ms tick with 32768-Hz functional clock
- Programmable divider clock source
- 16–32 bit addressing
- On-the-fly read/write registers
- Interrupts generated on overflow, compare and capture
- Interrupt enable
- Wake-up enable
- Write posted mode
- Dedicated input trigger for capture mode and dedicated output trigger/PWM signal
- Dedicated output signal for general purpose use PORGPOCFG
- OCP interface compatible

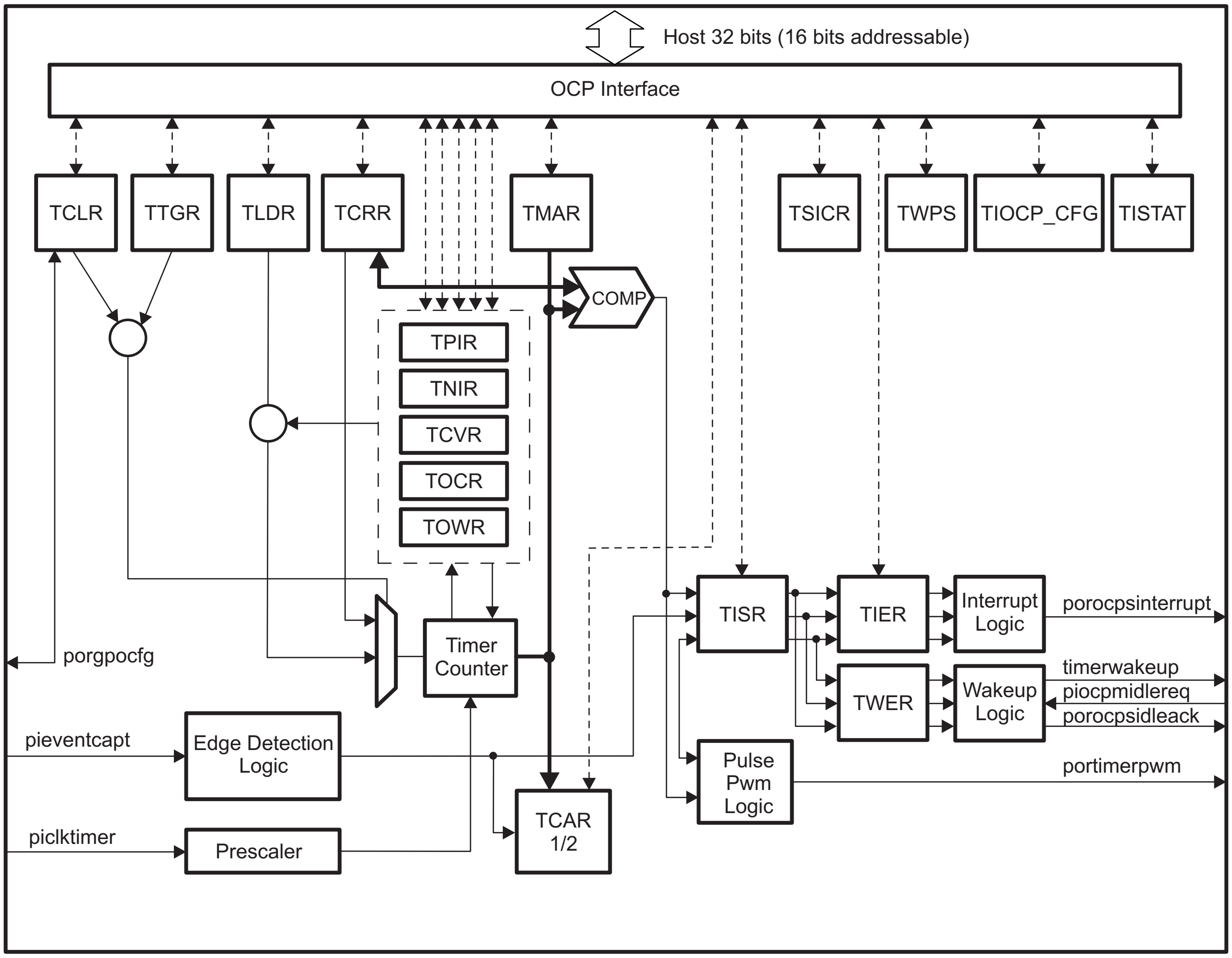
The timer module contains a free running upward counter with auto reload capability on overflow. The timer counter can be read and written on the fly (while counting). The timer module includes compare logic to allow interrupt event on programmable counter matching value.

A dedicated output signal can be pulsed or toggled on overflow and match event. This output offers timing stamp trigger signal or PWM (pulse width modulation) signal sources. A dedicated output signal can be used for general purpose PORGPOCFG (directly driven by the bit 14 of the TCLR register). A dedicated input signal is used to trigger automatic timer counter capture and interrupt event, on programmable input signal transition type. A programmable clock divider (prescaler) allows reduction of the timer input clock frequency. All internal timer interrupt sources are merged in one module interrupt line and one wake-up line. Each internal interrupt source can be independently enabled/disabled with a dedicated bit of TIER register for the interrupt features and a dedicated bit of TWER for the wake-up.

This module is controllable through the OCP peripheral bus.

As two clocks domains are managed inside this module, resynchronization is done by special logic between OCP clock domain and Timer clock domain. At reset, synchronization logic allows utilization of all ratios between OCP clock and Timer clock. Drawback of this mode is that full-resynchronization path is used with access latency performance impact in terms of OCP clock cycles. In order to improve module access latency, and under restricted conditions on clocks ratios (cf. 7.1 Write posted), write-posted mode can be used by setting POSTED bit of System Control register (TSICR). Under this mode, write posted mode is enabled, meaning that OCP write command is granted before the write process completes in the timer clock domain. This mode allows software (SW) to do concurrent writes on Dual Mode timer registers and to observe write process completion (synchronization) at SW level by reading independent write posted status bits in the Write Posted Status Register (TWPS).

Figure 19-29. Block Diagram



19.2.2 Integration

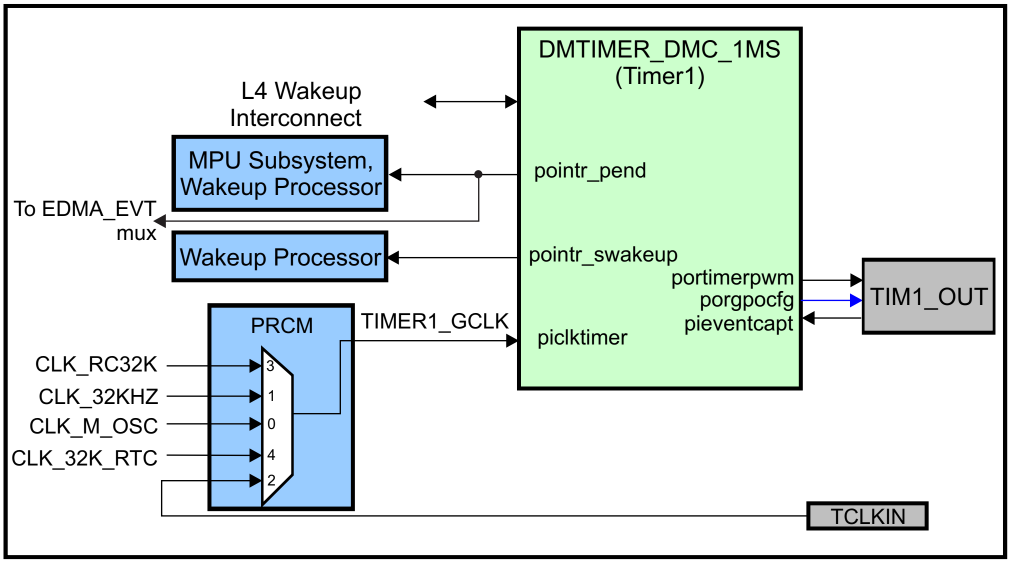


Figure 19-30. DMTimer 1 ms Integration

19.2.2.1 Timer Connectivity Attributes

Table 19-29. Timer1 Connectivity Attributes

Attributes	Type
Power Domain	Wakeup Domain
Clock Domain	PD_WKUP_L4_WKUP_GCLK (OCP) PD_WKUP_TIMER1_GCLK (Func)
Reset Signals	WKUP_DOM_RST_N
Idle/Wakeup Signals	Smart Idle / Slave Wakeup
Interrupt Requests	1 to MPU Subsystem (TINT1_1MS), Wakeup Processor, EDMA SWAKEUP to Wakeup Processor
DMA Requests	None
Physical Address	L4 Wakeup slave port

19.2.2.2 Timer Clock and Reset Manangement

The DMTimer 1ms timer functional clock can be selected from one of the following sources using the CLKSEL_TIMER1MS_CLK register in the PRCM:

- The 24 MHz (typ) system clock (CLK_M_OSC)
- The PER PLL generated 32.768 KHz clock (CLK_32KHZ)
- The TCLKIN external timer input clock
- The on-chip ~32.768 KHz oscillator (CLK_RC32K)
- The external 32.768 KHz crystal/clock (CLK_32K_RTC)

19.2.2.3 Timer Clock Signals

Table 19-30. Timer Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
Timer1 (1ms) Clock Signals			
PICLKOCP Interface clock	26 MHz	CLK_M_OSC	pd_wkup_l4_wkup_gclk from PRCM
PICLKTIMER Functional clock	26 MHz	CLK_M_OSC CLK_32KHZ (PER_CLKOUTM2 / 5859.375) TCLKIN CLK_RC32K CLK_32K_RTC	pd_wkup_timer1_gclk from PRCM

19.2.2.4 Timer Pin List

The timer PIEVENTCAPT input and PORTIMERPWM output signals are muxed onto a single TIMER I/O pad. The pad direction (and hence the pin function) are controlled from within the DMTimer module using the PORGPOCFG signal as an output enable.

Table 19-31. Timer Pin List

Pin	Type	Description
TIMER1	I/O	Timer 1 trigger input or PWM output.

19.2.3 Functional Description

The general-purpose timer is an upward counter. It supports three functional modes:

- Timer mode
- Capture mode
- Compare mode

By default, after core reset, the capture and compare modes are disabled.

19.2.3.1 Timer Mode Functionality

The timer is an upward counter that can be started and stopped at any time through the Timer Control Register (TCLR ST bit). The Timer Counter Register (TCRR) can be loaded when stopped or on the fly (while counting). TCRR can be loaded directly by a TCRR Write access with the new timer value. TCRR can also be loaded with the value held in the Timer Load Register (TLDR) by a trigger register (TTGR) Write access. The TCRR loading is done regardless of the value written to the TTGR register. The timer counter register TCRR value can be read when stopped or captured on the fly by a TCRR Read access. The timer is stopped and the counter value set to “0” when the module’s reset is asserted. The timer is maintained in stop after reset is released. When the timer is stopped, TCRR is frozen. The timer can be restarted from the frozen value unless TCRR has been reloaded with a new value.

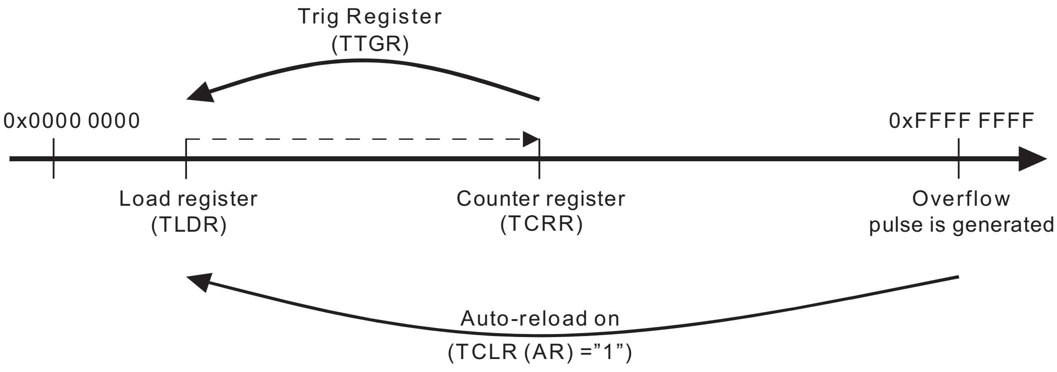
In the one shot mode (TCLR AR bit =”0”), the counter is stopped after counting overflow (counter value remains at zero).

When the auto-reload mode is enabled (TCLR AR bit =”1”), the TCRR is reloaded with the TLDR value after a counting overflow.

It is not recommended to put the overflow value (0xFFFFFFFF) in TLDR because it can lead to undesired results.

An interrupt can be issued on overflow if the overflow interrupt enable bit is set in the timer Interrupt Enable Register (TIER OVF_IT_ENA bit =”1”). A dedicated output pin (PORTIMERPWM) is programmed through TCLR (TRG and PT bits) to generate one positive pulse (prescaler duration) or to invert the current value (toggle mode) when an overflow occurs.

Figure 19-31. TCRR Timing Value



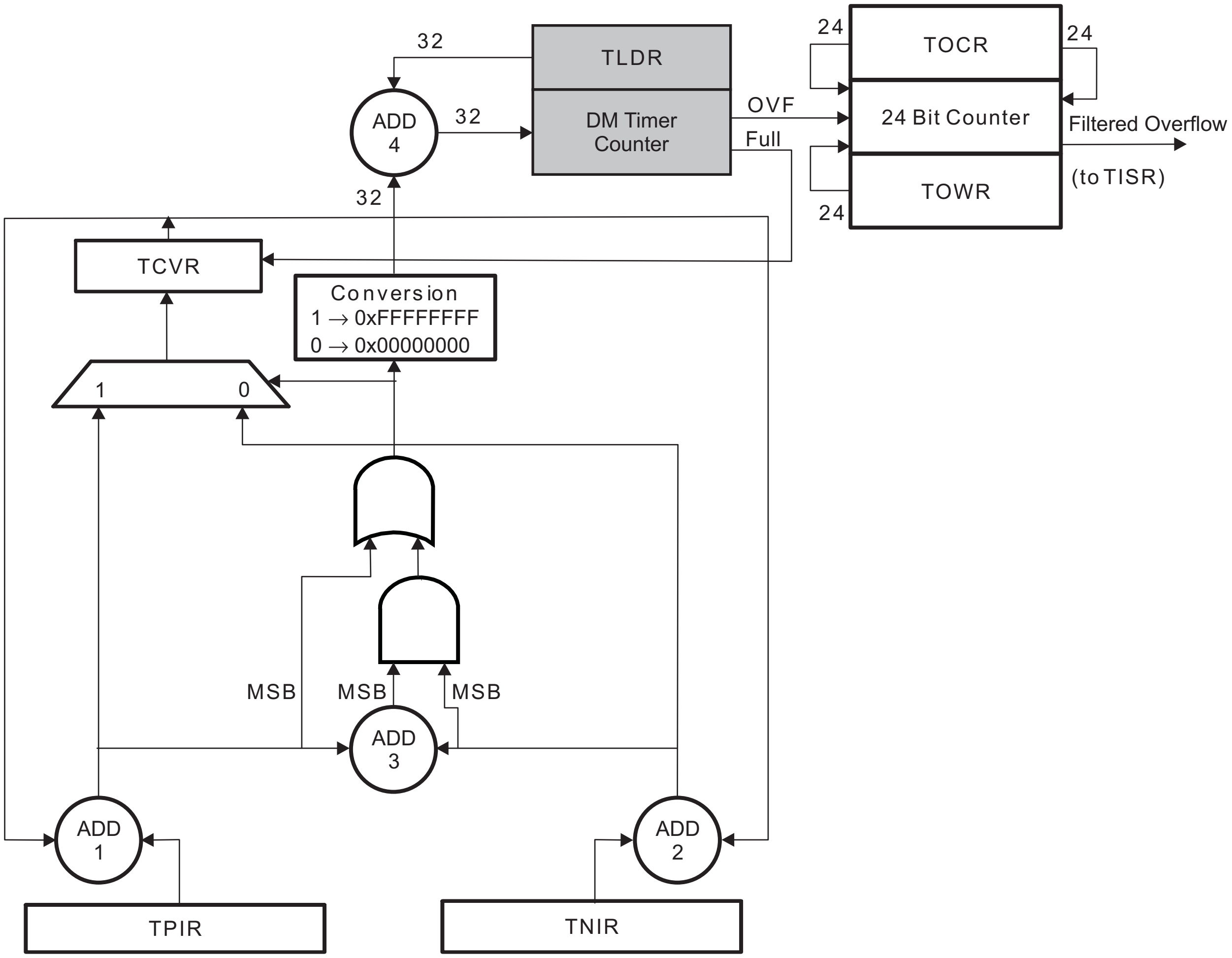
19.2.3.1.1 1 ms Tick Generation

To minimize the error between a true 1ms tick and the tick generated by the 32768 Hz timer, the sequencing of the sub-1ms periods and the over-1ms periods must be shuffled.

An additional block (1ms block) is used to correct this error.

In this implementation, the increment sequencing is automatically managed by the timer to minimize the error. The value of the Timer Positive Increment register (TPIR) and Timer Negative Increment register (TNIR) only need to be defined by the user. An auto adaptation mechanism is used to simplify the programming model.

Figure 19-32. 1ms Module Block Diagram



The TPIR, TNIR, TCVR and adders Add1–3 are used to define whether the next value loaded in the TCRR will be the value of the TLDR (sub-period value) or the value of TLDR – 1 (over-period value).

The following table shows the value loaded in TCRR according to the sign of the result of Add1, Add2 and Add3. MSB = ‘0’ means a positive value, MSB = ‘1’ means a negative value.

Table 19-32. Value Loaded in TCRR to Generate 1ms Tick

Add1 MSB	Add2 MSB	Add3 MSB	TCRR
0	0	0	TLDR
0	0	1	TLDR
0	1	0	TLDR
0	1	1	TLDR - 1
1	0	0	N.A.
1	0	1	N.A.
1	1	0	TLDR - 1
1	1	1	TLDR - 1

The values of TPIR and TNIR registers are calculated with formula:

Positive Increment Value = ((INTEGER[Fclk * Ttick] + 1) * 1e6) - (Fclk * Ttick * 1e6)
Negative Increment Value = (INTEGER[Fclk * Ttick] * 1e6) - (Fclk * Ttick * 1e6)

where:

Fclk – clock frequency (KHz)
Ttick – tick period (ms)

The Timer Overflow Counter Register (TOCR) and the Timer Overflow Wrapping Register (TOWR) are used for interrupt filtering. When the timer overflows, it increments the 24 bit TOCR register. When the 24-bit TOCR register value matches the value in the 24 bit TOWR register and timer overflow is asserted, the TOCR is reset and an interrupt is generated to TISR.

With the Conversion block in reset state (Positive Increment register, Negative Increment register and Counter Value register are all zeroed), the programming model and the behavior of the DMtimer_dmc1ms remain unchanged.

For 1 ms tick with a 32768-Hz clock:

TPIR = 232000
TNIR = -768000
TLDR = 0xFFFFFFE0

NOTE: Any value of the tick period can be generated with the appropriate values of the TPIR, TNIR and TLDR registers.

By default, the TPIR, TNIR, TCVR, TOCR, TOWR registers and the associated logic are in reset mode (all 0s) and have no action on the programming model of the DMtimer_dmc1ms.

19.2.3.2 Capture Mode Functionality

The timer value in TCRR can be captured and saved in TCAR1 or TCAR2 function of the mode selected in TCLR through the field CAPT_MODE when a transition is detected on the module input pin (PIEVENTCAPT). The edge detection circuitry monitors transitions on the input pin (PIEVENTCAPT).

Rising transition, falling transition or both can be selected in TCLR (TCM bit) to trigger the timer counter capture. The module sets the TISR (TCAR_IT_FLAG bit) when an active transition is detected and at the same time the counter value TCRR is stored in one of the timer capture registers TCAR1 or TCAR2 as follows:

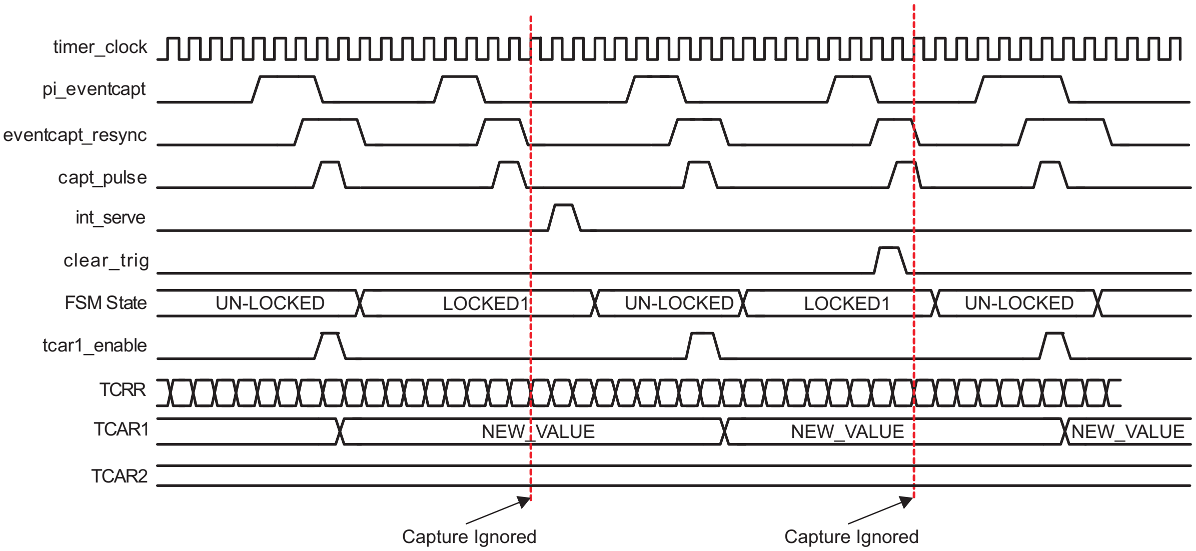
- If TCLR’s CAPT_MODE field is “0” then, on the first enabled capture event, the value of the counter register is saved in TCAR1 register and all the next events are ignored (no update on TCAR1 and no interrupt triggering) until the detection logic is reset or the interrupt status register is cleared on TCAR’s position by writing a “1” to it.
- If TCLR’s CAPT_MODE field is “1” then, on the first enabled captured event, the counter value is saved in TCAR1 register and, on the second enabled capture event, the value of the counter register is saved in TCAR2 register. If capture interrupt is enabled, the interrupt will be asserted on the second event capture. All the other events are ignored (no update on TCAR1/2 and no interrupt triggering) until the detection logic is reset or the interrupt status register is cleared on TCAR’s position writing a “1” in it. This mechanism is useful for period calculation of a clock if that clock is connected to the PIEVENTCAPT input pin.

The edge detection logic is reset (a new capture is enabled) when the active capture interrupt is served. The TCAR_IT_FLAG bit of TISR (previously ‘1’) is cleared by writing a "1" to it or when the edge detection mode bits TCLR (TCM bit) passed from the No Capture Mode detection to any other modes. The timer functional clock (input to prescaler) is used to sample the input pin (PIEVENTCAPT). Negative or positive input pulses can be detected when the pulse time exceeds the functional clock period. An interrupt can be issued on transition detection if the capture interrupt enable bit is set in the Timer Interrupt Enable Register TIER (TCAR_IT_ENA bit).

See the following examples:

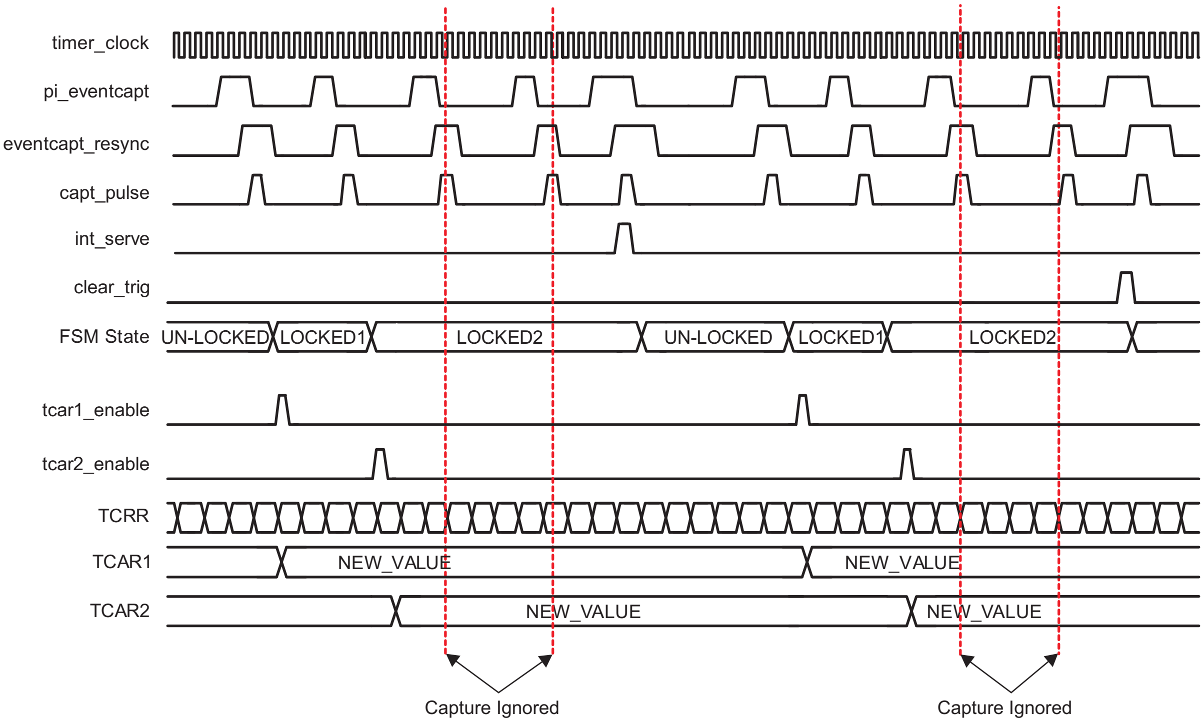
In the next wave, the TCM value is “01” and CAPT_MODE is “0”- only rising edge of the PIEVENTCAPT will trigger a capture in TCAR and only TCAR1 will update.

Figure 19-33. Capture Wave Example for CAPT_MODE 0



In the following example, the TCM value is “01” and CAPT_MODE is “1”- only rising edge of the PIEVENTCAPT will trigger a capture in TCAR1 on first enabled event and TCAR2 will update on the second enabled event.

Figure 19-34. Capture Wave Example for CAPT_MODE 1



19.2.3.3 Compare Mode Functionality

When Compare Enable TCLR (CE bit) is set to “1”, the timer value (TCRR) is permanently compared to the value held in timer match register (TMAR). TMAR value can be loaded at any time (timer counting or stop). When the TCRR and the TMAR values match, an interrupt can be issued if the TIER (MAT_IT_ENA bit) is set. The correct implementation is to write a compare value in TMAR register before setting TCLR (CE bit) to avoid any unwanted interrupts due to a reset value matching effect.

The dedicated output pin (PORTIMERPWM) can be programmed through TCLR (TRG and PT bits) to generate one positive pulse (TIMER clock duration) or to invert the current value (toggle mode) when an overflow and a match occur.

19.2.3.4 Prescaler Functionality

A prescaler counter can be used to divide the timer counter input clock frequency. The prescaler is enabled when TCLR bit 5 is set (PRE). The 2n division ratio value (PTV) can be configured in the TCLR register.

The prescaler counter is reset when the timer counter is stopped or reloaded on the fly.

Table 19-33. Prescaler/Timer Reload Values Versus Contexts

Contexts	Prescaler Counter	Timer Counter
Overflow (when Auto-reload on)	reset	TLDR
TCRR Write	reset	TCRR
TTGR Write	reset	TLDR
Stop	reset	Frozen

19.2.3.5 Pulse-Width Modulation

The timer can be configured to provide a programmable pulse-width modulation (PORTIMERPWM) output. The PORTIMERPWM output pin can be configured to toggle on a specified event. TCLR (TRG bits) determines on which register value the PORTIMERPWM pin toggles. Either overflow or match can be used to toggle the PORTIMERPWM pin, when a compare condition occurs.

In case of overflow and match mode, the match event will be ignored from the moment the mode was set-up until the first overflow event occurs

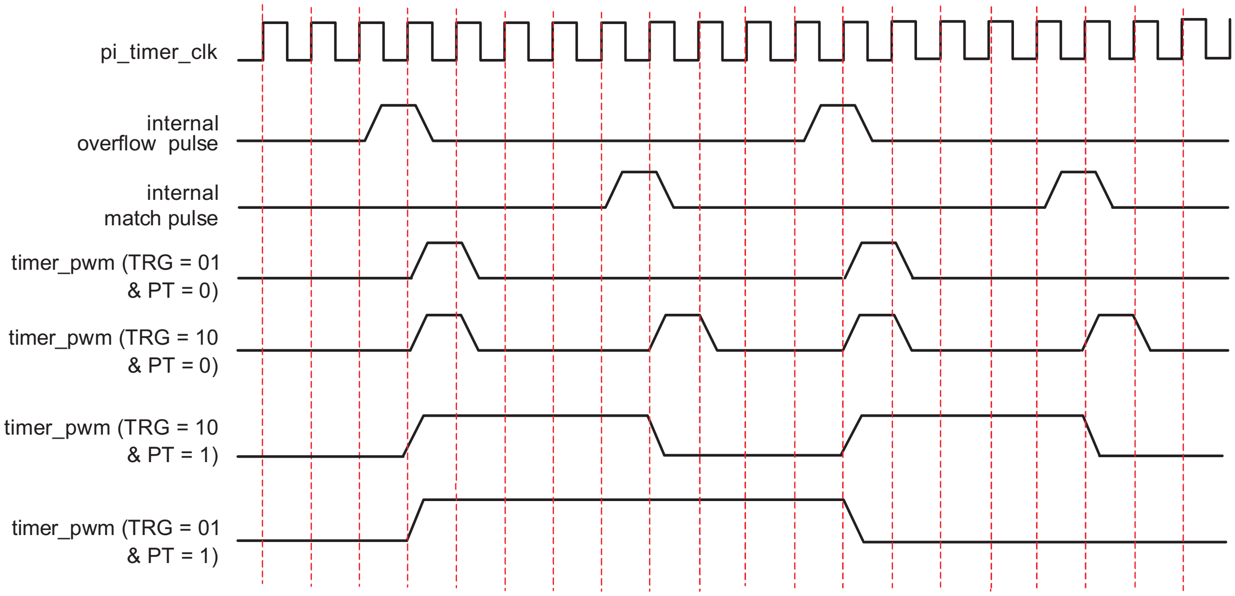
The TCLR (SCPWM bit) can be programmed to set or clear the PORTIMERPWM output signal while the counter is stopped or the triggering is off only. This allows fixing a deterministic state of the output pin when modulation is stopped. The modulation is synchronously stopped when TRG bit is cleared and an overflow has occurred.

In the following timing diagram, the internal overflow pulse is set each time (0xFFFF FFFFF – TLDR +1) value is reached, and the internal match pulse is set when the counter reaches TMAR register value. According to TCLR (TRG and PT bits) programming value, the timer provides pulse or PWM on the output pin (PORTIMERPWM).

The TLDR and TMAR registers must keep values smaller than the overflow value (0xFFFFFFFF) with at least 2 units. In case the PWM trigger events are both overflow and match, the difference between the values kept in TMAR register and the value in TLDR must be at least 2 units. When match event is used, the compare mode TCLR (CE) must be set.

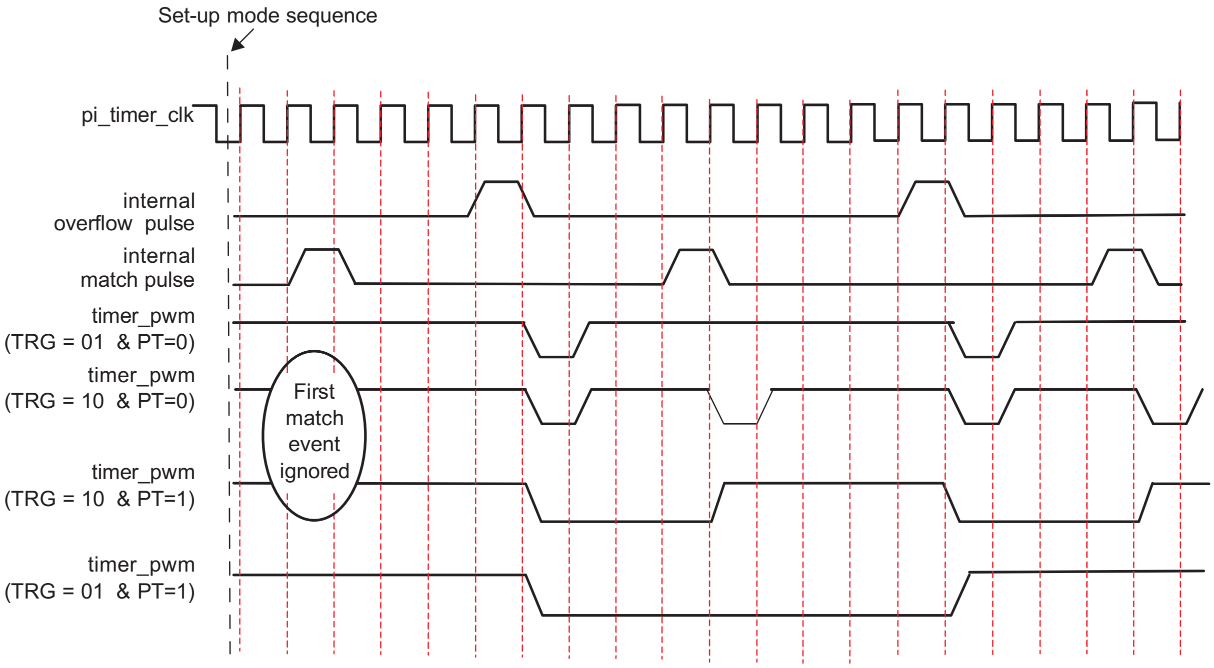
On the following wave TCLR (SCPWM bit) is set to ‘0’.

Figure 19-35. Timing Diagram of Pulse-Width Modulation, SCPWM Bit = 0



On the next wave TCLR (SCPWM bit) is set to ‘1’.

Figure 19-36. Timing Diagram of Pulse-Width Modulation, SCPWM Bit = 1



19.2.3.6 Timer Interrupt Control

The timer can issue an overflow interrupt, a timer match interrupt and a timer capture interrupt. Each internal interrupt source can be independently enabled/disabled in the Interrupt Enable Register (TIER). When the interrupt event has been issued, the associated interrupt status bit is set in the Timer Status Register (TISR). The pending interrupt event is reset when the set status bit is overwritten by a “1” value. Reading the Interrupt Status Register and writing the value back allows for a fast acknowledge interrupt process.

19.2.3.7 Sleep Mode Request and Acknowledge

Upon a Sleep mode request issued by the host processor (the Idle Request PIOCPMIDLEREQ signal is active), the timer module will enter Sleep mode according to the IdleMode field of the System configuration register (see TIOCP_CFG).

If the IdleMode field sets No-Idle mode, the Timer does not enter Sleep mode and the Idle acknowledge signal (POROCPSIDLEACK) is never asserted.

If the IdleMode field sets Force-Idle mode, the timer enters Sleep mode independently of the internal module state and the Idle acknowledge signal (POROCPSIDLEACK) is unconditionally asserted.

If the IdleMode field sets Smart-Idle mode, the timer module evaluates its internal capability to have the interface/functional clock switched off. Depending on the ClockActivity field, setting the timer module evaluates the internal activity and asserts the Idle acknowledge signal (POROCPSIDLEACK), entering in Sleep mode, ready to issue a wake-up request.

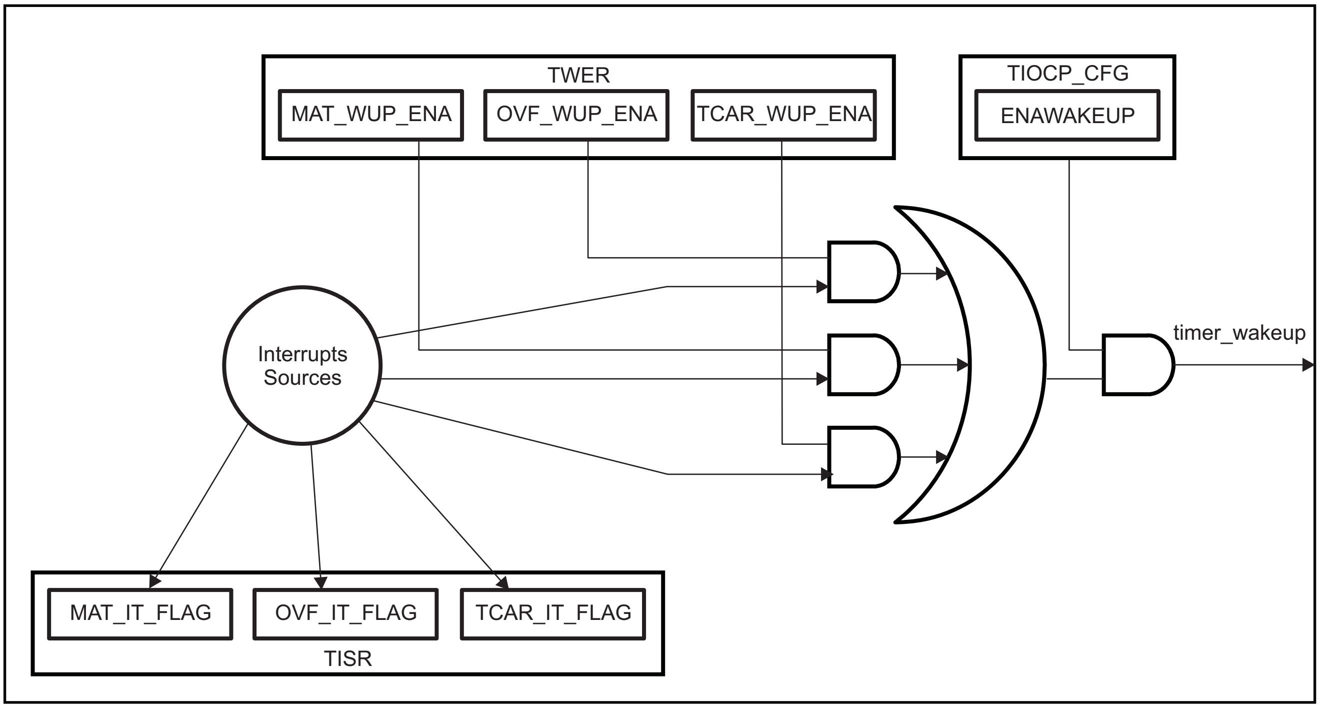
The following table describes the Smart Idle behavior according to the clock activity setting:

Table 19-34. SmartIdle - Clock Activity Field Configuration

Clock Activity	Functional Clock	OCP Clock	Module Behavior
11	ON	ON	The Idle acknowledge signal is asserted when there are no pending activities on the OCP clock domain, without evaluating the pending activities on the functional clock domain. (The module will enter in Sleep mode and if a pending interrupt event is finished during Idle mode the wake-up signal will be asserted).
10	ON	OFF	
01	OFF	ON	The Idle acknowledge signal is asserted when there are no pending activities on the functional and OCP clock domains (Improved latency in assertion of Idle acknowledge). The Wake-up capability of the module is disabled.
00	OFF	OFF	

This wake-up request is effectively sent only if the field ENAWAKEUP of TIOCP_CFG enables the timer wake-up capability. When the system is awaken, the Idle Request signal goes inactive and the wake-up request signal is also de-asserted.

Figure 19-37. Wake-up Request Generation



19.2.3.7.1 Wake-up Line Release

When the host processor receives a wake-up request issued by the timer peripheral, the interface clock is re-activated: the host processor deactivates the PIOCPMIDLEREQ, the timer deactivates the POROCPSIDLEACK signal and then the host can read the corresponding bit in TISR to find out which interrupt source has triggered a wake-up request. After acknowledging the wake-up request, the processor resets the status bit and releases the interrupt line by writing a ‘1’ in the corresponding bit of the TISR register.

19.2.3.8 Timer Counting Rate

The dmtimer’s counter is composed of a prescaler stage and a timer counter.

The prescaler clock ratio can be managed by accessing the ratio definition field of the control register (PTV and PRE of TCLR).

The timer rate is defined by:

- The value of the prescaler fields (PRE and PTV of TCLR register)
- The value loaded into the Timer Load Register (TLDR).

Table 19-35. Prescaler Clock Ratios Value

PRE	PTV	Divisor (PS)
0	X	1
1	0	2
1	1	4
1	2	8
1	3	16
1	4	32

Table 19-35. Prescaler Clock Ratios Value (continued)

PRE	PTV	Divisor (PS)
1	5	64
1	6	128
1	7	256

The timer rate equation is as follows:

(0xFFFF FFFF – TLDR + 1) x timer Clock period x Clock Divider (PS)

With timer Clock period = 1/ timer Clock frequency and PS = 2(PTV + 1).

As example, if we consider a timer clock input of 32 KHz, with a PRE field equals to “0”, the timer output period is:

Table 19-36. Value and Corresponding Interrupt Period

TLDR	Interrupt Period
0x0000 0000	37 h
0xFFFF 0000	2 s
0xFFFF FFF0	500 us
0xFFFF FFFE	62.5 us

19.2.3.9 Timer Behavior During Emulation

To configure the Timer to stop during emulation suspend events (for example, debugger breakpoints), set up the Timer and the Debug Subsystem:

1. Set TIOCP_CFG.EMUFREE=0. This will allow the Suspend_Control signal from the Debug Subsystem (Chapter 31) to stop and start the Timer. Note that if EMUFREE=1, the Suspend_Control signal is ignored and the Timer is free running regardless of any debug suspend event. This EMUFREE bit gives local control from a module perspective to gate the suspend signal coming from the Debug Subsystem.
2. Set the appropriate xxx_Suspend_Control register = 0x9, as described in Section 31.1.1.1, *Debug Suspend Support for Peripherals*. Choose the register appropriate to the peripheral you want to suspend during a suspend event.

19.2.4 DMTIMER_1MS Registers

Table 19-37 lists the memory-mapped registers for the DMTIMER_1MS. All register offset addresses not listed in Table 19-37 should be considered as reserved locations and the register contents should not be modified.

Table 19-37. DMTIMER_1MS Registers

Offset	Acronym	Register Name	Section
0h	DMTIMER_1MS_TIDR		Section 19.2.4.1
10h	DMTIMER_1MS_TIOCP_CFG		Section 19.2.4.2
14h	DMTIMER_1MS_TISTAT		Section 19.2.4.3
18h	DMTIMER_1MS_TISR		Section 19.2.4.4
1Ch	DMTIMER_1MS_TIER		Section 19.2.4.5
20h	DMTIMER_1MS_TWER		Section 19.2.4.6
24h	DMTIMER_1MS_TCLR		Section 19.2.4.7
28h	DMTIMER_1MS_TCRR		Section 19.2.4.8
2Ch	DMTIMER_1MS_TLDR		Section 19.2.4.9
30h	DMTIMER_1MS_TTGR		Section 19.2.4.10
34h	DMTIMER_1MS_TWPS		Section 19.2.4.11

Table 19-37. DMTIMER_1MS Registers (continued)

Offset	Acronym	Register Name	Section
38h	DMTIMER_1MS_TMAR		Section 19.2.4.12
3Ch	DMTIMER_1MS_TCAR1		Section 19.2.4.13
40h	DMTIMER_1MS_TSICR		Section 19.2.4.14
44h	DMTIMER_1MS_TCAR2		Section 19.2.4.15
48h	DMTIMER_1MS_TPIR		Section 19.2.4.16
4Ch	DMTIMER_1MS_TNIR		Section 19.2.4.17
50h	DMTIMER_1MS_TCVR		Section 19.2.4.18
54h	DMTIMER_1MS_TOCR		Section 19.2.4.19
58h	DMTIMER_1MS_TOWR		Section 19.2.4.20

19.2.4.1 DMTIMER_1MS_TIDR Register (offset = 0h) [reset = 15h]

Register mask: FFFFFFFFh

DMTIMER_1MS_TIDR is shown in Figure 19-38 and described in Table 19-38.

This register contains the IP revision code

Figure 19-38. DMTIMER_1MS_TIDR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">RESERVED</td><td colspan="8">TID_REV</td></tr>
<tr><td colspan="24">R-0h</td><td colspan="8">R-15h</td></tr>
</table>

Table 19-38. DMTIMER_1MS_TIDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reads return 0
7-0	TID_REV	R	15h	IP revision [7:4] Major revision [3:0] Minor revision Examples: 0x10 for 1.0, 0x21 for 2.1

19.2.4.2 DMTIMER_1MS_TIOCP_CFG Register (offset = 10h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_1MS_TIOCP_CFG is shown in Figure 19-39 and described in Table 19-39.

This register controls the various parameters of the OCP interface

Figure 19-39. DMTIMER_1MS_TIOCP_CFG Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED						CLOCKACTIVITY	
R/W-0h						R/W-0h	
7	6	5	4	3	2	1	0
RESERVED		EMUFREE	IDLEMODE		ENAWAKEUP	SOFTRESET	AUTOIDLE
R/W-0h		R/W-0h	R/W-0h		R/W-0h	Rreturns0s/W-0h	R/W-0h

Table 19-39. DMTIMER_1MS_TIOCP_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	0h	
9-8	CLOCKACTIVITY	R/W	0h	
7-6	RESERVED	R/W	0h	Write 0's for future compatibility Reads return 0
5	EMUFREE	R/W	0h	Sensitivity to emulation (debug) suspend event from Debug Subsystem. 0h (R/W) = Timer counter frozen during a debug suspend event. 1h (R/W) = Timer counter free-running. Debug suspend event is ignored.
4-3	IDLEMODE	R/W	0h	Power Management, req/ack control 0h (R/W) = Force-idle. An idle request is acknowledged unconditionally 1h (R/W) = No-idle. An idle request is never acknowledged 2h (R/W) = Smart-idle. Acknowledgement to an idle request is given based on the internal activity of the module. The module may generate wakeup events when in idle state. 3h (R/W) = reserved do not use
2	ENAWAKEUP	R/W	0h	Wake-up feature global control 0h (R/W) = No wakeup line assertion in idle mode 1h (R/W) = Wakeup line assertion enabled in smart-idle mode
1	SOFTRESET	Rreturns0s/W	0h	Software reset. This bit is automatically reset by the hardware. During reads, it always return 0 0h (R/W) = Normal mode 1h (R/W) = The module is reset
0	AUTOIDLE	R/W	0h	Internal OCP clock gating strategy 0h (R/W) = OCP clock is free-running 1h (R/W) = Automatic OCP clock gating strategy is applied, based on the OCP interface activity

19.2.4.3 DMTIMER_1MS_TISTAT Register (offset = 14h) [reset = 0h]

Register mask: FFFFFFFEh

DMTIMER_1MS_TISTAT is shown in Figure 19-40 and described in Table 19-40.

This register provides status information about the module, excluding the interrupt status information

Figure 19-40. DMTIMER_1MS_TISTAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							RESETDONE
R-0h							R-X

Table 19-40. DMTIMER_1MS_TISTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reads return 0 Reserved for OCP-socket status information
0	RESETDONE	R	X	Internal reset monitoring 0h (R) = Internal module reset in on-going 1h (R) = Reset completed

19.2.4.4 DMTIMER_1MS_TISR Register (offset = 18h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_1MS_TISR is shown in Figure 19-41 and described in Table 19-41.

The Timer Status Register is used to determine which of the timer events requested an interrupt.

Figure 19-41. DMTIMER_1MS_TISR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					TCAR_IT_FLAG	OVF_IT_FLAG	MAT_IT_FLAG
R-0h					R/W-0h	R/W-0h	R/W-0h

Table 19-41. DMTIMER_1MS_TISR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reads return 0
2	TCAR_IT_FLAG	R/W	0h	indicates when an external pulse transition of the correct polarity is detected on the external pin PIEVENTCAPT 0h (R/W) = no capture interrupt request 1h (R/W) = capture interrupt request
1	OVF_IT_FLAG	R/W	0h	TCRR overflow 0h (R/W) = no overflow interrupt request 1h (R/W) = overflow interrupt pending
0	MAT_IT_FLAG	R/W	0h	the compare result of TCRR and TMAR 0h (R/W) = no compare interrupt request 1h (R/W) = compare interrupt pending

19.2.4.5 DMTIMER_1MS_TIER Register (offset = 1Ch) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_1MS_TIER is shown in Figure 19-42 and described in Table 19-42.

This register controls (enable/disable) the interrupt events

Figure 19-42. DMTIMER_1MS_TIER Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					TCAR_IT_ENA	OVF_IT_ENA	MAT_IT_ENA
R-0h					R/W-0h	R/W-0h	R/W-0h

Table 19-42. DMTIMER_1MS_TIER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reads return 0
2	TCAR_IT_ENA	R/W	0h	Enable capture interrupt 0h (R/W) = Disable capture interrupt 1h (R/W) = Enable capture interrupt
1	OVF_IT_ENA	R/W	0h	Enable overflow interrupt 0h (R/W) = Disable overflow interrupt 1h (R/W) = Enable overflow interrupt
0	MAT_IT_ENA	R/W	0h	Enable match interrupt 0h (R/W) = Disable match interrupt 1h (R/W) = Enable match interrupt

19.2.4.6 DMTIMER_1MS_TWER Register (offset = 20h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_1MS_TWER is shown in Figure 19-43 and described in Table 19-43.

This register controls (enable/disable) the wakeup feature on specific interrupt events

Figure 19-43. DMTIMER_1MS_TWER Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					TCAR_WUP_ENA	OVF_WUP_ENA	MAT_WUP_ENA
R-0h					R/W-0h	R/W-0h	R/W-0h

Table 19-43. DMTIMER_1MS_TWER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reads return 0
2	TCAR_WUP_ENA	R/W	0h	Enable capture wake-up 0h (R/W) = Disable capture wake-up 1h (R/W) = Enable capture wake-up
1	OVF_WUP_ENA	R/W	0h	Enable overflow wake-up 0h (R/W) = Disable overflow wake-up 1h (R/W) = Enable overflow wake-up
0	MAT_WUP_ENA	R/W	0h	Enable match wake-up 0h (R/W) = Disable match wake-up 1h (R/W) = Enable match wake-up

19.2.4.7 DMTIMER_1MS_TCLR Register (offset = 24h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_1MS_TCLR is shown in Figure 19-44 and described in Table 19-44.

This register controls optional features specific to the timer functionality

Figure 19-44. DMTIMER_1MS_TCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	GPO_CFG	CAPT_MODE	PT	TRG		TCM	
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
SCPWM	CE	PRE	PTV			AR	ST
R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	R/W-0h

Table 19-44. DMTIMER_1MS_TCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	Reads return 0
14	GPO_CFG	R/W	0h	
13	CAPT_MODE	R/W	0h	Capture mode select bit (first/second) 0h (R/W) = Capture the first enabled capture event in TCAR1 1h (R/W) = Capture the second enabled capture event in TCAR2
12	PT	R/W	0h	Pulse or Toggle select bit 0h (R/W) = pulse modulation 1h (R/W) = toggle modulation
11-10	TRG	R/W	0h	Trigger Output Mode 0h (R/W) = No trigger 1h (R/W) = Overflow trigger 2h (R/W) = Overflow and match trigger 3h (R/W) = Reserved
9-8	TCM	R/W	0h	Transition Capture Mode 0h (R/W) = No capture 1h (R/W) = Capture on rising edges of PIEVETCAPT 2h (R/W) = Capture on falling edges of PIEVETCAPT 3h (R/W) = Capture on booth edges of PIEVETCAPT
7	SCPWM	R/W	0h	Pulse Width Modulation output pin default value 0h (R/W) = default value of PORPWM: 0 1h (R/W) = default value of PORPWM: 1
6	CE	R/W	0h	Compare enable 0h (R/W) = Compare disabled 1h (R/W) = Compare enabled
5	PRE	R/W	0h	Prescaler enable 0h (R/W) = Prescaler disabled 1h (R/W) = Prescaler enabled
4-2	PTV	R/W	0h	Trigger Output Mode

Table 19-44. DMTIMER_1MS_TCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	AR	R/W	0h	Auto-reload mode 0h (R/W) = One shot mode overflow 1h (R/W) = Auto-reload mode overflow
0	ST	R/W	0h	Start/Stop timer control 0h (R/W) = Stop the timer 1h (R/W) = Start the timer

19.2.4.8 DMTIMER_1MS_TCRR Register (offset = 28h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_1MS_TCRR is shown in Figure 19-45 and described in Table 19-45.

This register holds the value of the internal counter

Figure 19-45. DMTIMER_1MS_TCRR Register

31–0
TIMER_CTR
R/W-0h

Table 19-45. DMTIMER_1MS_TCRR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TIMER_CTR	R/W	0h	The value of the timer counter register

19.2.4.9 DMTIMER_1MS_TLDR Register (offset = 2Ch) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_1MS_TLDR is shown in Figure 19-46 and described in Table 19-46.

This register holds the timer's load value

Figure 19-46. DMTIMER_1MS_TLDR Register

31–0
LOAD_VALUE
R/W-0h

Table 19-46. DMTIMER_1MS_TLDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LOAD_VALUE	R/W	0h	The value of the timer load register

19.2.4.10 DMTIMER_1MS_TTGR Register (offset = 30h) [reset = FFFFFFFFh]

Register mask: FFFFFFFFh

DMTIMER_1MS_TTGR is shown in Figure 19-47 and described in Table 19-47.

This register triggers a counter reload of timer by writing any value in it.

Figure 19-47. DMTIMER_1MS_TTGR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTGR_VALUE																															
Rreturns1s/W-FFFFFFFFh																															

Table 19-47. DMTIMER_1MS_TTGR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TTGR_VALUE	Rreturns1s/ W	FFFFFFFFh	The value of the trigger register During reads, it always returns "0xFFFFFFFF"

19.2.4.11 DMTIMER_1MS_TWPS Register (offset = 34h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_1MS_TWPS is shown in Figure 19-48 and described in Table 19-48.

This register contains the write posting bits for all writ-able functional registers

Figure 19-48. DMTIMER_1MS_TWPS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED						W_PEND_TOWR	W_PEND_TOCR
R-0h						R-0h	R-0h
7	6	5	4	3	2	1	0
W_PEND_TCVR	W_PEND_TNIR	W_PEND_TPIR	W_PEND_TMAR	W_PEND_TTGR	W_PEND_TLDR	W_PEND_TCRR	W_PEND_TCLR
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 19-48. DMTIMER_1MS_TWPS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	Reads return 0
9	W_PEND_TOWR	R	0h	Write pending for register TOWR 0h (R) = No Overflow Wrapping Register write pending. 1h (R) = Overflow Wrapping Register write pending.
8	W_PEND_TOCR	R	0h	Write pending for register TOCR 0h (R) = No Overflow Counter Register write pending. 1h (R) = Overflow Counter Register write pending.
7	W_PEND_TCVR	R	0h	Write pending for register TCVR 0h (R) = No Counter Register write pending. 1h (R) = Counter Register write pending.
6	W_PEND_TNIR	R	0h	Write pending for register TNIR 0h (R) = No Negativ Increment Register write pending. 1h (R) = Negativ Increment Register write pending.
5	W_PEND_TPIR	R	0h	Write pending for register TPIR 0h (R) = No Positive Increment Register write pending. 1h (R) = Positive Increment Register write pending.
4	W_PEND_TMAR	R	0h	Write pending for register TMAR 0h (R) = No Match Register write pending 1h (R) = Match Register write pending
3	W_PEND_TTGR	R	0h	Write pending for register TTGR 0h (R) = No Trigger Register write pending 1h (R) = Trigger Register write pending
2	W_PEND_TLDR	R	0h	Write pending for register TLDR 0h (R) = No Load Register write pending 1h (R) = Load Register write pending
1	W_PEND_TCRR	R	0h	Write pending for register TCRR 0h (R) = No Counter Register write pending 1h (R) = Counter Register write pending

Table 19-48. DMTIMER_1MS_TWPS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	W_PEND_TCLR	R	0h	Write pending for register TCLR 0h (R) = No Control Register write pending 1h (R) = Control Register write pending

19.2.4.12 DMTIMER_1MS_TMAR Register (offset = 38h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_1MS_TMAR is shown in Figure 19-49 and described in Table 19-49.

This register holds the match value to be compared with the counter's value

Figure 19-49. DMTIMER_1MS_TMAR Register

31-0
COMPARE_VALUE
R/W-0h

Table 19-49. DMTIMER_1MS_TMAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMPARE_VALUE	R/W	0h	The value of the match register

19.2.4.13 DMTIMER_1MS_TCAR1 Register (offset = 3Ch) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_1MS_TCAR1 is shown in Figure 19-50 and described in Table 19-50.

This register holds the value of the first counter register capture

Figure 19-50. DMTIMER_1MS_TCAR1 Register

31-0
CAPTURE_VALUE1
R-0h

Table 19-50. DMTIMER_1MS_TCAR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPTURE_VALUE1	R	0h	The value of first captured counter register

19.2.4.14 DMTIMER_1MS_TSICR Register (offset = 40h) [reset = 0h]

Register mask: FFFFFFFBh

DMTIMER_1MS_TSICR is shown in Figure 19-51 and described in Table 19-51.

Timer Synchronous Interface Control Register

Figure 19-51. DMTIMER_1MS_TSICR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					POSTED	SFT	RESERVED
R-0h					R/W-X	Rreturns0s/W-0h	R-0h

Table 19-51. DMTIMER_1MS_TSICR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reads return 0
2	POSTED	R/W	X	PIFREQRATIO 0h (R/W) = Posted mode inactive: will delay the command accept output signal. Note: This mode is not recommended on this device. 1h (R/W) = Posted mode active (clocks ratio needs to fit freq (timer) less than freq (OCP)/4 frequency requirement).
1	SFT	Rreturns0s/W	0h	This bit reset all the functional part of the module 0h (R/W) = software reset is disabled 1h (R/W) = software reset is enabled
0	RESERVED	R	0h	Reads return 0

19.2.4.15 DMTIMER_1MS_TCAR2 Register (offset = 44h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_1MS_TCAR2 is shown in Figure 19-52 and described in Table 19-52.

This register holds the value of the second counter register capture

Figure 19-52. DMTIMER_1MS_TCAR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_VALUE2																															
R-0h																															

Table 19-52. DMTIMER_1MS_TCAR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPTURE_VALUE2	R	0h	The value of second captured counter register

19.2.4.16 DMTIMER_1MS_TPIR Register (offset = 48h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_1MS_TPIR is shown in Figure 19-53 and described in Table 19-53.

This register is used for 1ms tick generation. The TPIR register holds the value of the positive increment. The value of this register is added with the value of the TCVR to define whether next value loaded in TCRR will be the sub-period value or the over-period value.

Figure 19-53. DMTIMER_1MS_TPIR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POSITIVE_INC_VALUE																															
R/W-0h																															

Table 19-53. DMTIMER_1MS_TPIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	POSITIVE_INC_VALUE	R/W	0h	The value of the positive increment.

19.2.4.17 DMTIMER_1MS_TNIR Register (offset = 4Ch) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_1MS_TNIR is shown in Figure 19-54 and described in Table 19-54.

This register is used for 1ms tick generation. The TNIR register holds the value of the negative increment. The value of this register is added with the value of the TCVR to define whether next value loaded in TCRR will be the sub-period value or the over-period value.

Figure 19-54. DMTIMER_1MS_TNIR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEGATIVE_INV_VALUE																															
R/W-0h																															

Table 19-54. DMTIMER_1MS_TNIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NEGATIVE_INV_VALUE	R/W	0h	The value of the negative increment.

19.2.4.18 DMTIMER_1MS_TCVR Register (offset = 50h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_1MS_TCVR is shown in Figure 19-55 and described in Table 19-55.

This register is used for 1ms tick generation. The TCVR register defines whether next value loaded in TCRR will be the sub-period value or the over-period value.

Figure 19-55. DMTIMER_1MS_TCVR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTR_VALUE																															
R/W-0h																															

Table 19-55. DMTIMER_1MS_TCVR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CTR_VALUE	R/W	0h	The value of CVR counter.

19.2.4.19 DMTIMER_1MS_TOCR Register (offset = 54h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_1MS_TOCR is shown in Figure 19-56 and described in Table 19-56.

This register is used to mask the tick interrupt for a selected number of ticks.

Figure 19-56. DMTIMER_1MS_TOCR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="24">OVF_CTR_VALUE</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="24">R/W-0h</td></tr>
</table>

Table 19-56. DMTIMER_1MS_TOCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reads return 0.
23-0	OVF_CTR_VALUE	R/W	0h	The number of overflow events.

19.2.4.20 DMTIMER_1MS_TOWR Register (offset = 58h) [reset = 0h]

Register mask: FFFFFFFFh

DMTIMER_1MS_TOWR is shown in Figure 19-57 and described in Table 19-57.

This register holds the number of masked overflow interrupts.

Figure 19-57. DMTIMER_1MS_TOWR Register

31-24	23-0
RESERVED	OVF_WRAPPING_VALUE
R-0h	R/W-0h

Table 19-57. DMTIMER_1MS_TOWR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reads return 0
23-0	OVF_WRAPPING_VALUE	R/W	0h	The number of masked interrupts.

19.3 Sync Timer (32k)

19.3.1 Introduction

The SyncTimer32K module is a 32-bit counter clocked by the falling edge of a 32-kHz clock. The timer is reset to its default value when its input pin, NRESPWRON, is active low (on a global cold reset). When NRESPWRON is released, and after three 32-kHz clock periods, the counter will start counting on the falling edge of the clock. When the highest count value is reached, the counter wraps to zero and continues counting without any extra delay. Counting can be temporarily stopped by asserting MSUSPEND (active low).

19.3.1.1 Sync Timer (32k) Features

The general features of the SyncTimer32K module are:

- Counter Register (32 bit) is an incremented counter
- Counter Register is incremented on falling edge of 32KHz clock.
- Counter Register can be cleared by NRESPWRON assertion.
- Counter Register can be stopped using MSUSPEND
- Hardware revision of the module can be read by OCP interface (ID_TIM).
- Counter Register value can be read by OCP interface with 32-bit OCP read access or 16-bit OCP access.
  - In case of 16-bit access, a shadow register will be updated on 16-bit LSB read command. This means that, to read a coherent value and avoid side effect, OCP 16-bit LSB command must be performed before OCP 16-bit MSB.
  - In addition, to keep coherency of the data, no interleaved accesses (from different OCP registers) are allowed.
- OCP 2.0 support
- WR and WRNP are supported and treated the same way

19.3.1.2 Unsupported Features

The SyncTimer32K module does not support the following features.

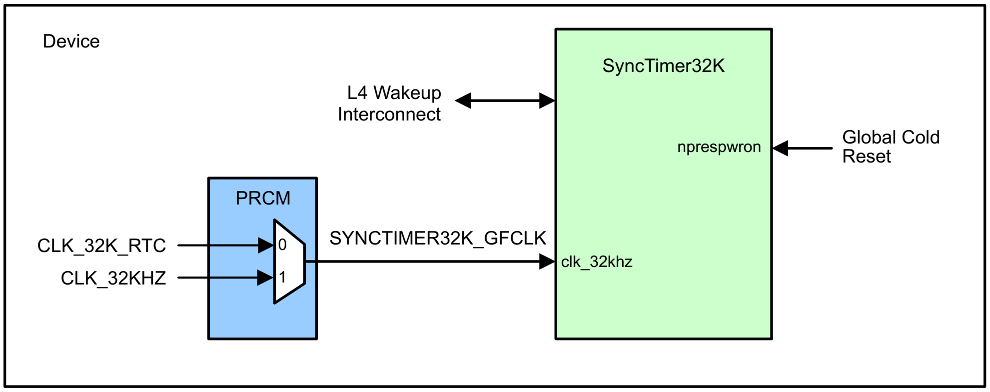
Table 19-58. Unsupported timer_32k Features

Feature	Reason
External event counter reset	Reset input not pinned out

19.3.2 Integration

This device contains a single SyncTimer32K module in the wake-up domain. Figure 19-58 shows the integration of the SyncTimer32K module in this device.

Figure 19-58. SyncTimer32K Integration



19.3.2.1 SyncTimer32K Connectivity Attributes

The general connectivity attributes for the SyncTimer32k module are shown in Table 19-59.

Table 19-59. timer_32k Connectivity Attributes

Attributes	Type
Power Domain	Wakeup domain
Clock Domain	PD_WKUP_L4_WKUP_AON_GCLK (OCP) SYNCTIMER32K_GFCLK (Functional)
Reset Signals	WKUP_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	None
DMA Requests	None
Physical Address	L4 wakeup slave port

19.3.2.2 SyncTimer32K Clock and Reset Management

The functional clock source (CLK_32KHZ input) is selected by the CLKSEL_SYNCTIMER_CLK register of the PRCM from:

- The RTC oscillator (CLK_32K_RTC)
- The PER PLL generated 32.768 KHz clock (CLK_32KHZ)

Table 19-60. SyncTimer32K Clock Signals

Clock Signal	Maximum Frequency	Reference Source	Comments
ocp_clk Interface clock	26 MHz	CLK_M_OSC	pd_wkup_l4_kwup_aon_gclk from PRCM
clk_32khz Functional clock	32.768 kHz	CLK_32K_RTC CLK_32KHZ (PER_CLKOUTM2 / 5859.375)	synctimer32k_gfclk from PRCM

19.3.2.3 SyncTimer32K Pin List

The SyncTimer32K has a single external interface signal.

Table 19-61. SyncTimer32K Pin List

Pin	Type	Description
NRESPWRON	I	Clear timer (when low)

19.3.3 Functional Description

The SyncTimer32K is a synchronized 32-bit counter clocked by the falling edge of a 32-kHz system clock. It is reset with asynchronous power on reset (NRESPWRON). When NRESPWRON is released, and after three 32-kHz clock periods, the counter starts counting from the reset value of the counter register on the falling edge of the 32-kHz clock. After reaching its maximum value, the counter wraps to zero and starts counting again.

19.3.3.1 OCP Interfacing

The SyncTimer32K has a 32-bit OCP interface. Due to the ocp_clk versus the clock of the peripheral frequency ratio, the ocp_clk is used to resynchronize the counter value for any ocp access.

The OCP bus interface handles:

- OCP bus peripheral address decoding
- Counter value read transaction synchronization to ensure the data correctness despite the asynchronous OCP and counter clocks
- OCP scmdaccept flag generation, synchronous with ocp_clk rising edge

19.3.3.2 Reading the Counter Register (CR)

Internal synchronization logic allows reading the counter value while the counter is running. Since the OCP is completely asynchronous with the 32-kHz clock, some synchronization is done, so as to make sure that the CR value is not read while it is being incremented. The synchronous logic ensures the read transaction correctness by synchronizing the counter register read access on the ocp_clk clock signal.

The counter register is a 32-bit “atomic datum” and has 16-bit capture. To read the value of CR correctly, the first OCP read access has to be to the lower 16-bit, followed by OCP read access to the upper 16-bit. Software performs a 32-bit access on the register while the device performs two consecutive 16-bit transactions. The time latency to read a synchronized register is one ocp_clk period.

In 16-bit mode, the following sequence must be followed to read the CR register properly:

a. Perform an OCP Read Transaction to read the lower 16-bit of the CR register.

   When the CR is read and synchronized, the lower 16-bit ‘LSB’ are driven onto the “ocp_sdata” bus and the upper 16-bit of the CR ‘MSB’ register are stored in a temporary register.

b. Perform an OCP Read Transaction to Read the upper 16-bit of the CR register.

   During this read, the value of the upper 16-bit ‘MSB’ that has been stored in a temporary register is forwarded onto the “ocp_sdata” bus.

19.3.3.3 Interrupt Control

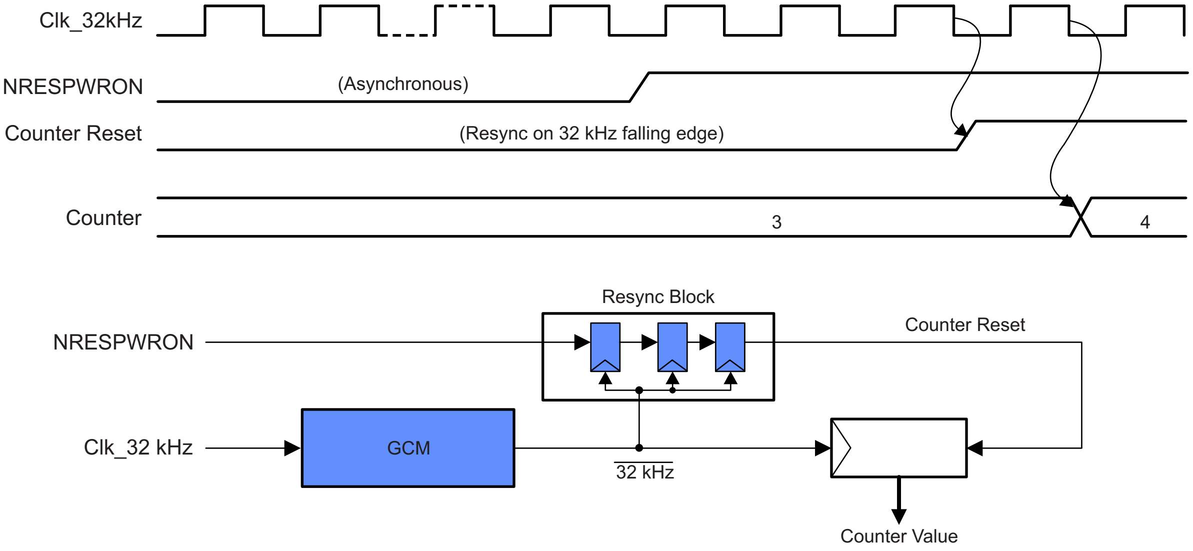
The SyncTimer32K has no interrupt outputs.

19.3.3.4 Reset

The NRESPWRON signal that is resynchronized three times on the rising edge of the inverted 32-kHz clock is used to synchronously reset the CR register.

The ocp_reset_n signal is used to asynchronously reset the OCP interface.

Figure 19-59. Reset Resynchronization Timing



19.3.4 SYNCTIMER Registers

Table 19-62 lists the memory-mapped registers for the SYNCTIMER. All register offset addresses not listed in Table 19-62 should be considered as reserved locations and the register contents should not be modified.

Table 19-62. SYNCTIMER Registers

Offset	Acronym	Register Name	Section
0h	SYNCTIMER32K_SYNCNT_REV		Section 19.3.4.1
4h	SYNCTIMER32K_SYSCONFIG		Section 19.3.4.2
10h	SYNCTIMER32K_CR		Section 19.3.4.3

19.3.4.1 SYNCTIMER32K_SYNCNT_REV Register (offset = 0h) [reset = 50h]

Register mask: FFFFFFFFh

SYNCTIMER32K_SYNCNT_REV is shown in Figure 19-60 and described in Table 19-63.

Figure 19-60. SYNCTIMER32K_SYNCNT_REV Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">RESERVED</td><td colspan="8">CID_REV</td></tr>
<tr><td colspan="24">Rreturns0s-0h</td><td colspan="8">R-50h</td></tr>
</table>

Table 19-63. SYNCTIMER32K_SYNCNT_REV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	Rreturns0s	0h	
7-0	CID_REV	R	50h	Revision number.

19.3.4.2 SYNCTIMER32K_SYSCONFIG Register (offset = 4h) [reset = 0h]

Register mask: FFFFFFFFh

SYNCTIMER32K_SYSCONFIG is shown in Figure 19-61 and described in Table 19-64.

Figure 19-61. SYNCTIMER32K_SYSCONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED			IDLEMODE		RESERVED		
R/W-0h			R/W-0h		R/W-0h		

Table 19-64. SYNCTIMER32K_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4-3	IDLEMODE	R/W	0h	Power Management Req/Ack Control. 0h = Force idle. An idle request is acknowledged unconditionally. 1h = No-idle. An idle request is never acknowledged. 2h = Reserved. 3h = Reserved.
2-0	RESERVED	R/W	0h	

19.3.4.3 SYNCTIMER32K_CR Register (offset = 10h) [reset = 3h]

Register mask: FFFFFFFFh

SYNCTIMER32K_CR is shown in Figure 19-62 and described in Table 19-65.

Figure 19-62. SYNCTIMER32K_CR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">CTR_HI</td><td colspan="16">CTR_LO</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">R-3h</td></tr>
</table>

Table 19-65. SYNCTIMER32K_CR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	CTR_HI	R	0h	Read counter high. Value of 32-kHz SYNCH counter (16 bit LSB).
15-0	CTR_LO	R	3h	Read counter low. Value of 32-kHz SYNCH counter (16 bit MSB).

19.4 Real-Time Clock (RTC)

19.4.1 Introduction

The real-time clock is a precise timer which can generate interrupts on intervals specified by the user. Interrupts can occur every second, minute, hour, or day. The clock itself can track the passage of real time for durations of several years, provided it has a sufficient power source the whole time.

The basic purpose for the RTC is to keep time of day. The other equally important purpose of RTC is for Digital Rights management. Some degree of tamper proofing is needed to ensure that simply stopping, resetting, or corrupting the RTC does not go unnoticed so that if this occurs, the application can re-acquire the time of day from a trusted source. The final purpose of the RTC is to wake the rest of chip up from a power down state.

Alarms are available to interrupt the CPU at a particular time, or at periodic time intervals, such as once per minute or once per day. In addition, the RTC can interrupt the CPU every time the calendar and time registers are updated, or at programmable periodic intervals.

19.4.1.1 Features

The real-time clock (RTC) provides the following features:

- 100-year calendar (xx00 to xx99)
- Counts seconds, minutes, hours, day of the week, date, month, and year with leap year compensation
- Binary-coded-decimal (BCD) representation of time, calendar, and alarm
- 12-hour clock mode (with AM and PM) or 24-hour clock mode
- Alarm interrupt
- Periodic interrupt
- Single interrupt to the CPU
- Supports external 32.768-kHz crystal or external clock source of the same frequency

19.4.1.2 Unsupported RTC Features

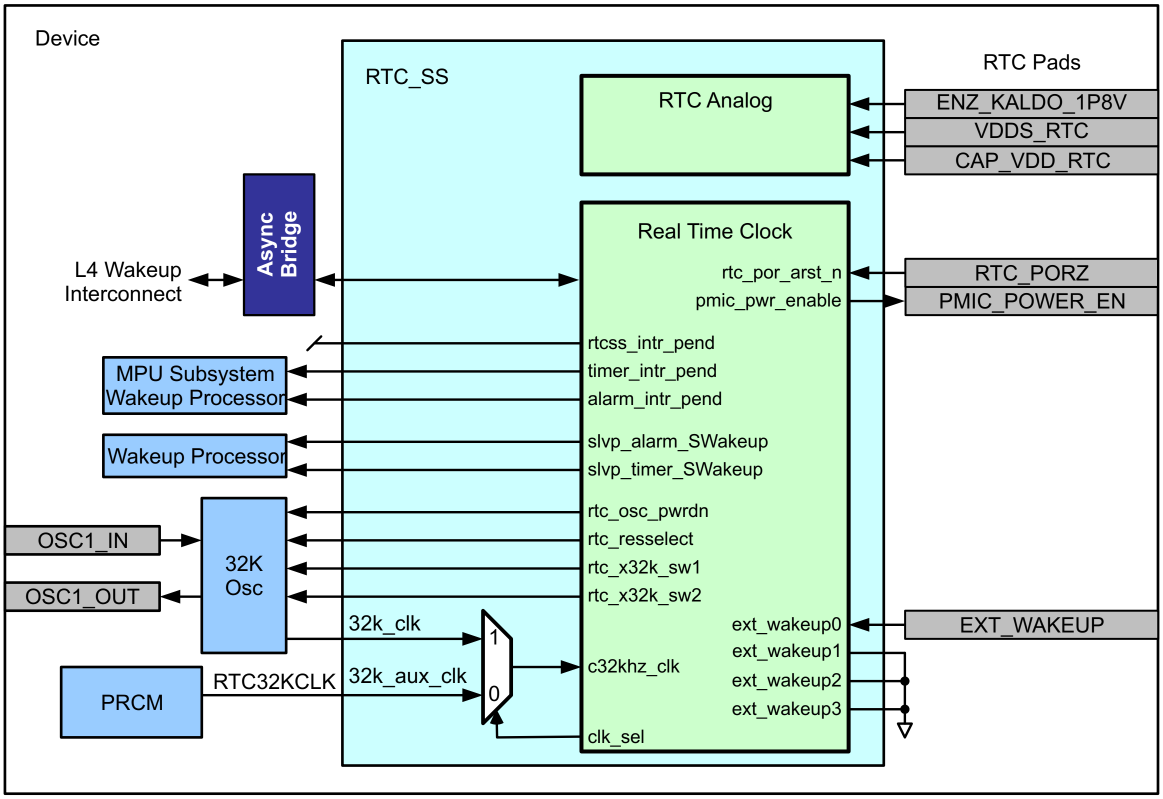
This device supports only a single RTC external wake-up event.

19.4.2 Integration

This device includes a Real-Time Clock Subsystem (RTCSS) module to allow easy tracking of time and date and the generation of real time alarms.

The integration of the RTC is shown in .

Figure 19-63. RTC Integration



19.4.2.1 RTC Connectivity Attributes

The general connectivity for the RTC module in the device is shown in Table 19-66.

Table 19-66. RTC Module Connectivity Attributes

Attributes	Type
Power Domain	RTC
Clock Domain	PD_RTC_L4_RTC_GCLK (Interface/OCP) PD_RTC_RTC32KCLK (Func) CLK_32K_RTC (Func)
Reset Signals	RTC_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	4 Interrupts Alarm interrupt to MPU Subsystem (RTCINT) and Wakeup Processor Timer interrupt to MPU Subsystem (RTCALARMINT) and Wakeup Processor Alarm wakeup to Wakeup Processor Timer wakeup to Wakeup Processor
DMA Requests	None
Physical Address	L4 Wakeup slave port

19.4.2.2 RTC Clock and Reset Management

The RTC functional clock (c32khz_clk input) is sourced by default from the CLK32_KHZ clock derived from the Peripheral PLL. It can also be sourced from the 32-KHz oscillator through a clock mux within and controlled by the RTC_SS.

Table 19-67. RTC Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
ocp_clk (Interface clock)	26 MHz	CLK_M_OSC	pd_rtc_l4_rtc_gclk From PRCM
rtc_32K_clk_rtc_32k_clk (Oscillator functional clock)	32.768 KHz	OSC1_IN	CLK_32K_RTC From OSC1_IN
rtc_32k_clk_rtc_32k_aux_clk (Internal functional clock)	32.768 KHz	PER_CLKOUTM2 / 5859.3752	pd_rtc_rtc_32kclk From PRCM

19.4.2.3 RTC Pin List

The RTC module does not include any external interface pins.

Table 19-68. RTC Pin List

Pin	Type	Description
RTC_PORz	I	RTC Power On Reset
EXT_WAKEUP	I	External wakeup
PMIC_POWER_EN	O	Power enable control for external power management IC
Analog Signals		
ENZ_KALDO_1P8V	I	Enable 1.8V LDO
VDDS_RTC	P	1.8V Voltage Supply
CAP_VDD_RTC	A	Decoupling Cap when internal 1.8 V LDO is enabled, 1.1 V supply when internal 1.8 V LDO is disabled.

19.4.3 Functional Description

This section defines the module interrupt capabilities and requirements.

19.4.3.1 Functional Block Diagram

Figure 19-64 shows the RTC module block diagram. Figure 19-65 shows a functional block diagram of the RTC.

Figure 19-64. RTC Block Diagram

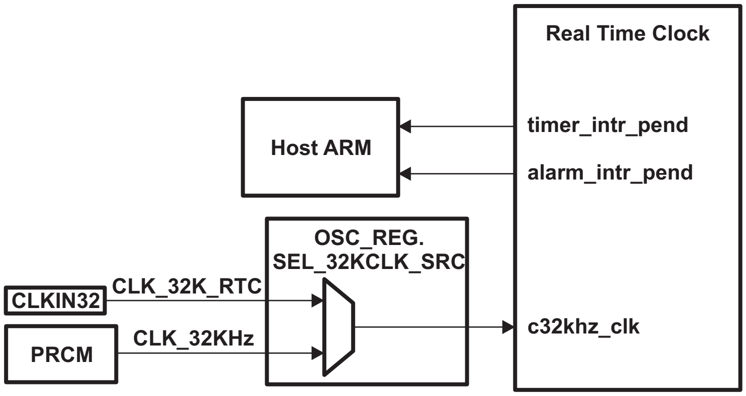
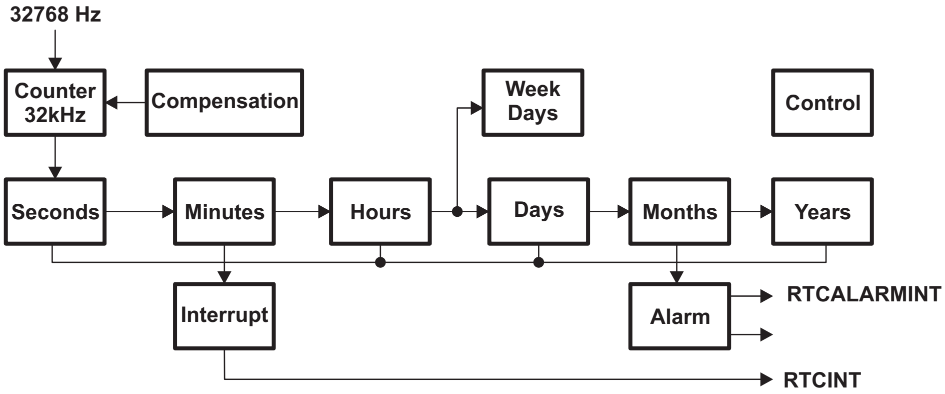


Figure 19-65. RTC Functional Block Diagram



19.4.3.2 Clock Source

The clock reference for the RTC can be sourced from an external crystal (used with the 32K RTC Oscillator), an external 32KHz oscillator, or from the Peripheral PLL. The RTC has an internal oscillator buffer to support direct operation with a crystal. The crystal is connected between pins RTC_XTALIN and RTC_XTALOUT. RTC_XTALIN is the input to the on-chip oscillator and RTC_XTALOUT is the output from the oscillator back to the crystal. The oscillator can be enabled or disabled by using the RTC_OSC_REG register. For more information about the RTC crystal connection, see your device-specific data manual.

An external 32.768-kHz clock oscillator may be used instead of a crystal. In such a case, the clock source is connected to RTC_XTALIN, and RTC_XTALOUT is left unconnected.

The source of the 32-KHz clock is selected using the OSC_CLK.SEL_32KCLK_SRC bit.

If the RTC is not used, the RTC_XTALIN pin should be held low and RTC_XTALOUT should be left unconnected. The RTC_disable bit in the control register (CTRL_REG) can be set to save power; however, the RTC_disable bit should not be cleared once it has been set. If the application requires the RTC module to stop and continue, the STOP_RTC bit in CTRL_REG should be used instead.

19.4.3.3 Signal Descriptions

Table 19-69 lists the signals and their descriptions for the RTC.

Table 19-69. RTC Signals

Signal	I/O	Description
RTC_XTALIN	I	RTC time base input signal. RTC_XTALIN can either be driven with a 32.768-kHz reference clock, or RTC_XTALIN and RTC_XTALOUT can be connected to an external crystal. This signal is the input to the RTC internal oscillator.
RTC_XTALOUT	O	RTC time base output signal. RTC_XTALOUT is the output from the RTC internal oscillator. If a crystal is not used as the time base for RTC_XTALIN, RTC_XTALOUT should be left unconnected.

19.4.3.4 Interrupt Support

19.4.3.4.1 CPU Interrupts

The RTC generates two interrupt outputs:

- timer_intr (RTCINT) is a timer interrupt.
- alarm_intr (RTCALARMINT) is an alarm interrupt.

NOTE: Both interrupt outputs support high-level and high-pulse.

19.4.3.4.2 Interrupt Description

19.4.3.4.2.1 Timer Interrupt RTCINT (timer_intr)

The timer interrupt can be generated periodically: every second, every minute, every hour, or every day (see INTRS_REG[1:0] for a description of how to set this up). The IT_TIMER bit of the interrupt register enables this interrupt. The timer interrupt is active-low.

The RTC_STS_REG[5:2] are only updated at each new interrupt and occur according to Table 19-70. For example, bit 2 (SEC) will always be set when one second has passed. It will also be set when one minute has passed since the completion of one minute also marks the completion of one second (from 59 seconds to 60 seconds). The same holds true for hours and days: each of them will also correspond to the passing of a second.

Conversely, bit 5 (DAY) will always be set when a day has passed. It might also be set when an hour, minute, or second has passed. However, this only occurs when the elapsed hour, minute, or second corresponds to the start of a new day.

Table 19-70. Interrupt Trigger Events

	One day has passed	One hour has passed	One minute has passed	One second has passed
STS_REG[5] (DAY)	1	0/1 (1)	0/1 (1)	0/1 (1)
STS_REG[4] (HOUR)	1	1	0/1 (1)	0/1 (1)
STS_REG[3] (MIN)	1	1	1	0/1 (1)
STS_REG[2] (SEC)	1	1	1	1

(1) This event is only triggered when the elapsed time unit (for example, Day) corresponds to the passage of another unit (for example, Seconds). For example, when the clock ticks from 00:23:59:59 (days : hours : minutes : seconds) to 01:00:00:00.

19.4.3.4.2.2 Alarm Interrupt RTCALARMINT (alarm_intr)

The alarm interrupt can be generated when the time set into TC ALARM registers is exactly the same as in the TC registers. This interrupt is then generated if the IT_ALARM bit of the interrupts register is set. This interrupt is low-level sensitive. RTC_STS_REG[6] indicates that IRQ_ALARM_CHIP has occurred. This interrupt is disabled by writing ‘1’ into the RTC_STS_REG[6].

To set up an alarm:

- Modify the ALARM_SECONDS, ALARM_MINUTES, ALARM_HOURS, ALARM_DAY, ALARM_WEEK, ALARM_MONTH, and ALARM_YEAR registers to the exact time you want an alarm to generate.
- Set the IT_ALARM bit in the RTC_INTRS register to enable the alarm interrupt.

19.4.3.5 Programming/Usage Guide

19.4.3.5.1 Time/Calendar Data Format

The time and calendar data in the RTC is stored as binary-coded decimal (BCD) format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. Although most of the time/calendar registers have 4 bits assigned to each BCD digit, some of the register fields are shorter since the range of valid numbers may be limited. For example, only 3 bits are required to represent the day of the week (WEEKS_REG) since only BCD numbers 1 through 7 are required. The following time and calendar registers are supported (BCD Format):

Note that the ALARM registers which share the names above also share the same BCD formatting.

- SECOND - Second Count (00-59)
- MINUTE - Minute Count (00-59)
- HOUR - Hour Count (12HR: 01-12; 24HR: 00-23)
- DAY - Day of the Month Count (01-31)
- WEEK - Day of the Week (0-6: SUN = 0)
- MONTH - Month Count (01-12; JAN = 1)
- YEAR - Year Count (00-99)

19.4.3.5.2 Register Access

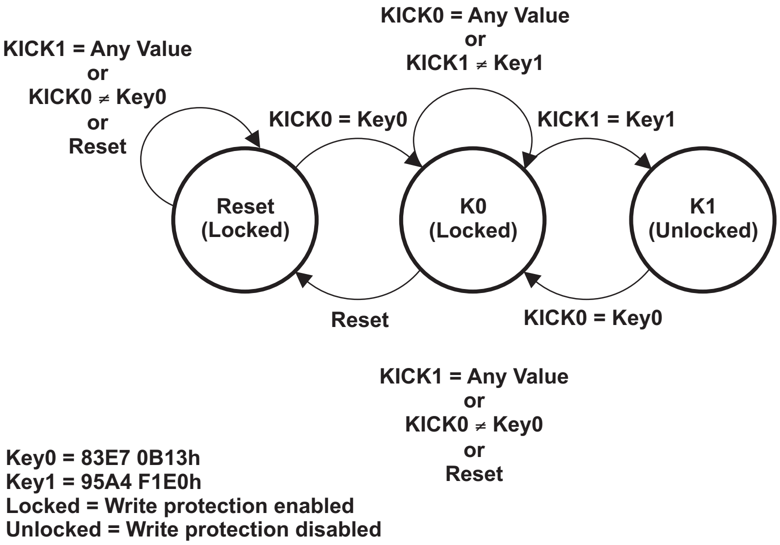
The three register types are as follows and each has its own access constraints:

- TC and TC alarm registers
- General registers
- Compensation registers

19.4.3.5.3 OCP MMR Spurious WRT Protection

The module also contains a kicker mechanism (Figure 19-66) to prevent any spurious writes from changing the register values. This mechanism requires two MMR writes to the Kick0 and Kick1 registers with exact data values before the kicker lock mechanism is released. Once released, the MMRs are writeable. The Kick0 data is 83E7 0B13h; the Kick1 data is 95A4 F1E0h. Note that it remains in an unlocked state until an OCP reset or invalid data pattern is written to one of the Kick0 or Kick1 registers.

Figure 19-66. Kick Register State Machine Diagram



- S0 is the Reset/Idle state
- S1 is an OCP wrt cycle of 83E7 0B13h at Kick0 completed state
- S2 is the UNLOCK MMR WRT state
- S0 -> S1 when OCP wrt cycle of 83E7 0B13h at Kick0
- S1 -> S2 when OCP wrt cycle of 95A4 F1E0h at Kick1
- S1 -> S0 when OCP reset event
- S2 -> S0 when OCP reset event OR OCP wrt cycle of NOT 83E7 0B13h at Kick0 OR OCP wrt cycle at Kick1
- S2 ->S1 when OCP wrt cycle of 83E7 0B13h at Kick0

19.4.3.5.4 Reading the Timer/Calendar (TC) Registers

The TC registers have a read-show register. The reading of the Seconds register will update all of the TC registers. For example, the Year will only get updated on a reading of the Seconds register. The time/calendar registers are updated every second as the time changes. During a read of the SECONDS register, the RTC copies the current values of the time/date registers into shadow read registers. This isolation assures that the CPU can capture all the time/date values at the moment of the SECONDS read request and not be subject to changing register values from time updates.

If desired, the RTC also provides a one-time-triggered minute-rounding feature to round the MINUTE:SECOND registers to the nearest minute (with zero seconds). This feature is enabled by setting the ROUND_30S bit in the control register (CTRL); the RTC automatically rounds the time values to the nearest minute upon the next read of the SECONDS register.

NOTE: Software should always read the Seconds register first. However, the software does not have to poll any status bit to determine when to read the TC registers. Table 19-71 defines the TC set that gets shadowed.

Table 19-71. RTC Register Names and Values

Time Unit	Range	Remarks
Year	00 to 99	
Month	01 to 12	
Day	01 to 31	Months 1, 3, 5, 7, 8, 10, 12
	01 to 30	Months 4, 6, 9, 11
	01 to 29	Month 2 (leap year)
	01 to 28	Month 2 (common year)
Week	00 to 06	Day of week
Hour	00 to 23	24 hour mode
	01 to 12	AM/PM mode
Minute	00 to 59	
Seconds	00 to 59	

19.4.3.5.4.1 Rounding Seconds

Time can be rounded to the closest minute, by setting the ROUND_30S bit of the control register. When this bit is set, TC values are set to the closest minute value at the next second. The ROUND_30S bit will be automatically cleared when rounding time is performed.

Example:

- If current time is 10H59M45S, round operation will change time to 11H00M00S.
- If current time is 10H59M29S, round operation will change time to 10H59M00S.

19.4.3.5.5 Modifying the TC Registers

To write correct data from/to the TC and TC alarm registers and read the TC alarm registers, the ARM must first read the BUSY bit of the STS register until BUSY is equal to zero. Once the BUSY flag is zero, there is a 15 µs access period in which the ARM can program the TC and TC alarm registers. Once the 15 µs access period passes, the BUSY flag has to be read again from the STS register as described previously. If the ARM accesses the TC registers outside of the access period, then the access is not guaranteed.

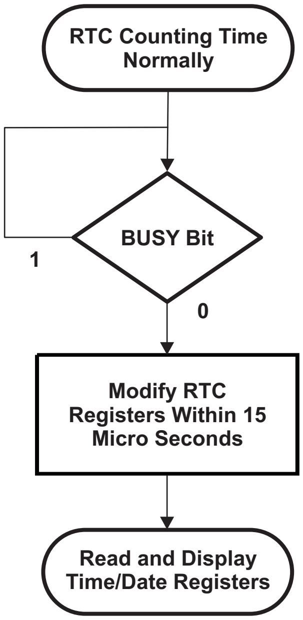
The ARM can access the STS_REG and CTRL_REG at any time, with the exception of CTRL_REG[5] which can only be changed when the RTC is stopped. The ARM can stop the RTC by clearing the STOP_RTC bit of the control register. After clearing this bit, the RUN bit in the STS_REG (bit 1) needs to be checked to verify the RTC has in fact stopped. Once this is confirmed, the TC values can be updated. After the values have been updated, the RTC can be re-started by resetting the STOP_RTC bit.

NOTE: After writing to a TC register, the user must wait 4 OCP clock cycles before reading the value from the register. If this wait time is not observed and the TC register is accessed, then old data will be read from the register.

CAUTION

In order to remove any possibility of interrupting the register's read process, thus introducing a potential risk of violating the authorized 15-microsecond access period, it is strongly recommended that you disable all incoming interrupts during the register read process.

Figure 19-67. Flow Control for Updating RTC Registers



19.4.3.5.5.1 General Registers

The ARM can access the STS_REG and the CTRL_REG at any time (except the CTRL_REG[5] bit which must be changed only when the RTC is stopped). For the INTRS_REG, the ARM should respect the available access period to prevent false interrupts.

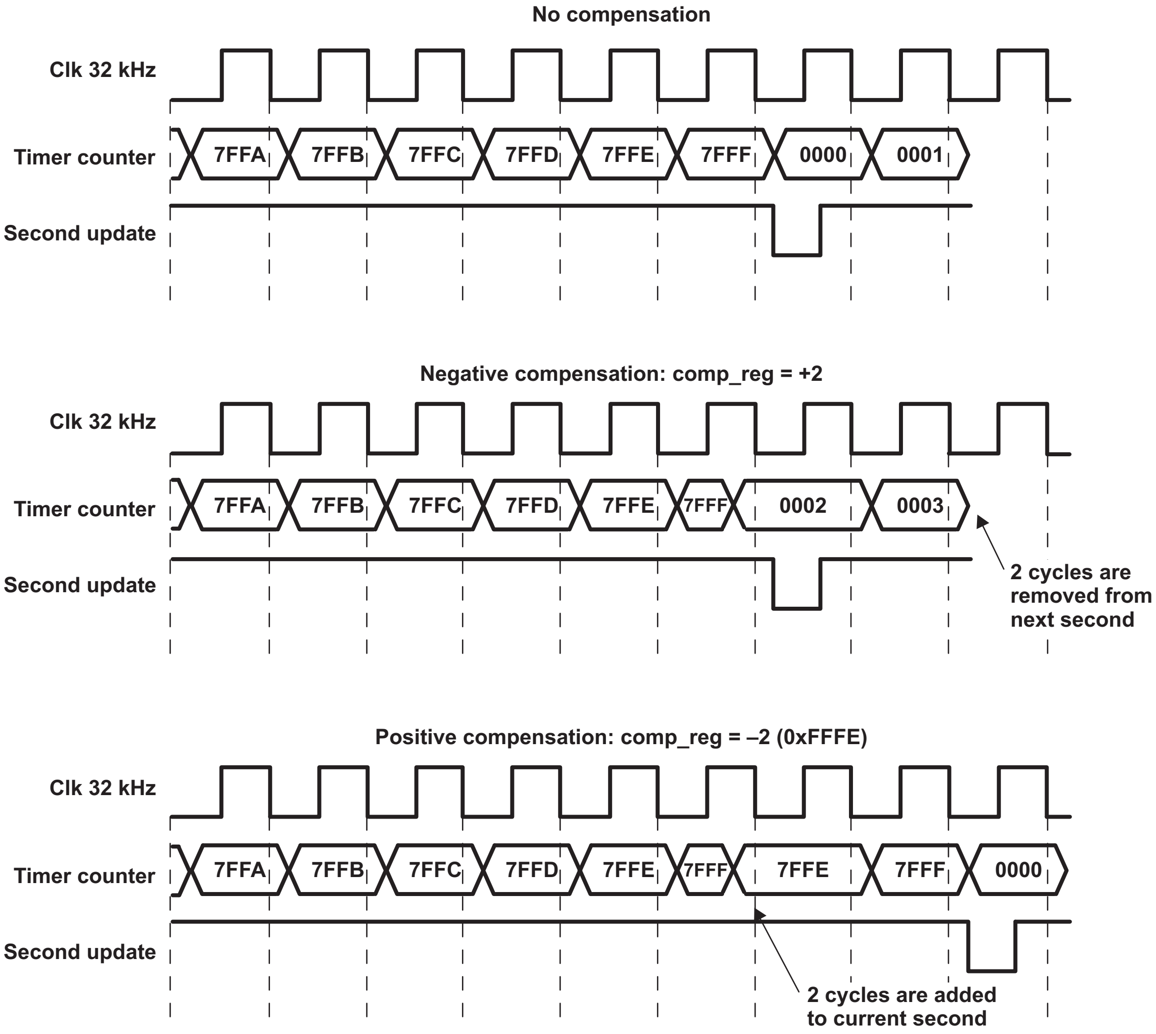
The RTC_DISABLE bit of the CTRL register must only be used to completely disable the RTC function. When this bit is set, the 32 kHz clock is gated, and the RTC is frozen. From this point, resetting this bit to zero can lead to unexpected behavior. In order to save power, this bit should only be used if the RTC function is unwanted in the application.

19.4.3.5.6 Crystal Compensation

To compensate for any inaccuracy of the 32 kHz oscillator, the ARM can perform a calibration of the oscillator frequency, calculate the drift compensation versus one-hour period, and load the compensation registers with the drift compensation value. Auto compensation is enabled by AUTO_COMP_EN bit in the RTC_CTRL register. If the COMP_REG value is positive, compensation occurs after the second change event. COMP_REG cycles are removed from the next second. If the COMP_REG value is negative, compensation occurs before the second change event. COMP_REG cycles are added to the current second. This enables compensation with a 1 32-kHz period accuracy each hour. The waveform below summarizes positive and negative compensation effect.

Access to the COMP_MSB_REG and COMP_LSB_REG registers must respect the available access period. These registers should not be updated during compensation (first second of each hour), but it is alright to update them during the second preceding a compensation event. For example, the ARM could load the compensation value into these registers after each hour event, during an available access period.

Figure 19-68. Compensation Illustration



19.4.3.6 Scratch Registers

The RTC provides three general-purpose registers (SCRATCHx_REG) that can be used to store 32-bit words -- these registers have no functional purpose for the RTC. Software using the RTC may find the SCRATCHx registers to be useful in indicating RTC states. For example, the SCRATCHx_REG registers may be used to indicate write-protection lock status or unintentional power downs. To indicate write-protection, the software should write a unique value to one of the SCRATCHx_REG registers when write-protection is disabled and another unique value when write-protection is enabled again. In this way, the lock-status of the registers can be determined quickly by reading the SCRATCH register. To indicate unintentional power downs, the software should write a unique value to one of the SCRATCHx_REG registers when RTC is configured and enabled. If the RTC is unintentionally powered down, the value written to the SCRATCH register is cleared. For more information, see Section 19.4.5, *RTC Registers*.

19.4.3.7 Power Management

The RTC supports the power idle protocol. It has two SWakeup ports: one for the alarm event and one for a timer event.

When the RTC is in IDLE mode, the OCP clock is turned off and the 32 kHz clock remains on. The time and calendar continue to count in IDLE mode. When the RTC is placed back in FUNCTIONAL mode, the TC registers can be read.

The Alarm SWakeup event can be used to wakeup the RTC when it is in IDLE state. In order to do so, the alarm needs to be set and enabled before RTC enters the IDLE state. Once this is done, the SWakeup will occur when the alarm event triggers.

NOTE: Since SWakeup is not periodic, using it to wake up the RTC when in IDLE state is not recommended. Please use Alarm SWakeup instead.

19.4.3.8 Power Management—System Level (PMIC Mode)

The RTC genertates pmic_power_en control which can be used to control an external PMIC.

Table 19-72. pmic_power_en Description

Port	Direction	Function
rtc_pwronrstn	Input	Not optional. RTC true power on domain reset. Only assert when RTC has lost power. Always de-assert when RTC voltage is greater than Vmin. The port remains de-asserted during normal operations.
pmic_power_en	Output	Optional. Can be used to control an external PMIC. 0 = OFF 1 = ON (reset state) ON → OFF (Turn OFF) By ALARM2 event OFF → ON (Turn ON) By ALARM event OR ext_wakeup event
ext_wakeup	Input	

19.4.4 Use Cases

The following list includes high-level steps to start using the RTC:

1. Enable the module clock domains (for details on which clock domain, see Section 19.4.2, *Integration*).
2. Enable the RTC module using CTRL_REG.RTC_disable.
3. Enable the 32K clock from PER PLL, if using the internal RTC oscillator.
4. Write to the kick registers (KICK0R, KICK1R) in the RTC.
5. Configure the timer in RTCSS for desired application (set time and date, alarm wakeup, and so on).
6. Start the RTC (in CTRL_REG.STOP_RTC).

19.4.5 RTC Registers

Table 19-73 lists the memory-mapped registers for the RTC. All register offset addresses not listed in Table 19-73 should be considered as reserved locations and the register contents should not be modified.

Table 19-73. RTC Registers

Offset	Acronym	Register Name	Section
0h	RTCSS_SECONDS_REG	Seconds Register	Section 19.4.5.1
4h	RTCSS_MINUTES_REG	Minutes Register	Section 19.4.5.2
8h	RTCSS_HOURS_REG	Hours Register	Section 19.4.5.3
Ch	RTCSS_DAYS_REG	Day of the Month Register	Section 19.4.5.4
10h	RTCSS_MONTHS_REG	Month Register	Section 19.4.5.5
14h	RTCSS_YEARS_REG	Year Register	Section 19.4.5.6
18h	RTCSS_WEEKS_REG	Day of the Week Register	Section 19.4.5.7
20h	RTCSS_ALARM_SECONDS_REG	Alarm Seconds Register	Section 19.4.5.8
24h	RTCSS_ALARM_MINUTES_REG	Alarm Minutes Register	Section 19.4.5.9
28h	RTCSS_ALARM_HOURS_REG	Alarm Hours Register	Section 19.4.5.10
2Ch	RTCSS_ALARM_DAYS_REG	Alarm Day of the Month Register	Section 19.4.5.11
30h	RTCSS_ALARM_MONTHS_REG	Alarm Months Register	Section 19.4.5.12
34h	RTCSS_ALARM_YEARS_REG	Alarm Years Register	Section 19.4.5.13
40h	RTCSS_CTRL_REG	Control Register	Section 19.4.5.14
44h	RTCSS_STS_REG	Status Register	Section 19.4.5.15
48h	RTCSS_INTRS_REG	Interrupt Enable Register	Section 19.4.5.16
4Ch	RTCSS_COMP_LSB_REG	Compensation (LSB) Register	Section 19.4.5.17
50h	RTCSS_COMP_MSB_REG	Compensation (MSB) Register	Section 19.4.5.18
54h	RTCSS_OSC_REG	Oscillator Register	Section 19.4.5.19
60h	RTCSS_SCRATCH0_REG	Scratch 0 Register (General-Purpose)	Section 19.4.5.20
64h	RTCSS_SCRATCH1_REG	Scratch 1 Register (General-Purpose)	Section 19.4.5.21
68h	RTCSS_SCRATCH2_REG	Scratch 2 Register (General-Purpose)	Section 19.4.5.22
6Ch	RTCSS_KICK0R	Kick 0 Register (Write Protect)	Section 19.4.5.23
70h	RTCSS_KICK1R	Kick 1 Register (Write Protect)	Section 19.4.5.24
74h	RTCSS_REVISION	Revision Register	Section 19.4.5.25
78h	RTCSS_SYSCONFIG	System Configuration Register	Section 19.4.5.26
7Ch	RTCSS_IRQWAKEEN	Wakeup Enable Register	Section 19.4.5.27
80h	RTCSS_ALARM2_SECONDS_REG	Alarm2 Seconds Register	Section 19.4.5.28
84h	RTCSS_ALARM2_MINUTES_REG	Alarm2 Minutes Register	Section 19.4.5.29
88h	RTCSS_ALARM2_HOURS_REG	Alarm2 Hours Register	Section 19.4.5.30
8Ch	RTCSS_ALARM2_DAYS_REG	Alarm2 Day of the Month Register	Section 19.4.5.31
90h	RTCSS_ALARM2_MONTHS_REG	Alarm2 Months Register	Section 19.4.5.32
94h	RTCSS_ALARM2_YEARS_REG	Alarm2 Years Register	Section 19.4.5.33
98h	RTCSS_PMIC	RTC PMIC Register	Section 19.4.5.34

Table 19-73. RTC Registers (continued)

Offset	Acronym	Register Name	Section
9Ch	RTCSS_DEBOUNCE	RTC Debounce Register	Section 19.4.5.35

19.4.5.1 RTCSS_SECONDS_REG Register (Offset = 0h) [reset = 0h]

RTCSS_SECONDS_REG is shown in Figure 19-69 and described in Table 19-74.

The SECONDS_REG is used to program the required seconds value of the current time. Seconds are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. If the seconds value is 45, then the value of SEC0 is 5 and value of SEC1 is 4.

Figure 19-69. RTCSS_SECONDS_REG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									SEC1			SEC0			
R-0h									R/W-0h			R/W-0h			

Table 19-74. RTCSS_SECONDS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6-4	SEC1	R/W	0h	2nd digit of seconds, Range is 0 to 5
3-0	SEC0	R/W	0h	1st digit of seconds, Range is 0 to 9

19.4.5.2 RTCSS_MINUTES_REG Register (Offset = 4h) [reset = 0h]

RTCSS_MINUTES_REG is shown in Figure 19-70 and described in Table 19-75.

The MINUTES_REG is used to program the minutes value of the current time. Minutes are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. If the minutes value is 32, then the value of MIN0 is 2 and value of MIN1 is 3.

Figure 19-70. RTCSS_MINUTES_REG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									MIN1			MIN0			
R-0h									R/W-0h			R/W-0h			

Table 19-75. RTCSS_MINUTES_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6-4	MIN1	R/W	0h	2nd digit of minutes, Range is 0 to 5
3-0	MIN0	R/W	0h	1st digit of minutes, Range is 0 to 9

19.4.5.3 RTCSS_HOURS_REG Register (Offset = 8h) [reset = 0h]

RTCSS_HOURS_REG is shown in Figure 19-71 and described in Table 19-76.

The HOURS_REG is used to program the hours value of the current time. Hours are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. In 24Hr time mode if you want to set the hour as 18, then HOUR0 is set as 8 and HOUR1 is set as 1.

Figure 19-71. RTCSS_HOURS_REG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
PM_NAM	RESERVED	HOUR1		HOUR0			
R/W-0h	R-0h	R/W-0h		R/W-0h			

Table 19-76. RTCSS_HOURS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	PM_NAM	R/W	0h	Only used in PM_AM mode (otherwise 0) 0h (R/W) = AM 1h (R/W) = PM
6	RESERVED	R	0h	
5-4	HOUR1	R/W	0h	2nd digit of hours, Range is 0 to 2
3-0	HOUR0	R/W	0h	1st digit of hours, Range is 0 to 9

19.4.5.4 RTCSS_DAYS_REG Register (Offset = Ch) [reset = 1h]

RTCSS_DAYS_REG is shown in Figure 19-72 and described in Table 19-77.

The DAYS_REG is used to program the day of the month value of the current date. Days are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. If the day value of the date is 28, DAY0 is set as 8 and DAY1 is set as 2.

Figure 19-72. RTCSS_DAYS_REG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										DAY1		DAY0			
R-0h										R/W-0h		R/W-1h			

Table 19-77. RTCSS_DAYS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5-4	DAY1	R/W	0h	2nd digit of days, Range is 0 to 3
3-0	DAY0	R/W	1h	1st digit of days, Range is 0 to 9

19.4.5.5 RTCSS_MONTHS_REG Register (Offset = 10h) [reset = 1h]

RTCSS_MONTHS_REG is shown in Figure 19-73 and described in Table 19-78.

The MONTHS_REG is used to set the month in the year value of the current date. Months are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. Usual notation is taken for month value: 1 = January, 2 = February, continuing until 12 = December.

Figure 19-73. RTCSS_MONTHS_REG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">RESERVED</td><td>MONTH1</td><td colspan="4">MONTH0</td></tr>
<tr><td colspan="3">R-0h</td><td>R/W-0h</td><td colspan="4">R/W-1h</td></tr>
</table>

Table 19-78. RTCSS_MONTHS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4	MONTH1	R/W	0h	2nd digit of months, Range is 0 to 1
3-0	MONTH0	R/W	1h	1st digit of months, Range is 0 to 9

19.4.5.6 RTCSS_YEARS_REG Register (Offset = 14h) [reset = 0h]

RTCSS_YEARS_REG is shown in Figure 19-74 and described in Table 19-79.

The YEARS_REG is used to program the year value of the current date. The year value is represented by only the last 2 digits and is stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. The year 1979 is programmed as 79 with YEAR0 set as 9 and YEAR1 set as 7.

Figure 19-74. RTCSS_YEARS_REG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="4">YEAR1</td><td colspan="4">YEAR0</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 19-79. RTCSS_YEARS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-4	YEAR1	R/W	0h	2nd digit of years, Range is 0 to 9
3-0	YEAR0	R/W	0h	1st digit of years, Range is 0 to 9

19.4.5.7 RTCSS_WEEKS_REG Register (Offset = 18h) [reset = 0h]

RTCSS_WEEKS_REG is shown in Figure 19-75 and described in Table 19-80.

The WEEKS_REG is used to program the day of the week value of the current date. The day of the week is stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. Sunday is treated as 0, Monday 1, and ending at Saturday with 6.

Figure 19-75. RTCSS_WEEKS_REG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													WEEK		
R-0h													R/W-0h		

Table 19-80. RTCSS_WEEKS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2-0	WEEK	R/W	0h	1st digit of days in a week, Range from 0 (Sunday) to 6 (Saturday)

19.4.5.8 RTCSS_ALARM_SECONDS_REG Register (Offset = 20h) [reset = 0h]

RTCSS_ALARM_SECONDS_REG is shown in Figure 19-76 and described in Table 19-81.

The ALARM_SECONDS_REG is used to program the second value for the alarm interrupt. Seconds are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

Figure 19-76. RTCSS_ALARM_SECONDS_REG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="9">RESERVED</td><td colspan="3">ALARMSEC1</td><td colspan="4">ALARMSEC0</td></tr>
<tr><td colspan="9">R-0h</td><td colspan="3">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 19-81. RTCSS_ALARM_SECONDS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6-4	ALARMSEC1	R/W	0h	2nd digit of seconds, Range is 0 to 5
3-0	ALARMSEC0	R/W	0h	1st digit of seconds, Range is 0 to 9

19.4.5.9 RTCSS_ALARM_MINUTES_REG Register (Offset = 24h) [reset = 0h]

RTCSS_ALARM_MINUTES_REG is shown in Figure 19-77 and described in Table 19-82.

The ALARM_MINUTES_REG is used to program the minute value for the alarm interrupt. Minutes are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

Figure 19-77. RTCSS_ALARM_MINUTES_REG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	ALARM_MIN1			ALARM_MIN0			
R-0h	R/W-0h			R/W-0h			

Table 19-82. RTCSS_ALARM_MINUTES_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6-4	ALARM_MIN1	R/W	0h	2nd digit of minutes, Range is 0 to 5
3-0	ALARM_MIN0	R/W	0h	1st digit of minutes, Range is 0 to 9

19.4.5.10 RTCSS_ALARM_HOURS_REG Register (Offset = 28h) [reset = 0h]

RTCSS_ALARM_HOURS_REG is shown in Figure 19-78 and described in Table 19-83.

The ALARM_HOURS_REG is used to program the hour value for the alarm interrupt. Hours are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

Figure 19-78. RTCSS_ALARM_HOURS_REG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
ALARM_PM_NAM	RESERVED	ALARM_HOUR1		ALARM_HOUR0			
R/W-0h	R-0h	R/W-0h		R/W-0h			

Table 19-83. RTCSS_ALARM_HOURS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	ALARM_PM_NAM	R/W	0h	Only used in PM_AM mode (otherwise 0) 0h (R/W) = AM 1h (R/W) = PM
6	RESERVED	R	0h	
5-4	ALARM_HOUR1	R/W	0h	2nd digit of hours, Range is 0 to 2
3-0	ALARM_HOUR0	R/W	0h	1st digit of hours, Range is 0 to 9

19.4.5.11 RTCSS_ALARM_DAYS_REG Register (Offset = 2Ch) [reset = 1h]

RTCSS_ALARM_DAYS_REG is shown in Figure 19-79 and described in Table 19-84.

The ALARM_DAYS_REG is used to program the day of the month value for the alarm interrupt. Days are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

Figure 19-79. RTCSS_ALARM_DAYS_REG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		ALARM_DAY1		ALARM_DAY0			
R-0h		R/W-0h		R/W-1h			

Table 19-84. RTCSS_ALARM_DAYS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5-4	ALARM_DAY1	R/W	0h	2nd digit for days, Range from 0 to 3
3-0	ALARM_DAY0	R/W	1h	1st digit for days, Range from 0 to 9

19.4.5.12 RTCSS_ALARM_MONTHS_REG Register (Offset = 30h) [reset = 1h]

RTCSS_ALARM_MONTHS_REG is shown in Figure 19-80 and described in Table 19-85.

The ALARM_MONTHS_REG is used to program the month in the year value for the alarm interrupt. The month is stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

Figure 19-80. RTCSS_ALARM_MONTHS_REG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">RESERVED</td><td>ALARM_MONTH1</td><td colspan="4">ALARM_MONTH0</td></tr>
<tr><td colspan="3">R-0h</td><td>R/W-0h</td><td colspan="4">R/W-1h</td></tr>
</table>

Table 19-85. RTCSS_ALARM_MONTHS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4	ALARM_MONTH1	R/W	0h	2nd digit of months, Range from 0 to 1
3-0	ALARM_MONTH0	R/W	1h	1st digit of months, Range from 0 to 9

19.4.5.13 RTCSS_ALARM_YEARS_REG Register (Offset = 34h) [reset = 0h]

RTCSS_ALARM_YEARS_REG is shown in Figure 19-81 and described in Table 19-86.

The ALARM_YEARS_REG is used to program the year for the alarm interrupt. Only the last two digits are used to represent the year and is stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

Figure 19-81. RTCSS_ALARM_YEARS_REG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED								ALARM_YEAR1				ALARM_YEAR0			
R-0h								R/W-0h				R/W-0h			

Table 19-86. RTCSS_ALARM_YEARS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-4	ALARM_YEAR1	R/W	0h	2nd digit of years, Range from 0 to 9
3-0	ALARM_YEAR0	R/W	0h	1st digit of years, Range from 0 to 9

19.4.5.14 RTCSS_CTRL_REG Register (Offset = 40h) [reset = 0h]

RTCSS_CTRL_REG is shown in Figure 19-82 and described in Table 19-87.

The RTC_CTRL_REG contains the controls to enable/disable the RTC, set the 12/24 hour time mode, to enable the 30 second rounding feature, and to STOP/START the RTC. The SET_32_COUNTER bit must only be used when the RTC is frozen. The RTC_DISABLE bit must only be used to completely disable the RTC function. When this bit is set, the 32 kHz clock is gated and the RTC is frozen. From this point, resetting this bit to zero can lead to unexpected behavior. This bit should only be used if the RTC function is unwanted in the application, in order to save power. MODE_12_24: It is possible to switch between the two modes at any time without disturbing the RTC. Read or write is always performed with the current mode. Auto compensation is enabled by the AUTO_COMP bit. If the COMP_REG value is positive, compensation occurs after the second change event. COMP_REG cycles are removed from the next second. If the COMP_REG value is negative, compensation occurs before the second change event. COMP_REG cycles are added to the current second. This enables it to compensate with one 32-kHz period accuracy each hour. The ROUND_30S bit is a toggle bit; the ARM can only write 1 and the RTC clears it. If the ARM sets the ROUND_30S bit and then reads it, the ARM reads 1 until the round-to-the-closest-minute is performed at the next second. The ARM can stop the RTC by clearing the STOP_RTC bit (owing to internal resynchronization, the RUN bit of the status register (STATUS_REG) must be checked to ensure that the RTC is frozen), then update TC values, and re-start the RTC by resetting the STOP_RTC bit.

Figure 19-82. RTCSS_CTRL_REG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	RTC_DISABLE	SET_32_CTR	TEST_MODE	MODE_12_24	AUTO_COMP	ROUND_30S	STOP_RTC
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 19-87. RTCSS_CTRL_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	RTC_DISABLE	R/W	0h	Disable RTC module and gate 32-kHz reference clock. 0h (R/W) = RTC enable 1h (R/W) = RTC disable (no 32 kHz clock)
5	SET_32_CTR	R/W	0h	Set the 32-kHz counter with the value stored in the compensation registers when the SET_32_COUNTER bit is set. 0h (R/W) = No action. 1h (R/W) = Set the 32Khz counter with compensation registers value
4	TEST_MODE	R/W	0h	Test mode. 0h (R/W) = Functional mode 1h (R/W) = Test mode (Auto compensation is enabled when the 32Khz counter reaches its end)
3	MODE_12_24	R/W	0h	Enable 12-hour mode for HOURS and ALARMHOURS registers. 0h (R/W) = 24-hr mode 1h (R/W) = 12-hour mode

Table 19-87. RTCSS_CTRL_REG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	AUTO_COMP	R/W	0h	Enable oscillator compensation mode. 0h (R/W) = No auto compensation 1h (R/W) = Auto compensation enabled
1	ROUND_30S	R/W	0h	Enable one-time rounding to nearest minute on next time register read. 0h (R/W) = No update 1h (R/W) = Time is rounded to the nearest minute
0	STOP_RTC	R/W	0h	Stop the RTC 32-kHz counter. 0h (R/W) = RTC is frozen 1h (R/W) = RTC is running

19.4.5.15 RTCSS_STS_REG Register (Offset = 44h) [reset = 0h]

RTCSS_STS_REG is shown in Figure 19-83 and described in Table 19-88.

The RTC_STATUS_REG contains bits that signal the status of interrupts, events to the processor. Status for the alarm interrupt and timer events are notified by the register. The alarm interrupt keeps its low level until the ARM writes 1 in the ALARM bit of the RTC_STATUS_REG register. ALARM2: This bit will indicate the status of the alarm interrupt. Writing a 1 to the bit clears the interrupt. ALARM: This bit will indicate the status of the alarm interrupt. Writing a 1 to the bit clears the interrupt. 1D_EVENT1: This bit will indicate if a day event has occurred. An interrupt will be generated to the processor based on the masking of the interrupt controller. 1H_EVENT1: This bit will indicate if an hour event has occurred. An interrupt will be generated to the processor based on the masking of the interrupt controller. 1M_EVENT1: This bit will indicate if a minute event has occurred. An interrupt will be generated to the processor based on the masking of the interrupt controller. 1S_EVENT1: This bit will indicate if a second event has occurred. An interrupt will be generated to the processor based on the masking of the interrupt controller. RUN: This bit will indicate if RTC is frozen or it is running. The RUN bit shows the real state of the RTC. Indeed, because the STOP_RTC signal is resynchronized on 32-kHz clock the action of this bit is delayed. BUSY: This bit will give the status of RTC module. The Time and alarm registers can be modified only when this bit is 0. The timer interrupt is a negative edge sensitive low-level pulse (1 OCP cycle duration).

Figure 19-83. RTCSS_STS_REG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
ALARM2	ALARM	1D_EVT	1H_EVT	1M_EVT	1S_EVT	RUN	BUSY
R/W-0h	R/W-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 19-88. RTCSS_STS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	ALARM2	R/W	0h	Indicates that an alarm2 interrupt has been generated. Software needs to wait 31 us before it clears this status to allow pmic_pwr_enable 1 ' 0 transmission.
6	ALARM	R/W	0h	Indicates that an alarm interrupt has been generated
5	1D_EVT	R	0h	One day has occurred
4	1H_EVT	R	0h	One hour has occurred
3	1M_EVT	R	0h	One minute has occurred
2	1S_EVT	R	0h	One second has occurred
1	RUN	R	0h	RTC is frozen or is running. 0h (R/W) = RTC is frozen 1h (R/W) = RTC is running
0	BUSY	R	0h	Status of RTC module. 0h (R/W) = Updating event in more than 15 s 1h (R/W) = Updating event

19.4.5.16 RTCSS_INTRS_REG Register (Offset = 48h) [reset = 0h]

RTCSS_INTRS_REG is shown in Figure 19-84 and described in Table 19-89.

The RTC_INTERRUPTS_REG is used to enable or disable the RTC from generating interrupts. The timer interrupt and alarm interrupt can be controlled using this register. The ARM must respect the BUSY period to prevent spurious interrupt. To set a period timer interrupt, the respective period value must be set in the EVERY field. For example, to set a periodic timer interrupt for every hour, the EVERY field has to be set to 2. Along with this the IT_TIMER bit also has to be set for the periodic interrupt to be generated. IT_ALARM bit has to be set to generate an alarm interrupt.

Figure 19-84. RTCSS_INTRS_REG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED			IT_ALARM2	IT_ALARM	IT_TIMER	EVERY	
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	

Table 19-89. RTCSS_INTRS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4	IT_ALARM2	R/W	0h	Enable one interrupt when the alarm value is reached (TC ALARM2 registers) by the TC registers 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled
3	IT_ALARM	R/W	0h	Enable one interrupt when the alarm value is reached (TC ALARM registers) by the TC registers 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled
2	IT_TIMER	R/W	0h	Enable periodic interrupt. 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled
1-0	EVERY	R/W	0h	Interrupt period. 0h (R/W) = Every second 1h (R/W) = Every minute 2h (R/W) = Every hour 3h (R/W) = Every day

19.4.5.17 RTCSS_COMP_LSB_REG Register (Offset = 4Ch) [reset = 0h]

RTCSS_COMP_LSB_REG is shown in Figure 19-85 and described in Table 19-90.

The COMP_LSB_REG is used to program the LSB value of the 32 kHz periods to be added to the 32 kHz counter every hour. This is used to compensate the oscillator drift. The COMP_LSB_REG works together with the compensation (MSB) register (COMP_MSB_REG). The AUTOCOMP bit in the control register (CTRL_REG) must be enabled for compensation to take place. This register must be written in two's complement. That means that to add one 32-kHz oscillator period every hour, the ARM must write FFFFh into RTC_COMP_MSB_REG and RTC_COMP_LSB_REG. To remove one 32-kHz oscillator period every hour, the ARM must write 0001h into RTC_COMP_MSB_REG and RTC_COMP_LSB_REG. The 7FFFh value is forbidden.

Figure 19-85. RTCSS_COMP_LSB_REG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RTC_COMP_LSB							
R-0h								R/W-0h							

Table 19-90. RTCSS_COMP_LSB_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	RTC_COMP_LSB	R/W	0h	Indicates number of 32-kHz periods to be added into the 32-kHz counter every hour

19.4.5.18 RTCSS_COMP_MSB_REG Register (Offset = 50h) [reset = 0h]

RTCSS_COMP_MSB_REG is shown in Figure 19-86 and described in Table 19-91.

The COMP_MSB_REG is used to program the MSB value of the 32 kHz periods to be added to the 32 kHz counter every hour. This is used to compensate the oscillator drift. The COMP_MSB_REG works together with the compensation (LSB) register (COMP_LSB_REG) to set the hourly oscillator compensation value. The AUTOCOMP bit in the control register (CTRL_REG) must be enabled for compensation to take place. This register must be written in two's complement. That means that to add one 32-kHz oscillator period every hour, the ARM must write FFFFh into RTC_COMP_MSB_REG and RTC_COMP_LSB_REG. To remove one 32-kHz oscillator period every hour, the ARM must write 0001h into RTC_COMP_MSB_REG and RTC_COMP_LSB_REG. The 7FFFh value is forbidden.

Figure 19-86. RTCSS_COMP_MSB_REG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">RTC_COMP_MSB</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 19-91. RTCSS_COMP_MSB_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	RTC_COMP_MSB	R/W	0h	Indicates number of 32-kHz periods to be added into the 32-kHz counter every hour

19.4.5.19 RTCSS_OSC_REG Register (Offset = 54h) [reset = 10h]

RTCSS_OSC_REG is shown in Figure 19-87 and described in Table 19-92.

Figure 19-87. RTCSS_OSC_REG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	32KCLK_EN	RESERVED	OSC32K_GZ	32KCLK_SEL	RES_SELECT	SW2	SW1
R-0h	R/W-0h	R-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 19-92. RTCSS_OSC_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	32KCLK_EN	R/W	0h	32-kHz clock enable post clock mux of rtc_32k_clk_rtc_32k_aux_clk and rtc_32k_clk_rtc_32k_clk 0h (R/W) = Disable clock mux 1h (R/W) = Enable clock mux
5	RESERVED	R	0h	
4	OSC32K_GZ	R/W	1h	Disable the oscillator and apply high impedance to the output 0h (R/W) = Enable 1h (R/W) = Disabled and high impedance
3	32KCLK_SEL	R/W	0h	32-kHz clock source select 0h (R/W) = Selects internal clock source, namely rtc_32k_clk_rtc_32k_aux_clk 1h (R/W) = Selects external clock source, namely rtc_32k_clk_rtc_32k_clk that is from the 32-kHz oscillator
2	RES_SELECT	R/W	0h	External feedback resistor 0h (R/W) = Internal 1h (R/W) = External
1	SW2	R/W	0h	Inverter size adjustment
0	SW1	R/W	0h	Inverter size adjustment

19.4.5.20 RTCSS_SCRATCH0_REG Register (Offset = 60h) [reset = 0h]

RTCSS_SCRATCH0_REG is shown in Figure 19-88 and described in Table 19-93.

The RTC_SCRATCH0_REG is used to hold some required values for the RTC register.

Figure 19-88. RTCSS_SCRATCH0_REG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCSCRATCH0																															
R/W-0h																															

Table 19-93. RTCSS_SCRATCH0_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTCSCRATCH0	R/W	0h	Scratch registers, available to program

19.4.5.21 RTCSS_SCRATCH1_REG Register (Offset = 64h) [reset = 0h]

RTCSS_SCRATCH1_REG is shown in Figure 19-89 and described in Table 19-94.

The RTC_SCRATCH1_REG is used to hold some required values for the RTC register.

Figure 19-89. RTCSS_SCRATCH1_REG Register

31-0
RTCSCRATCH1
R/W-0h

Table 19-94. RTCSS_SCRATCH1_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTCSCRATCH1	R/W	0h	Scratch registers, available to program

19.4.5.22 RTCSS_SCRATCH2_REG Register (Offset = 68h) [reset = 0h]

RTCSS_SCRATCH2_REG is shown in Figure 19-90 and described in Table 19-95.

The RTC_SCRATCH2_REG is used to hold some required values for the RTC register.

Figure 19-90. RTCSS_SCRATCH2_REG Register

31–0
RTCSCRATCH2
R/W-0h

Table 19-95. RTCSS_SCRATCH2_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTCSCRATCH2	R/W	0h	Scratch registers, available to program

19.4.5.23 RTCSS_KICK0R Register (Offset = 6Ch) [reset = 0h]

RTCSS_KICK0R is shown in Figure 19-91 and described in Table 19-96.

The kick registers (KICKnR) are used to enable and disable write protection on the RTC registers. Out of reset, the RTC registers are write-protected. To disable write protection, correct keys must be written to the KICKnR registers. The Kick0 register allows writing to unlock the kick0 data. To disable RTC register write protection, the value of 83E7 0B13h must be written to KICK0R, followed by the value of 95A4 F1E0h written to KICK1R. RTC register write protection is enabled when any value is written to KICK0R.

Figure 19-91. RTCSS_KICK0R Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">KICK0</td></tr>
<tr><td colspan="32">W-0h</td></tr>
</table>

Table 19-96. RTCSS_KICK0R Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	KICK0	W	0h	Kick0 data

19.4.5.24 RTCSS_KICK1R Register (Offset = 70h) [reset = 0h]

RTCSS_KICK1R is shown in Figure 19-92 and described in Table 19-97.

The kick registers (KICKnR) are used to enable and disable write protection on the RTC registers. Out of reset, the RTC registers are write-protected. To disable write protection, correct keys must be written to the KICKnR registers. The Kick1 register allows writing to unlock the kick1 data and the kicker mechanism to write to other MMRs. To disable RTC register write protection, the value of 83E7 0B13h must be written to KICK0R, followed by the value of 95A4 F1E0h written to KICK1R.

Figure 19-92. RTCSS_KICK1R Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">KICK1</td></tr>
<tr><td colspan="32">W-0h</td></tr>
</table>

Table 19-97. RTCSS_KICK1R Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	KICK1	W	0h	Kick1 data

19.4.5.25 RTCSS_REVISION Register (Offset = 74h) [reset = 4EB00904h]

RTCSS_REVISION is shown in Figure 19-93 and described in Table 19-98.

Figure 19-93. RTCSS_REVISION Register

31	30	29	28	27	26	25	24
SCHEME		RESERVED		FUNC			
R-1h		R-0h		R-EB0h			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
FUNC							
R-EB0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
R_RTL					X_MAJOR		
R-1h					R-1h		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CUSTOM		Y_MINOR					
R-0h		R-4h					

Table 19-98. RTCSS_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	Used to distinguish between old scheme and current
29-28	RESERVED	R	0h	
27-16	FUNC	R	EB0h	Function indicates a software compatible module family
15-11	R_RTL	R	1h	RTL Version (R)
10-8	X_MAJOR	R	1h	Major Revision
7-6	CUSTOM	R	0h	Indicates a special version for a particular device
5-0	Y_MINOR	R	4h	Minor Revision (Y)

19.4.5.26 RTCSS_SYSCONFIG Register (Offset = 78h) [reset = 2h]

RTCSS_SYSCONFIG is shown in Figure 19-94 and described in Table 19-99.

Figure 19-94. RTCSS_SYSCONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						IDLEMODE	
R-0h						R/W-2h	

Table 19-99. RTCSS_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	IDLEMODE	R/W	2h	Configuration of the local target state management mode, By definition target can handle read/write transaction as long as it is out of IDLE state. 0h (R/W) = Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e., regardless of the IP module's internal requirements; Backup mode, for debug only. 1h (R/W) = No-idle mode: local target never enters idle state, Backup mode, for debug only. 2h (R/W) = Smart-idle mode: local target's state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements, IP module shall not generate (IRQ- or DMA-request-related) wakeup events. 3h (R/W) = Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges the system's idle requests, depending on the IP module's internal requirements, IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state.

19.4.5.27 RTCSS_IRQWAKEEN Register (Offset = 7Ch) [reset = 0h]

RTCSS_IRQWAKEEN is shown in Figure 19-95 and described in Table 19-100.

Figure 19-95. RTCSS_IRQWAKEEN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						ALARM_WAKEEN	TIMER_WAKEEN
R-0h						R/W-0h	R/W-0h

Table 19-100. RTCSS_IRQWAKEEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	ALARM_WAKEEN	R/W	0h	Wakeup generation for event Alarm. 0h (R/W) = Wakeup disabled 1h (R/W) = Wakeup enabled
0	TIMER_WAKEEN	R/W	0h	Wakeup generation for event Timer. 0h (R/W) = Wakeup disabled 1h (R/W) = Wakeup enabled

19.4.5.28 RTCSS_ALARM2_SECONDS_REG Register (Offset = 80h) [reset = 0h]

RTCSS_ALARM2_SECONDS_REG is shown in Figure 19-96 and described in Table 19-101.

The ALARM2_SECONDS_REG is used to program the second value for the alarm2 time. Seconds are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

Figure 19-96. RTCSS_ALARM2_SECONDS_REG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	ALARM2_SEC1			ALARM2_SEC0			
R-0h	R/W-0h			R/W-0h			

Table 19-101. RTCSS_ALARM2_SECONDS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6-4	ALARM2_SEC1	R/W	0h	2nd digit of seconds, Range is 0 to 5
3-0	ALARM2_SEC0	R/W	0h	1st digit of seconds, Range is 0 to 9

19.4.5.29 RTCSS_ALARM2_MINUTES_REG Register (Offset = 84h) [reset = 0h]

RTCSS_ALARM2_MINUTES_REG is shown in Figure 19-97 and described in Table 19-102.

The ALARM2_MINUTES_REG is used to program the minute value for the alarm2 time. Minutes are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

Figure 19-97. RTCSS_ALARM2_MINUTES_REG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	ALARM2_MIN1			ALARM2_MIN0			
R-0h	R/W-0h			R/W-0h			

Table 19-102. RTCSS_ALARM2_MINUTES_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6-4	ALARM2_MIN1	R/W	0h	2nd digit of minutes, Range is 0 to 5
3-0	ALARM2_MIN0	R/W	0h	1st digit of minutes, Range is 0 to 9

19.4.5.30 RTCSS_ALARM2_HOURS_REG Register (Offset = 88h) [reset = 0h]

RTCSS_ALARM2_HOURS_REG is shown in Figure 19-98 and described in Table 19-103.

The ALARM2_HOURS_REG is used to program the hour value for the alarm2 time. Hours are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

Figure 19-98. RTCSS_ALARM2_HOURS_REG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
ALARM2_PM_NAM	RESERVED	ALARM2_HOUR1		ALARM2_HOUR0			
R/W-0h	R-0h	R/W-0h		R/W-0h			

Table 19-103. RTCSS_ALARM2_HOURS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	ALARM2_PM_NAM	R/W	0h	Only used in PM_AM mode (otherwise 0) 0h (R/W) = AM 1h (R/W) = PM
6	RESERVED	R	0h	
5-4	ALARM2_HOUR1	R/W	0h	2nd digit of hours, Range is 0 to 2
3-0	ALARM2_HOUR0	R/W	0h	1st digit of hours, Range is 0 to 9

19.4.5.31 RTCSS_ALARM2_DAYS_REG Register (Offset = 8Ch) [reset = 1h]

RTCSS_ALARM2_DAYS_REG is shown in Figure 19-99 and described in Table 19-104.

The ALARM2_DAYS_REG is used to program the day of the month value for the alarm2 date. Days are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

Figure 19-99. RTCSS_ALARM2_DAYS_REG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED		ALARM2_DAY1		ALARM2_DAY0			
R-0h		R/W-0h		R/W-1h			

Table 19-104. RTCSS_ALARM2_DAYS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5-4	ALARM2_DAY1	R/W	0h	2nd digit for days, Range from 0 to 3
3-0	ALARM2_DAY0	R/W	1h	1st digit for days, Range from 0 to 9

19.4.5.32 RTCSS_ALARM2_MONTHS_REG Register (Offset = 90h) [reset = 1h]

RTCSS_ALARM2_MONTHS_REG is shown in Figure 19-100 and described in Table 19-105.

The ALARM2_MONTHS_REG is used to program the month in the year value for the alarm2 date. The month is stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

Figure 19-100. RTCSS_ALARM2_MONTHS_REG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">RESERVED</td><td>ALARM2_MONTH1</td><td colspan="4">ALARM2_MONTH0</td></tr>
<tr><td colspan="3">R-0h</td><td>R/W-0h</td><td colspan="4">R/W-1h</td></tr>
</table>

Table 19-105. RTCSS_ALARM2_MONTHS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4	ALARM2_MONTH1	R/W	0h	2nd digit of months, Range from 0 to 1
3-0	ALARM2_MONTH0	R/W	1h	1st digit of months, Range from 0 to 9

19.4.5.33 RTCSS_ALARM2_YEARS_REG Register (Offset = 94h) [reset = 0h]

RTCSS_ALARM2_YEARS_REG is shown in Figure 19-101 and described in Table 19-106.

The ALARM2_YEARS_REG is used to program the year for the alarm2 date. Only the last two digits are used to represent the year and stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

Figure 19-101. RTCSS_ALARM2_YEARS_REG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ALARM2_YEAR1				ALARM2_YEAR0			
R-0h								R/W-0h				R/W-0h			

Table 19-106. RTCSS_ALARM2_YEARS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-4	ALARM2_YEAR1	R/W	0h	2nd digit of years, Range from 0 to 9
3-0	ALARM2_YEAR0	R/W	0h	1st digit of years, Range from 0 to 9

19.4.5.34 RTCSS_PMIC Register (Offset = 98h) [reset = 0h]

RTCSS_PMIC is shown in Figure 19-102 and described in Table 19-107.

Figure 19-102. RTCSS_PMIC Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="5">RESERVED</td><td colspan="2">PWR_EN_SM</td><td>PWR_EN</td></tr>
<tr><td colspan="5">R-0h</td><td colspan="2">R-0h</td><td>R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="4">EXT_WAKEUP_STS</td><td colspan="4">EXT_WAKEUP_DB_EN</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">EXT_WAKEUP_POL</td><td colspan="4">EXT_WAKEUP_EN</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 19-107. RTCSS_PMIC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	
18-17	PWR_EN_SM	R	0h	Power state machine state. 0h (R/W) = Idle/Default 1h (R/W) = Shutdown (ALARM2 and PWR_ENABLE_EN is set to 1). Note: 31 us latency from ALARM2 event). 10h (R/W) = Time-based wakeup (ALARM status is set). 11h (R/W) = External-event-based wakeup (one or more bit set in EXT_WAKEUP_STATUS)
16	PWR_EN	R/W	0h	Enable for PMIC_POWER_EN signal 0h (R/W) = Disable. When Disabled, pmic_power_en signal will always be driven as 1, ON state. 1h (R/W) = Enable. When Enabled: pmic_power_en signal will be controlled by ext_wakeup, alarm, and alarm2; ON -> OFF (Turn OFF) only by ALARM2 event; OFF -> ON (TURN ON) only by ALARM event OR ext_wakeup event.
15-12	EXT_WAKEUP_STS	R/W	0h	External wakeup status. Write 1 to clear EXT_WAKEUP_STATUS[n] status of ext_wakeup[n]. 0h (R/W) = External wakeup event has not occurred 1h (R/W) = External wakeup event has occurred
11-8	EXT_WAKEUP_DB_EN	R/W	0h	External wakeup debounce enabled. EXT_WAKEUP_DB_EN[n] controls ext_wakeup[n] 0h (R/W) = Disable 1h (R/W) = Enable. When enabled, RTC_DEBOUNCE_REG defines the debounce time.
7-4	EXT_WAKEUP_POL	R/W	0h	External wakeup inputs polarity. EXT_WAKEUP_POL[n] controls ext_wakeup[n]. 0h (R/W) = Active high 1h (R/W) = Active low
3-0	EXT_WAKEUP_EN	R/W	0h	Enable external wakeup inputs. EXT_WAKEUP_EN[n] controls ext_wakeup[n]. 0h (R/W) = Ext. wakeup disabled 1h (R/W) = Ext. wakeup enabled

19.4.5.35 RTCSS_DEBOUNCE Register (Offset = 9Ch) [reset = 0h]

RTCSS_DEBOUNCE is shown in Figure 19-103 and described in Table 19-108.

The debounce timer uses the 32768-Hz clock. It allows choosing the timing or the accuracy of debouncing . A register receives a bit from the reference pin. You will choose the timing if you use the debouncing like a timer, or you will choose the accuracy if you use the debouncing like a real debouncing. The debouncing will be finished when the reference pin will stay the same value (defined in DEBOUNCE_REG) for a defined time.

Figure 19-103. RTCSS_DEBOUNCE Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">DEBOUNCE_REG</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 19-108. RTCSS_DEBOUNCE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	DEBOUNCE_REG	R/W	0h	Debounce time. A value, n, other than 0 results in a debounce time of 30.52 s*(n+1). 0h (R/W) = Debounce time is 30.52 s.

19.5 WATCHDOG

19.5.1 Introduction

The watchdog timer is an upward counter capable of generating a pulse on the reset pin and an interrupt to the device system modules following an overflow condition. The watchdog timer serves resets to the PRCM module and serves watchdog interrupts to the host ARM. The reset of the PRCM module causes a warm reset of the device.

The watchdog timer can be accessed, loaded, and cleared by registers through the L4 interface. The timer clock input is a 32-kHz clock.

The watchdog timer connects to a single target agent port on the L4 interconnect. The default state of the watchdog timer is enabled and not running.

19.5.1.1 Features

The main features of the watchdog timer controllers are:

- L4 slave interface support:
  - 32-bit data bus width
  - 32-/16-bit access supported
  - 8-bit access not supported
  - 11-bit address bus width
  - Burst mode not supported
  - Write nonposted transaction mode only
- Free-running 32-bit upward counter
- Programmable divider clock source (2n where n = 0-7)
- On-the-fly read/write register (while counting)
- Subset programming model of the GP timer
- The watchdog timers are reset either on power-on or after a warm reset before they start counting.
- Reset or interrupt actions when a timer overflow condition occurs
- The watchdog timer generates a reset or an interrupt in its hardware integration.

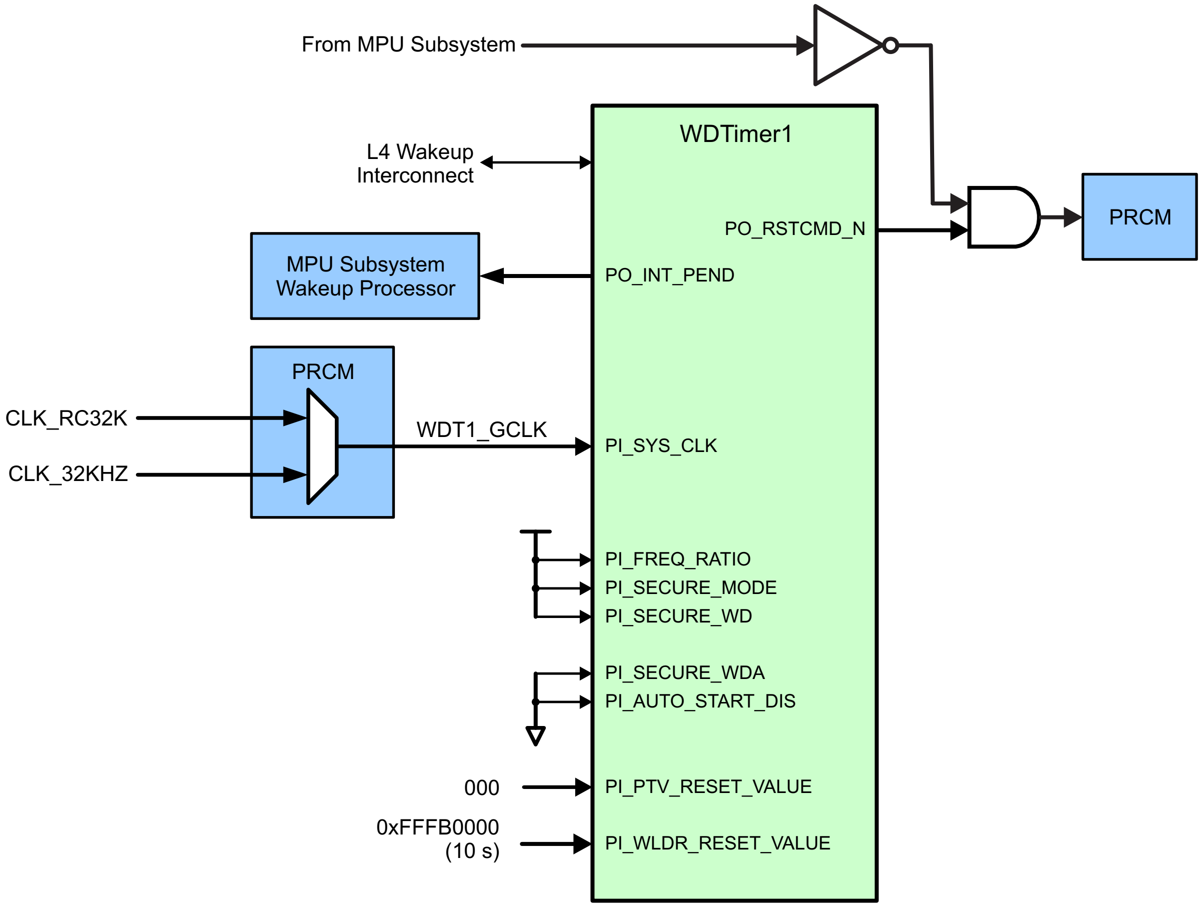
19.5.1.2 Unsupported Featres

There are no unsupported WD Timer features in this device.

19.5.2 Integration

The integration of the public WD Timer is shown in Figure 19-104.

Figure 19-104. Public WDTimer Integration



19.5.2.1 WD Timer Connectivity Attributes

The general connectivity for the public WD Timer module in this device is shown in Table 19-109.

Table 19-109. Public WD Timer Module Connectivity Attributes

Attributes	Type
Power Domain	Wakeup Domain
Clock Domain	PD_WKUP_L4_WKUP_GCLK (OCP) PD_WKUP_WDT1_GCLK (Func)
Reset Signals	WKUP_DOM_RST_N
Idle/Wakeup Signals	Smart Idle / Slave Wakeup
Interrupt Requests	1 Interrupt to MPU Subsystem (WDT1INT) and Wakeup Processor SWakeup Interrupt to Wakeup Processor
DMA Requests	None
Physical Address	L4 Wakeup slave port

19.5.2.2 WD Timer Clock and Reset Management

The Public Watchdog Timer functional clock (pi_sys_clk input) is sourced from either the on-chip ~32768 Hz oscillator (CLK_RC32K) or the PER PLL generated 32.768 KHz clock (CLK_32KHZ) as selected using CLKSEL_WDT1_CLK[CLKSEL] in the PRCM.

Table 19-110. Public WD Timer Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
PI_OCP_CLK Interface clock	26 MHz	CLK_M_OSC	pd_wkup_l4_wkup_gclk from PRCM
PI_SYS_CLK Functional clock	32768 Hz	CLK_RC32K or CLK_32KHZ (PER_CLKOUTM2 / 5859.375)	pd_wkup_wdt1_gclk from PRCM

19.5.3 Functional Description

19.5.3.1 Power Management

There are two clock domains in the watchdog timers:

- Functional clock domain: WDTi_FCLK is a 32 kHz watchdog timer functional clock. It is used to clock the watchdog timer internal logic.
- Interface clock domain: WDTi_ICLK is a 125 MHz watchdog timer interface clock. It is used to synchronize the watchdog timer L4 port to the L4 interconnect. All accesses from the interconnect are synchronous to WDTi_ICLK.

In this device, the clocks to the watchdog timers are always On. The clocks cannot be turned off if the watchdog timers are not being used.

19.5.3.2 Interrupts

Table 19-111 list the event flags, and their masks, that cause module interrupts.

Table 19-111. Watchdog Timer Events

Event Flag	Event Mask	Mapping	Comments
WDT_WIRQSTAT[0] EVENT_OVF	WDT_WIRQENSET/WDT_WIRQENCLR[0] OVF_IT_ENA	WDTINT	Watchdog timer overflow
WDT_WIRQSTAT[1] EVENT_DLY	WDT_WIRQENSET/WDT_WIRQENCLR[1] DLY_IT_ENA	WDTINT	Watchdog delay value reached

19.5.3.3 General Watchdog Timer Operation

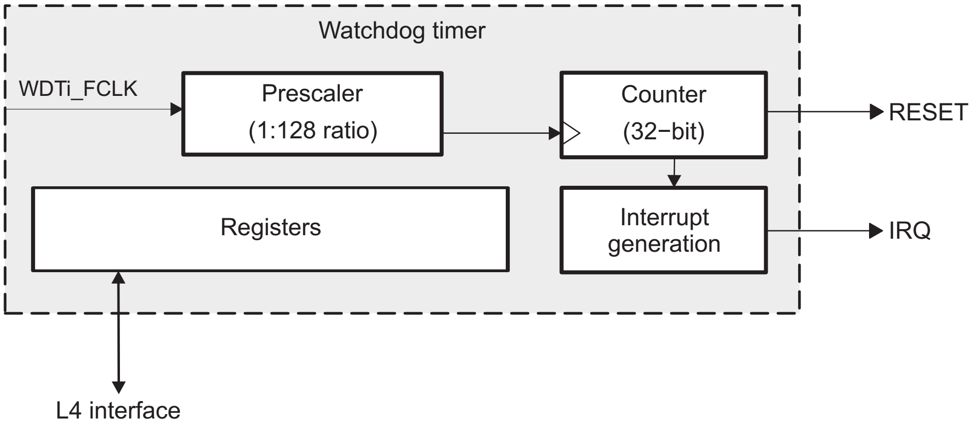
The watchdog timers are based on an upward 32-bit counter coupled with a prescaler. The counter overflow is signaled through two independent signals: a simple reset signal and an interrupt signal, both active low. Figure 19-105 is a functional block diagram of the watchdog timer.

The interrupt generation mechanism is controlled through the WDT_WIRQENSET/WDT_WIRQENCLR and WDT_WIRQSTAT registers.

The prescaler ratio can be set from 1 to 128 by accessing the WDT_WCLR[4:2] PTV bit field and the WDT_WCLR[5] PRE bit of the watchdog control register (WDT_WCLR).

The current timer value can be accessed on-the-fly by reading the watchdog timer counter register (WDT_WCRR), modified by accessing the watchdog timer load register (WDT_WLDR) (no on-the-fly update), or reloaded by following a specific reload sequence on the watchdog timer trigger register (WDT_WTGR). A start/stop sequence applied to the watchdog timer start/stop register (WDT_WSPR) can start and stop the watchdog timers.

Figure 19-105. 32-Bit Watchdog Timer Functional Block Diagram



19.5.3.4 Reset Context

The watchdog timers are enabled after reset. Table 19-112 lists the default reset values of the two watchdog timer load registers (the WDT_WLDR) and prescaler ratios (the WDT_WCLR[4:2] PTV bit field). To get these values, software must read the corresponding WDT_WCLR[4:2] PTV bit field and the 32-bit register to retrieve the static configuration of the module.

Table 19-112. Count and Prescaler Default Reset Values

Timer	WDT_WLDR Reset Value	PTV Reset Value
WDT	FFFF FFBEh	0

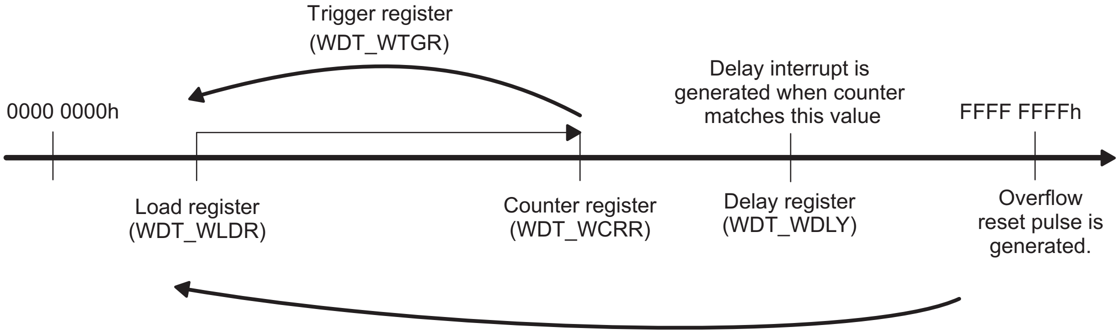
19.5.3.5 Overflow/Reset Generation

When the watchdog timer counter register (WDT_WCRR) overflows, an active-low reset pulse is generated to the PRCM module. This RESET pulse causes the PRCM module to generate global WARM reset of the device, which causes the nRESETIN_OUT pin to be driven out of the device. This pulse is one prescaled timer clock cycle wide and occurs at the same time as the timer counter overflow.

After reset generation, the counter is automatically reloaded with the value stored in the watchdog load register (WDT_WLDR) and the prescaler is reset (the prescaler ratio remains unchanged). When the reset pulse output is generated, the timer counter begins incrementing again.

Figure 19-106 shows a general functional view of the watchdog timers.

Figure 19-106. Watchdog Timers General Functional View



19.5.3.6 Prescaler Value/Timer Reset Frequency

Each watchdog timer is composed of a prescaler stage and a timer counter.

The timer rate is defined by the following values:

- Value of the prescaler fields (the WDT_WCLR[5] PRE bit and the WDT_WCLR[4:2] PTV bit field)
- Value loaded into the timer load register (WDT_WLDR)

The prescaler stage is clocked with the timer clock and acts as a clock divider for the timer counter stage. The ratio is managed by accessing the ratio definition field (the WDT_WCLR[4:2] PTV bit field) and is enabled with the WDT_WCLR[5] PRE bit.

Table 19-113 lists the prescaler clock ratio values.

Table 19-113. Prescaler Clock Ratio Values

WDT_WCLR[5] PRE	WDT_WCLR[4:2] PTV	Clock Divider (PS)
0	X	1
1	0	1
1	1	2
1	2	4
1	3	8
1	4	16
1	5	32
1	6	64
1	7	128

Thus the watchdog timer overflow rate is expressed as:

OVF_Rate = (FFFF FFFFh – WDT_WLDR + 1) × (wd-functional clock period) × PS

where wd-functional clock period = 1/(wd-functional clock frequency) and PS = $2^{(PTV)}$

CAUTION

Internal resynchronization causes some latency in any software write to WDT_WSPR before WDT_WSPR is updated with the programmed value:

1.5 × functional clock cycles ≤ write_WDT_WSPR_latency ≤ 2.5 × functional clock cycles

Remember to consider this latency whenever the watchdog timer must be started or stopped.

For example, for a timer clock input of 32 kHz with a prescaler ratio value of 1 (clock divided by 2) and WDT_WCLR[5] PRE = 1 (clock divider enabled), the reset period is as listed in Table 19-114.

Table 19-114. Reset Period Examples

WDT_WLDR Value	Reset Period
0000 0000h	74 h 34 min
FFFF 0000h	4 s
FFFF FFF0h	1 ms
FFFF FFFFh	62.5 µs

CAUTION

- Ensure that the reloaded value allows the correct operation of the application. When a watchdog timer is enabled, software must periodically trigger a reload before the counter overflows. Hence, the value of the WDT_WLDR[31:0] bit field must be chosen according to the ongoing activity preceding the watchdog reload.
- Due to design reasons, WDT_WLDR[31:0] = FFFF FFFFh is a special case, although such a value of WDT_WLDR is meaningless. When WDT_WLDR is programmed with the overflow value, a triggering event generates a reset/interrupt one functional clock cycle later, even if the watchdog timer is stopped.

Table 19-115 lists the default reset periods for the watchdog timers.

Table 19-115. Default Watchdog Timer Reset Periods

Watchdog Timers	Clock Source	Default Reset Period
WDT	32 kHz	2 s

19.5.3.7 Triggering a Timer Reload

To reload the timer counter and reset the prescaler before reaching overflow, a reload command is executed by accessing the watchdog timer trigger register (WDT_WTGR) using a specific reload sequence.

The specific reload sequence is performed whenever the written value on the WDT_WTGR register differs from its previous value. In this case, reload is executed in the same way as an overflow autoreload, but without the generation of a reset pulse.

The timer counter is loaded with the value of the watchdog timer load register (the WDT_WLDR[31:0] TIMER_LOAD bit field), and the prescaler is reset.

19.5.3.8 Start/Stop Sequence for Watchdog Timers (Using the WDT_WSPR Register)

To start and stop a watchdog timer, access must be made through the start/stop register (WDT_WSPR) using a specific sequence.

To disable the timer, follow this sequence:

1. Write XXXX AAAAh in WDT_WSPR.
2. Poll for posted write to complete using WDT_WWPS.W_PEND_WSPR.
3. Write XXXX 5555h in WDT_WSPR.
4. Poll for posted write to complete using WDT_WWPS.W_PEND_WSPR.

To enable the timer, follow this sequence:

1. Write XXXX BBBBh in WDT_WSPR.
2. Poll for posted write to complete using WDT_WWPS.W_PEND_WSPR.
3. Write XXXX 4444h in WDT_WSPR.
4. Poll for posted write to complete using WDT_WWPS.W_PEND_WSPR.

All other write sequences on the WDT_WSPR register have no effect on the start/stop feature of the module.

19.5.3.9 Modifying Timer Count/Load Values and Prescaler Setting

To modify the timer counter value (the WDT_WCRR register), prescaler ratio (the WDT_WCLR[4:2] PTV bit field), delay configuration value (the WDT_WDLY[31:0] DLY_VALUE bit field), or the load value (the WDT_WLDR[31:0] TIMER_LOAD bit field), the watchdog timer must be disabled by using the start/stop sequence (the WDT_WSPR register).

After a write access, the load register value and prescaler ratio registers are updated immediately, but new values are considered only after the next consecutive counter overflow or after a new trigger command (the WDT_WTGR register).

19.5.3.10 Watchdog Counter Register Access Restriction (WDT_WCRR Register)

A 32-bit shadow register is implemented to read a coherent value of the WDT_WCRR register because the WDT_WCRR register is directly related to the timer counter value and is updated on the timer clock (WDT_FCLK). The shadow register is updated by a 16-bit LSB read command.

NOTE: Although the L4 clock (WDT_ICLK) is completely asynchronous with the timer clock (WDT_FCLK), some synchronization is performed to ensure that the value of the WDT_WCRR register is not read while it is being incremented.

When 32-bit read access is performed, the shadow register is not updated. Read access is performed directly from the accessed register.

To ensure that a coherent value is read inside WDT_WCRR, the first read access is to the lower 16 bits (offset = 8h), followed by read access to the upper 16 bits (offset = Ah).

19.5.3.11 Watchdog Timer Interrupt Generation

When an interrupt source occurs, the interrupt status bit (the WDT_WIRQSTAT[0] EVENT_OVF or WDT_WIRQSTAT[1] EVENT_DLY bit) is set to 1. The output interrupt line (WDTi_IRQ) is asserted (active low) when status (the EVENT_xxx bit) and enable (the xxx_IT_ENA bit) flags are set to 1; the order is not relevant. Writing 1 to the enable bit (the status is already set at 1) also triggers the interrupt in the normal order (enable first, status next). The pending interrupt event is cleared when the set status bit is overwritten by a value of 1 by a write command in the WDT_WIRQSTAT register. Reading the WDT_WIRQSTAT register and writing the value back allows a fast interrupt acknowledge process.

The watchdog timer issues an overflow interrupt if this interrupt is enabled in the watchdog interrupt enable register (WDT_WIRQENSET[0] OVF_IT_ENA = 1). When the overflow occurs, the interrupt status bit (the WDT_WIRQSTAT[0] EVENT_OVF bit) is set to 1. The output interrupt line (WDT_IRQ) is asserted (active low) when status (EVENT_OVF) and enable (OVF_IT_ENA) flags are set to 1; the order is not relevant. This interrupt can be disabled by setting the WDT_WIRQENCLR[0] OVF_IT_ENA bit to 1.

The watchdog can issue the delay interrupt if this interrupt is enabled in the interrupt enable register (WDT_WIRQENSET[1] DLY_IT_ENA = 1). When the counter is running and the counter value matches the value stored in the delay configuration register (WDT_WDLY), the corresponding interrupt status bit is set in the watchdog status register (WDT_WIRQSTAT) and the output interrupt line is asserted (active low) when the flag (EVENT_DLY) and enable (DLY_IT_ENA) bits are 1 in the WDT_WIRQSTAT and WDT_WIRQENSET registers, respectively; the order (normally enable, then flag), is not relevant. This interrupt can be disabled by setting the WDT_WIRQENCLR[1] DLY_IT_ENA bit to 1.

NOTE: Writing 0 to the WDT_WIRQSTAT[0] EVENT_OVF bit or the WDT_WIRQSTAT[1] EVENT_DLY bit has no effect.

The two clock domains are resynchronized because the interrupt event is generated on the functional clock domain (WDTi_FCLK) during the updating of the interrupt status register (WDT_WIRQSTAT).

The WDT_WDLY register is used to specify the value of the delay configuration register. The delay time to interrupt is the difference between the reload value stored in the counter load register (WDT_WLDR) and the programmed value in this register (WDT_WDLY).

Use the following formula to estimate the delay time:

Delay time period = (WDT_WDLY – WDT_WLDR + 1) × Timer clock period × Clock divider

Where:

- Timer clock period = 1/(Timer clock frequency)
- Clock divider = 2**PTV

If the counter value (WDT_WCRR) reaches the programmed value (WDT_WDLY), the status bit (EVENT_DLY) gets set in the interrupt status register (WDT_WIRQSTAT), and an interrupt occurs if the corresponding enable bit is set in the interrupt enable register (WDT_WIRQENSET).

CAUTION

If the reload event occurs (after a triggering sequence or after a reset sequence) before reaching the programmed value (WDT_WDLY[31:0] WDLY_VALUE), no interrupt is generated.

Also, no interrupt is generated if the value programmed in the delay configuration register (WDT_WDLY) is less than the value stored in the counter load register (WDT_WLDR).

19.5.3.12 Watchdog Timers Under Emulation

To configure the WDT to stop during emulation suspend events (for example, debugger breakpoints), set up the WDT and the Debug Subsystem:

1. Set WDT_WDSC.EMUFREE=0. This will allow the Suspend_Control signal from the Debug Subsystem (Chapter 31) to stop and start the WDT. Note that if EMUFREE=1, the Suspend_Control signal is ignored and the WDT is free running regardless of any debug suspend event. This EMUFREE bit gives local control from a module perspective to gate the suspend signal coming from the Debug Subsystem.
2. Set the appropriate xxx_Suspend_Control register = 0x9, as described in Section 31.1.1.1, *Debug Suspend Support for Peripherals*. Choose the register appropriate to the peripheral you want to suspend during a suspend event.

19.5.3.13 Accessing Watchdog Timer Registers

Posted/nonposted selection applies only to functional registers that require synchronization on/from the timer functional clock domain (WDTi_FCLK). For write/read operation, the following registers are affected:

- WDT_WCLR
- WDT_WCRR
- WDT_WLDR
- WDT_WTGR
- WDT_WDLY
- WDT_WSPR

The timer interface clock domain synchronous registers are not affected by the posted/nonposted selection; the write/read operation is effective and acknowledged (command accepted) after one WDT_ICLK cycle from the command assertion. The timer interface clock domain synchronous registers are:

- WDT_WIDR
- WDT_WDSC
- WDT_WDST
- WDT_WIRQSTATRAW
- WDT_WIRQSTAT
- WDT_WIRQENSET
- WDT_WIRQENCLR
- WDT_WWPS

19.5.3.14 Low-Level Programming Model

This section covers the low-level hardware programming sequences for configuration and use of the module.

19.5.3.14.1 Global Initialization

19.5.3.14.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the watchdog timer is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the watchdog timer (see Table 19-116).

Table 19-116. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	The module interface and functional clocks must be enabled.
Control module	Module-specific pad multiplexing must be set in the control module.
MPU INTC	The MPU INTC configuration must be performed to enable the interrupts from the watchdog timer.

19.5.3.14.1.2 Main Sequence – Watchdog Timer Module Global Initialization

Table 19-117 lists the steps for initializing the watchdog timer module when the module is to be used for the first time.

Table 19-117. Watchdog Timer Module Global Initialization

Step	Register/Bit Field/Programming Model	Value
Execute software reset.	WDT_WDSC[1] SOFTRESET	1
Wait until reset release?	WDT_WDSC[1] SOFTRESET	0
Enable delay interrupt.	WDT_WIRQENSET[1] ENABLE_DLY	1
Enable overflow interrupt.	WDT_WIRQENSET[0] ENABLE_OVF	1

19.5.3.14.2 Operational Mode Configuration

19.5.3.14.2.1 Main Sequence – Watchdog Timer Basic Configuration

Table 19-118 lists the steps for the basic configuration of the watchdog timer.

Table 19-118. Watchdog Timer Basic Configuration

Step	Register/Bit Field/Programming Model	Value
Disable the watchdog timer.	See Section 19.5.3.14.2.2.	
Set prescaler value.	WDT_WCLR[4:2] PTV	xxx
Enable prescaler.	WDT_WCLR[5] PRE	1
Load delay configuration value.	WDT_WDLY	xxx
Load timer counter value.	WDT_WCRR	xxx
Enable the watchdog timer.	See Section 19.5.3.14.2.3.	

19.5.3.14.2.2 Subsequence – Disable the Watchdog Timer

Table 19-119 lists the steps to disable the watchdog timer.

Table 19-119. Disable the Watchdog Timer

Step	Register/Bit Field/Programming Model	Value
Write disable sequence Data1.	WDT_WSPR	XXXX AAAAh
Write disable sequence Data2.	WDT_WSPR	XXXX 5555h

19.5.3.14.2.3 Subsequence – Enable the Watchdog Timer

Table 19-120 lists the steps to enable the watchdog timer.

Table 19-120. Enable the Watchdog Timer

Step	Register/Bit Field/Programming Model	Value
Write enable sequence Data1.	WDT_WSPR	XXXX BBBBh
Write enable sequence Data2.	WDT_WSPR	XXXX 4444h

19.5.4 WDT Registers

Table 19-121 lists the memory-mapped registers for the WDT. All register offset addresses not listed in Table 19-121 should be considered as reserved locations and the register contents should not be modified.

Table 19-121. WDT Registers

Offset	Acronym	Register Name	Section
0h	WDT_WIDR	Watchdog Identification Register	Section 19.5.4.1
10h	WDT_WDSC	Watchdog System Control Register	Section 19.5.4.2
14h	WDT_WDST	Watchdog Status Register	Section 19.5.4.3
18h	WDT_WISR	Watchdog Interrupt Status Register	Section 19.5.4.4
1Ch	WDT_WIER	Watchdog Interrupt Enable Register	Section 19.5.4.5
24h	WDT_WCLR	Watchdog Control Register	Section 19.5.4.6
28h	WDT_WCRR	Watchdog Counter Register	Section 19.5.4.7
2Ch	WDT_WLDR	Watchdog Load Register	Section 19.5.4.8
30h	WDT_WTGR	Watchdog Trigger Register	Section 19.5.4.9
34h	WDT_WWPS	Watchdog Write Posting Bits Register	Section 19.5.4.10
44h	WDT_WDLY	Watchdog Delay Configuration Register	Section 19.5.4.11
48h	WDT_WSPR	Watchdog Start/Stop Register	Section 19.5.4.12
54h	WDT_WIRQSTATRAW	Watchdog Raw Interrupt Status Register	Section 19.5.4.13
58h	WDT_WIRQSTAT	Watchdog Interrupt Status Register	Section 19.5.4.14
5Ch	WDT_WIRQENSET	Watchdog Interrupt Enable Set Register	Section 19.5.4.15
60h	WDT_WIRQENCLR	Watchdog Interrupt Enable Clear Register	Section 19.5.4.16

19.5.4.1 WDT_WIDR Register (offset = 0h) [reset = 0h]

WDT_WIDR is shown in Figure 19-107 and described in Table 19-122.

Watchdog Identification Register

Figure 19-107. WDT_WIDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															
R-0h																															

Table 19-122. WDT_WIDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REVISION	R	0h	IP Revision

19.5.4.2 WDT_WDSC Register (offset = 10h) [reset = 10h]

WDT_WDSC is shown in Figure 19-108 and described in Table 19-123.

The Watchdog System Control Register controls the various parameters of the L4 interface.

Figure 19-108. WDT_WDSC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED		EMUFREE	IDLEMODE		RESERVED	SOFTRESET	RESERVED
R-0h		R/W-0h	R/W-2h		R-0h	R/W-0h	R-0h

Table 19-123. WDT_WDSC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5	EMUFREE	R/W	0h	Sensitivity to emulation (debug) suspend event from Debug Subsystem. 0h (R/W) = Timer counter frozen during debug suspend event. 1h (R/W) = Timer counter free-running. Debug suspend event is ignored.
4-3	IDLEMODE	R/W	2h	Configuration of the local target state management mode.By definition, target can handle read/write transaction as long as it is out of IDLE state. 0h (R/W) = Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements. Backup mode, for debug only. 1h (R/W) = No-idle mode: local target never enters idle state.Backup mode, for debug only. 2h (R/W) = Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wakeup events. 3h (R/W) = Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state.
2	RESERVED	R	0h	
1	SOFTRESET	R/W	0h	Software reset. (Optional) 0h (W) = No action 0h (R) = Reset done, no pending action 1h (R) = Reset (software or other) ongoing 1h (W) = Initiate software reset
0	RESERVED	R	0h	

19.5.4.3 WDT_WDST Register (offset = 14h) [reset = 1h]

WDT_WDST is shown in Figure 19-109 and described in Table 19-124.

The Watchdog Status Register provides status information about the module.

Figure 19-109. WDT_WDST Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							RESETDONE
R-0h							R-1h

Table 19-124. WDT_WDST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	RESETDONE	R	1h	Internal module reset monitoring 0h (R) = Internal module reset is ongoing. 1h (R) = Reset completed

19.5.4.4 WDT_WISR Register (offset = 18h) [reset = 0h]

WDT_WISR is shown in Figure 19-110 and described in Table 19-125.

The Watchdog Interrupt Status Register shows which interrupt events are pending inside the module.

Figure 19-110. WDT_WISR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						DLY_IT_FLAG	OVF_IT_FLAG
R-0h						R/W-0h	R/W-0h

Table 19-125. WDT_WISR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	DLY_IT_FLAG	R/W	0h	Pending delay interrupt status. 0h (W) = Status unchanged 0h (R) = No delay interrupt pending 1h (W) = Status bit cleared 1h (R) = Delay interrupt pending
0	OVF_IT_FLAG	R/W	0h	Pending overflow interrupt status. 0h (W) = Status unchanged 0h (R) = No overflow interrupt pending 1h (W) = Status bit cleared 1h (R) = Overflow interrupt pending

19.5.4.5 WDT_WIER Register (offset = 1Ch) [reset = 0h]

WDT_WIER is shown in Figure 19-111 and described in Table 19-126.

The Watchdog Interrupt Enable Register controls (enable/disable) the interrupt events.

Figure 19-111. WDT_WIER Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						DLY_IT_ENA	OVF_IT_ENA
R-0h						R/W-0h	R/W-0h

Table 19-126. WDT_WIER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	DLY_IT_ENA	R/W	0h	Delay interrupt enable/disable 0h (R/W) = Disable delay interrupt. 1h (R/W) = Enable delay interrupt.
0	OVF_IT_ENA	R/W	0h	Overflow interrupt enable/disable 0h (R/W) = Disable overflow interrupt. 1h (R/W) = Enable overflow interrupt.

19.5.4.6 WDT_WCLR Register (offset = 24h) [reset = 20h]

WDT_WCLR is shown in Figure 19-112 and described in Table 19-127.

The Watchdog Control Register controls the prescaler stage of the counter.

Figure 19-112. WDT_WCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED		PRE	PTV			RESERVED	
R-0h		R/W-1h	R/W-0h			R-0h	

Table 19-127. WDT_WCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5	PRE	R/W	1h	Prescaler enable/disable configuration 0h (R/W) = Prescaler disabled 1h (R/W) = Prescaler enabled
4-2	PTV	R/W	0h	Prescaler value. The timer counter is prescaled with the value: 2**PTV. Example: PTV = 3 then counter increases value if started after 8 functional clock periods. On reset, it is loaded from PI_PTV_RESET_VALUE input port.
1-0	RESERVED	R	0h	

19.5.4.7 WDT_WCRR Register (offset = 28h) [reset = 0h]

WDT_WCRR is shown in Figure 19-113 and described in Table 19-128.

The Watchdog Counter Register holds the value of the internal counter.

Figure 19-113. WDT_WCRR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER_CTR																															
R/W-0h																															

Table 19-128. WDT_WCRR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TIMER_CTR	R/W	0h	Value of the timer counter register

19.5.4.8 WDT_WLDR Register (offset = 2Ch) [reset = 0h]

WDT_WLDR is shown in Figure 19-114 and described in Table 19-129.

The Watchdog Load Register holds the timer load value.

Figure 19-114. WDT_WLDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER_LOAD																															
R/W-0h																															

Table 19-129. WDT_WLDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TIMER_LOAD	R/W	0h	Value of the timer load register

19.5.4.9 WDT_WTGR Register (offset = 30h) [reset = 0h]

WDT_WTGR is shown in Figure 19-115 and described in Table 19-130.

Writing a different value than the one already written in the Watchdog Trigger Register does a watchdog counter reload.

Figure 19-115. WDT_WTGR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">TTGR_VALUE</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 19-130. WDT_WTGR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TTGR_VALUE	R/W	0h	Value of the trigger register

19.5.4.10 WDT_WWPS Register (offset = 34h) [reset = 0h]

WDT_WWPS is shown in Figure 19-116 and described in Table 19-131.

The Watchdog Write Posting Bits Register contains the write posting bits for all writeable functional registers.

Figure 19-116. WDT_WWPS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		W_PEND_WDLY	W_PEND_WSPR	W_PEND_WTGR	W_PEND_WLDR	W_PEND_WCRR	W_PEND_WCLR
R-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 19-131. WDT_WWPS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5	W_PEND_WDLY	R	0h	Write pending for register WDLY 0h (R) = No register write pending 1h (R) = Register write pending
4	W_PEND_WSPR	R	0h	Write pending for register WSPR 0h (R) = No register write pending 1h (R) = Register write pending
3	W_PEND_WTGR	R	0h	Write pending for register WTGR 0h (R) = No register write pending 1h (R) = Register write pending
2	W_PEND_WLDR	R	0h	Write pending for register WLDR 0h (R) = No register write pending 1h (R) = Register write pending
1	W_PEND_WCRR	R	0h	Write pending for register WCRR 0h (R) = No register write pending 1h (R) = Register write pending
0	W_PEND_WCLR	R	0h	Write pending for register WCLR 0h (R) = No register write pending 1h (R) = Register write pending

19.5.4.11 WDT_WDLY Register (offset = 44h) [reset = 0h]

WDT_WDLY is shown in Figure 19-117 and described in Table 19-132.

The Watchdog Delay Configuration Register holds the delay value that controls the internal pre-overflow event detection.

Figure 19-117. WDT_WDLY Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDLY_VALUE																															
R/W-0h																															

Table 19-132. WDT_WDLY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	WDLY_VALUE	R/W	0h	Value of the delay register

19.5.4.12 WDT_WSPR Register (offset = 48h) [reset = 0h]

WDT_WSPR is shown in Figure 19-118 and described in Table 19-133.

The Watchdog Start/Stop Register holds the start-stop value that controls the internal start-stop FSM.

Figure 19-118. WDT_WSPR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WSPR_VALUE																															
R/W-0h																															

Table 19-133. WDT_WSPR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	WSPR_VALUE	R/W	0h	Value of the start-stop register

19.5.4.13 WDT_WIRQSTATRAW Register (offset = 54h) [reset = 0h]

WDT_WIRQSTATRAW is shown in Figure 19-119 and described in Table 19-134.

In the Watchdog Raw Interrupt Status Register, IRQ unmasked status, status set per-event raw interrupt status vector, line 0. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.

Figure 19-119. WDT_WIRQSTATRAW Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>EVT_DLY</td><td>EVT_OVF</td></tr>
<tr><td colspan="6">R-0h</td><td>R/W1toSet-0h</td><td>R/W1toSet-0h</td></tr>
</table>

Table 19-134. WDT_WIRQSTATRAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	EVT_DLY	R/W1toSet	0h	Settable raw status for delay event 0h (W) = No action 0h (R) = No event pending 1h (R) = Event pending 1h (W) = Set event (debug)
0	EVT_OVF	R/W1toSet	0h	Settable raw status for overflow event 0h (W) = No action 0h (R) = No event pending 1h (R) = Event pending 1h (W) = Set event (debug)

19.5.4.14 WDT_WIRQSTAT Register (offset = 58h) [reset = 0h]

WDT_WIRQSTAT is shown in Figure 19-120 and described in Table 19-135.

In the Watchdog Interrupt Status Register, IRQ masked status, status clear per-event enabled interrupt status vector, line 0. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).

Figure 19-120. WDT_WIRQSTAT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>EVT_DLY</td><td>EVT_OVF</td></tr>
<tr><td colspan="6">R-0h</td><td>R/W1toClr-0h</td><td>R/W1toClr-0h</td></tr>
</table>

Table 19-135. WDT_WIRQSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	EVT_DLY	R/W1toClr	0h	Clearable, enabled status for delay event 0h (W) = No action 0h (R) = No (enabled) event pending 1h (R) = Event pending 1h (W) = Clear (raw) event
0	EVT_OVF	R/W1toClr	0h	Clearable, enabled status for overflow event 0h (W) = No action 0h (R) = No (enabled) event pending 1h (R) = Event pending 1h (W) = Clear (raw) event

19.5.4.15 WDT_WIRQENSET Register (offset = 5Ch) [reset = 0h]

WDT_WIRQENSET is shown in Figure 19-121 and described in Table 19-136.

In the Watchdog Interrupt Enable Set Register, IRQ enable set per-event interrupt enable bit vector, line 0. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.

Figure 19-121. WDT_WIRQENSET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED						EN_DLY	EN_OVF
R-0h						R/W1toSet-0h	R/W1toSet-0h

Table 19-136. WDT_WIRQENSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	EN_DLY	R/W1toSet	0h	Enable for delay event 0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (R) = Interrupt enabled 1h (W) = Enable interrupt
0	EN_OVF	R/W1toSet	0h	Enable for overflow event 0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (R) = Interrupt enabled 1h (W) = Enable interrupt

19.5.4.16 WDT_WIRQENCLR Register (offset = 60h) [reset = 0h]

WDT_WIRQENCLR is shown in Figure 19-122 and described in Table 19-137.

In the Watchdog Interrupt Enable Clear Register, IRQ enable clear per-event interrupt enable bit vector, line 0. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.

Figure 19-122. WDT_WIRQENCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						EN_DLY	EN_OVF
R-0h						R/W1toClr-0h	R/W1toClr-0h

Table 19-137. WDT_WIRQENCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	EN_DLY	R/W1toClr	0h	Enable for delay event 0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Disable interrupt 1h (R) = Interrupt enabled
0	EN_OVF	R/W1toClr	0h	Enable for overflow event 0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Disable interrupt 1h (R) = Interrupt enabled

Pulse-Width Modulation Subsystem (PWMSS)

This chapter describes the PWMSS of the device.

Topic **Page**

20.1 Pulse-Width Modulation Subsystem (PWMSS)

20.1.1 Introduction

20.1.1.1 Features

The general features of the PWMSS are:

eHRPWM

- Dedicated 16 bit time-base with Period / Frequency control
- Can support 2 independent PWM outputs with Single edge operation
- Can support 2 independent PWM outputs with Dual edge symmetric operation
- Can support 1 independent PWM output with Dual edge asymmetric operation
- Supports Dead-band generation with independent Rising and Falling edge delay control
- Provides asynchronous over-ride control of PWM signals during fault conditions
- Supports “trip zone” allocation of both latched and un-latched fault conditions
- Allows events to trigger both CPU interrupts and start of ADC conversions
- Support PWM chopping by high frequency carrier signal, used for pulse transformer gate drives.
- High-resolution module with programmable delay line.
  - Programmable on a per PWM period basis.
  - Can be inserted either on the rising edge or falling edge of the PWM pulse or both or not at all.

eCAP

- Dedicated input Capture pin
- 32 bit Time Base (counter)
- 4 x 32 bit Time-stamp Capture registers (CAP1-CAP4)
- 4 stage sequencer (Mod4 counter) which is synchronized to external events (ECAPx pin edges)
- Independent Edge polarity (Rising / Falling edge) selection for all 4 events
- Input Capture signal pre-scaling (from 1 to 16)
- One-shot compare register (2 bits) to freeze captures after 1 to 4 Time-stamp events
- Control for continuous Time-stamp captures using a 4 deep circular buffer (CAP1-CAP4) scheme
- Interrupt capabilities on any of the 4 capture events

eQEP

- Input Synchronization
- Quadrature Decoder Unit
- Position Counter and Control unit for position measurement
- Quadrature Edge Capture unit for low speed measurement
- Unit Time base for speed/frequency measurement
- Watchdog Timer for detecting stalls

20.1.1.2 Unsupported Features

The PWMSS module features not supported are shown in Table 20-1.

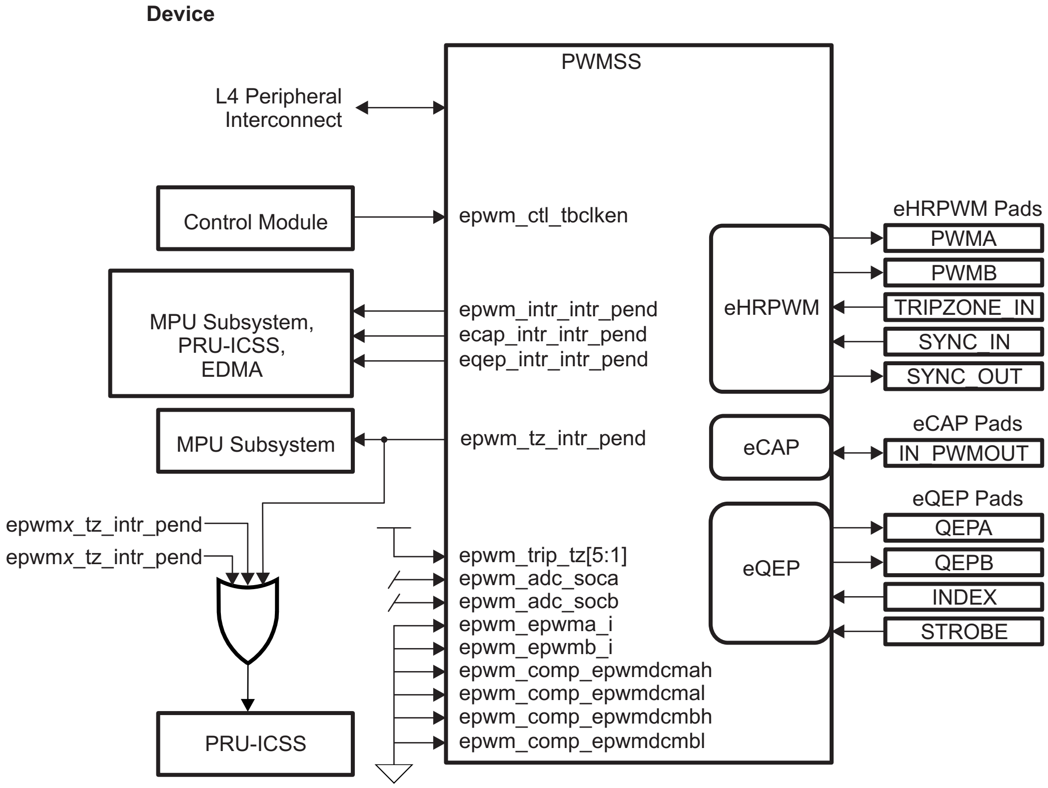
Table 20-1. Unsupported Features

Feature	Reason
ePWM inputs	Not pinned out
ePWM tripzone 1-5 inputs	Only Tripzone0 is pinned out
ePWM digital comparators	Inputs not connected
eQEP quadrature outputs	Only input signals are connected
eCAP3–5	Module not used
eQEP3–5	Module not used

20.1.2 Integration

The Pulse Width Modulation Subsystem (PWMSS) includes a single instance of the Enhanced High Resolution Pulse Width Modulator (eHRPWM), Enhanced Capture (eCAP), and Enhanced Quadrature Encoded Pulse (eQEP) modules. This device includes six instantiations of the PWMSS.

Figure 20-1. PWMSS Integration



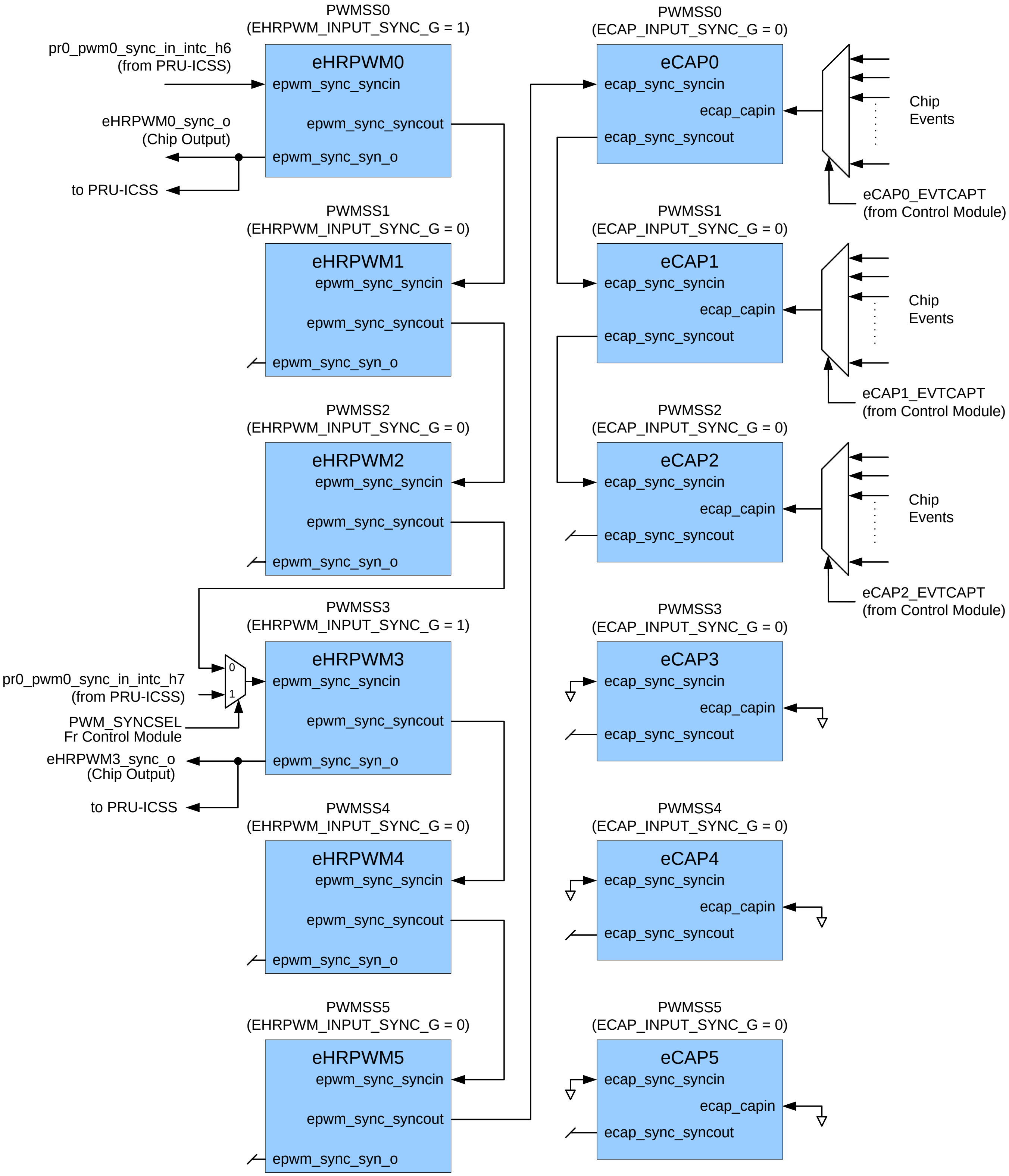
20.1.2.1 PWMSS Synchronization Detail

The PWM (eHRPWM) and capture (eCAP) components of the PWMSS provide synchronization signals to allow them to be synchronized to other modules or events. On this device, these signals are connected in a daisy-chain scheme for PWMSS0–5, as shown in Figure 20-2.

The eCAP capture events may be selected from among 31 different pins or internal interrupt signals, as detailed in Section 8.2. The event is selected using the corresponding ECAPx_EVTCAPT field of the ECAP_EVT_CAP register in the Control Module.

For PWMSS3–5, the eHRPWM modules can be synchronized to a separate external event routed through the PRU-ICSS.

Figure 20-2. PWMSS Synchronization



20.1.2.2 PWMSS Connectivity Attributes

The general connectivity attributes for the PWMSS module are shown in Table 20-2.

The tripzone interrupts from the three PWMSS instantiations (0-2, 3-5) are ORed together to form a single interrupt to the PRU-ICSS but are routed individually to MPU Subsystem, as shown in Figure 20-1.

Table 20-2. PWMSS Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L4LS_GCLK
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	2 ePWM interrupts per instance epwm_intr_intr - Event interrupt, ePWMxINT for ARM subsystem, epwm_intr_intr_pend for PRU-ICSS epwm_tz_intr - Tripzone interrupt, ePWMx_TZINT for ARM subsystem, pwm_trip_zone for PRU-ICSS (OR'd together as two interrupts, ePWM(0-2) and ePWM(3-5)) 1 eCAP interrupt per instance ecap_intr - Capture/PWM event interrupt, eCAPxINT for ARM subsystem, ecap_intr_intr_pend for PRU-ICSS 1 eQEP Interrupt per instance eqep_intr_intr - Event interrupt, eQEPxINT for ARM subsystem, eqep_intr_intr_pend for PRU-ICSS (only for eQEP0)
DMA Requests	Interrupt requests are redirected as DMA requests: • 1 DMA request from ePWM per instance (ePWMEVTx) • 1 DMA request from eCAP per instance (eCAPEVTx) • 1 DMA request from eQEP per instance (eQEPEVTx)
Physical Address	L4 Peripheral slave port

20.1.2.3 PWMSS Clock and Reset Management

The PWMSS controllers have separate bus interface and functional clocks.

Table 20-3. PWMSS Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
PWMSS_ocp_clk Interface / Functional clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk from PRCM

20.1.2.4 PWMSS Pin list

The external signals for the PWMSS module are shown in the following table.

Table 20-4. PWMSS Pin List

Pin	Type*	Description
EPWMxA	O	PWM output A
EPWMxB	O	PWM output B
EPWM_SYNCIN	I	PWM Sync input
EPWM_SYNCOUT	O	PWM Sync output
EPWM_TRIPZONE[5:0]	I	PWM Tripzone inputs
ECAP_CAPIN_APWMOUT	I/O	eCAP Capture input / PWM output
EQEP_A	I/O	eQEP Quadrature input/output
EQEP_B	I/O	eQEP Quadrature input/output
EQEP_INDEX	I/O	eQEP Index input/output
EQEP_STROBE	I/O	eQEP Strobe input/output

20.1.3 PWMSS Registers

Table 20-5 lists the memory-mapped registers for the PWMSS. All register offset addresses not listed in Table 20-5 should be considered as reserved locations and the register contents should not be modified.

Table 20-5. PWMSS Registers

Offset	Acronym	Register Name	Section
0h	IDVER	IP Revision Register	Section 20.1.3.1
4h	SYSCONFIG	System Configuration Register	Section 20.1.3.2
8h	CLKCONFIG	Clock Configuration Register	Section 20.1.3.3
Ch	CLKSTATUS	Clock Status Register	Section 20.1.3.4

20.1.3.1 IDVER Register (Offset = 0h) [reset = 40000000h]

IDVER is shown in Figure 20-3 and described in Table 20-6.

Return to Summary Table.

The IP revision register is used by software to track features, bugs, and compatibility.

Figure 20-3. IDVER Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="2">SCHEME</td><td colspan="2">RESERVED</td><td colspan="4">FUNC</td></tr>
<tr><td colspan="2">R-1h</td><td colspan="2">R-0h</td><td colspan="4">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">FUNC</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="5">R_RTL</td><td colspan="3">X_MAJOR</td></tr>
<tr><td colspan="5">R-0h</td><td colspan="3">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CUSTOM</td><td colspan="6">Y_MINOR</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="6">R-0h</td></tr>
</table>

Table 20-6. IDVER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	Used to distinguish between the old scheme and current.
29-28	RESERVED	R	0h	
27-16	FUNC	R	0h	FUNC
15-11	R_RTL	R	0h	RTL version (R), maintained by IP design owner.
10-8	X_MAJOR	R	0h	Major revision (X)
7-6	CUSTOM	R	0h	CUSTOM
5-0	Y_MINOR	R	0h	Minor revision (Y)

20.1.3.2 SYSCONFIG Register (Offset = 4h) [reset = 28h]

SYSCONFIG is shown in Figure 20-4 and described in Table 20-7.

Return to Summary Table.

The system configuration register is used for clock management configuration.

Figure 20-4. SYSCONFIG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="2">STANDBYMODE</td><td colspan="2">IDLEMODE</td><td>FREEEMU</td><td>SOFTRESET</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="2">R/W-2h</td><td colspan="2">R/W-2h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 20-7. SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5-4	STANDBYMODE	R/W	2h	Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of STANDBY state. 0h (R/W) = Force-standby mode: local initiator is unconditionally placed in standby state. Backup mode, for debug only. 1h (R/W) = No-standby mode: local initiator is unconditionally placed out of standby state. Backup mode, for debug only. 2h (R/W) = Smart-standby mode: local initiator standby status depends on local conditions, i.e., the module's functional requirement from the initiator. IP module should not generate (initiator-related) wakeup events.
3-2	IDLEMODE	R/W	2h	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state. 0h (R/W) = Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements. Backup mode, for debug only. 1h (R/W) = No-idle mode: local target never enters idle state. Backup mode, for debug only. 2h (R/W) = Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wakeup events.
1	FREEEMU	R/W	0h	Sensitivity to emulation (debug) suspend event from Debug Subsystem. 0h (R/W) = IP module is sensitive to emulation suspend. 1h (R/W) = IP module is not sensitive to emulation suspend event. Debug suspend event is ignored.
0	SOFTRESET	R/W	0h	Software reset (optional)

20.1.3.3 CLKCONFIG Register (Offset = 8h) [reset = 111h]

CLKCONFIG is shown in Figure 20-5 and described in Table 20-8.

Return to Summary Table.

The clock configuration register is used in the PWMSS submodule for clkstop req and clk_en control. Note: PWMSS Modules Local Clock Gating feature is not supported. This register should not be modified. Clock gating functionality is controlled by PRCM.

Figure 20-5. CLKCONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED						ePWMCLKSTOP_REQ	ePWMCLK_EN
R-0h						R/W-0h	R/W-1h
7	6	5	4	3	2	1	0
RESERVED		eQEPCLKSTOP_REQ	eQEPCLK_EN	RESERVED		eCAPCLKSTOP_REQ	eCAPCLK_EN
R-0h		R/W-0h	R/W-1h	R-0h		R/W-0h	R/W-1h

Table 20-8. CLKCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9	ePWMCLKSTOP_REQ	R/W	0h	This bit controls the clkstop_req input to the ePWM module.
8	ePWMCLK_EN	R/W	1h	This bit controls the clk_en input to the ePWM module.
7-6	RESERVED	R	0h	
5	eQEPCLKSTOP_REQ	R/W	0h	This bit controls the clkstop_req input to the eQEP module
4	eQEPCLK_EN	R/W	1h	This bit controls the clk_en input to the eQEP module.
3-2	RESERVED	R	0h	
1	eCAPCLKSTOP_REQ	R/W	0h	This bit controls the clkstop_req input to the eCAP module.
0	eCAPCLK_EN	R/W	1h	This bit controls the clk_en input to the eCAP module.

20.1.3.4 CLKSTATUS Register (Offset = Ch) [reset = 0h]

CLKSTATUS is shown in Figure 20-6 and described in Table 20-9.

Return to Summary Table.

The clock status register is used in the PWMSS submodule for clkstop ack and clk_en ack status. Note: PWMSS Modules Local Clock Gating feature is not supported. This register should not be modified. Clock gating functionality is controlled by PRCM.

Figure 20-6. CLKSTATUS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED						ePWM_CLKSTOP_ACK	ePWM_CLK_EN_ACK
R-0h						R-0h	R-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED		eQEP_CLKSTOP_ACK	eQEP_CLK_EN_ACK	RESERVED		eCAP_CLKSTOP_ACK	eCAP_CLK_EN_ACK
R-0h		R-0h	R-0h	R-0h		R-0h	R-0h

Table 20-9. CLKSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9	ePWM_CLKSTOP_ACK	R	0h	This bit is the clkstop_req_ack status output of the ePWM module.
8	ePWM_CLK_EN_ACK	R	0h	This bit is the clk_en status output of the ePWM module.
7-6	RESERVED	R	0h	
5	eQEP_CLKSTOP_ACK	R	0h	This bit is the clkstop_req_ack status output of the eQEP module.
4	eQEP_CLK_EN_ACK	R	0h	This bit is the clk_en status output of the eQEP module.
3-2	RESERVED	R	0h	
1	eCAP_CLKSTOP_ACK	R	0h	This bit is the clkstop_req_ack status output of the eCAP module.
0	eCAP_CLK_EN_ACK	R	0h	This bit is the clk_en status output of the eCAP module.

20.2 Enhanced PWM (ePWM) Module

20.2.1 Introduction

An effective PWM peripheral must be able to generate complex pulse width waveforms with minimal CPU overhead or intervention. It needs to be highly programmable and very flexible while being easy to understand and use. The ePWM unit described here addresses these requirements by allocating all needed timing and control resources on a per PWM channel basis. Cross coupling or sharing of resources has been avoided; instead, the ePWM is built up from smaller single channel modules with separate resources and that can operate together as required to form a system. This modular approach results in an orthogonal architecture and provides a more transparent view of the peripheral structure, helping users to understand its operation quickly.

In this chapter, the letter x within a signal or module name is used to indicate a generic ePWM instance on a device. For example, output signals EPWMxA and EPWMxB refer to the output signals from the ePWMx instance. Thus, EPWM1A and EPWM1B belong to ePWM1 and, likewise, EPWM4A and EPWM4B belong to ePWM4.

20.2.1.1 Submodule Overview

The ePWM module represents one complete PWM channel composed of two PWM outputs: EPWMxA and EPWMxB. Multiple ePWM modules are instanced within a device as shown in Figure 20-7. Each ePWM instance is identical with one exception. Some instances include a hardware extension that allows more precise control of the PWM outputs. This extension is the high-resolution pulse width modulator (HRPWM) and is described in Section 20.2.2.10. See Section 20.1.2 to determine which ePWM instances include this feature. Each ePWM module is indicated by a numerical value starting with 1. For example ePWM0 is the first instance and ePWM2 is the third instance in the system and ePWMx indicates any instance.

The ePWM modules are chained together via a clock synchronization scheme that allows them to operate as a single system when required. Additionally, this synchronization scheme can be extended to the capture peripheral modules (eCAP). The number of modules is device-dependent and based on target application needs. Modules can also operate stand-alone.

Each ePWM module supports the following features:

- Dedicated 16-bit time-base counter with period and frequency control
- Two PWM outputs (EPWMxA and EPWMxB) that can be used in the following configurations::
  - Two independent PWM outputs with single-edge operation
  - Two independent PWM outputs with dual-edge symmetric operation
  - One independent PWM output with dual-edge asymmetric operation
- Asynchronous override control of PWM signals through software.
- Programmable phase-control support for lag or lead operation relative to other ePWM modules.
- Hardware-locked (synchronized) phase relationship on a cycle-by-cycle basis.
- Dead-band generation with independent rising and falling edge delay control.
- Programmable trip zone allocation of both cycle-by-cycle trip and one-shot trip on fault conditions.
- A trip condition can force either high, low, or high-impedance state logic levels at PWM outputs.
- Programmable event prescaling minimizes CPU overhead on interrupts.
- PWM chopping by high-frequency carrier signal, useful for pulse transformer gate drives.

Each ePWM module is connected to the input/output signals shown in Figure 20-7. The signals are described in detail in subsequent sections.

The order in which the ePWM modules are connected may differ from what is shown in Figure 20-7. See Section 20.2.2.3.3.2 for the synchronization scheme for a particular device. Each ePWM module consists of seven submodules and is connected within a system via the signals shown in Figure 20-8.

Figure 20-7. Multiple ePWM Modules

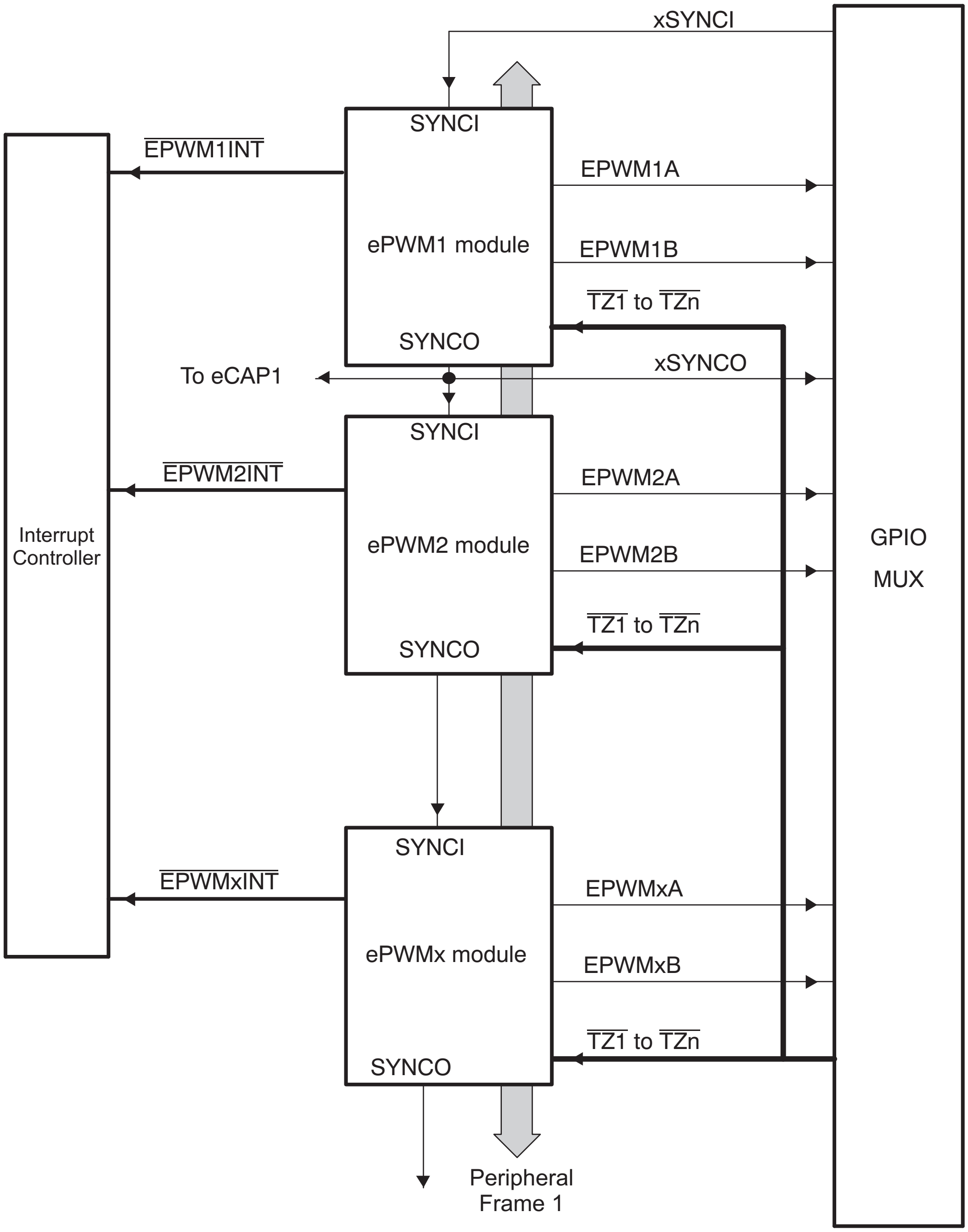


Figure 20-8. Submodules and Signal Connections for an ePWM Module

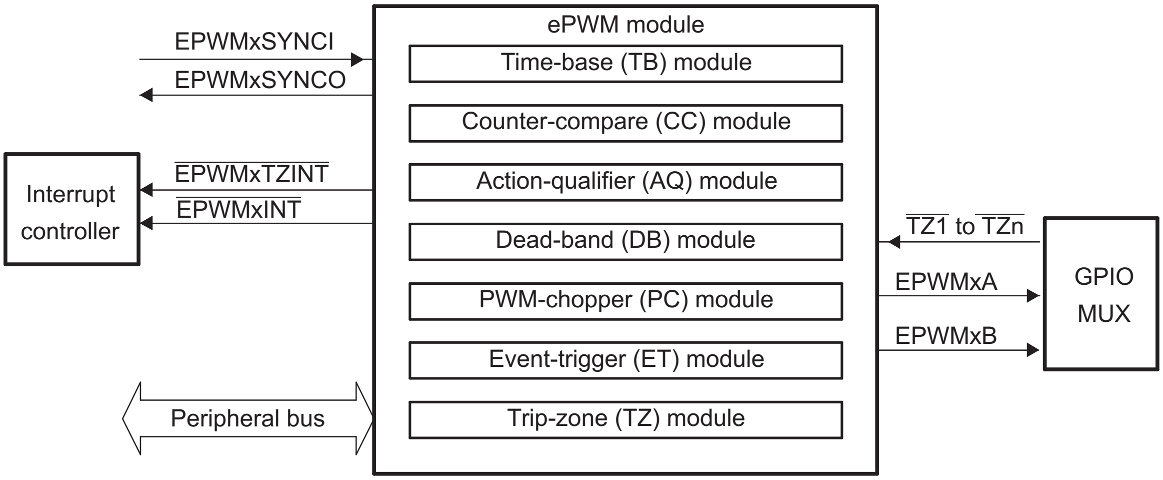
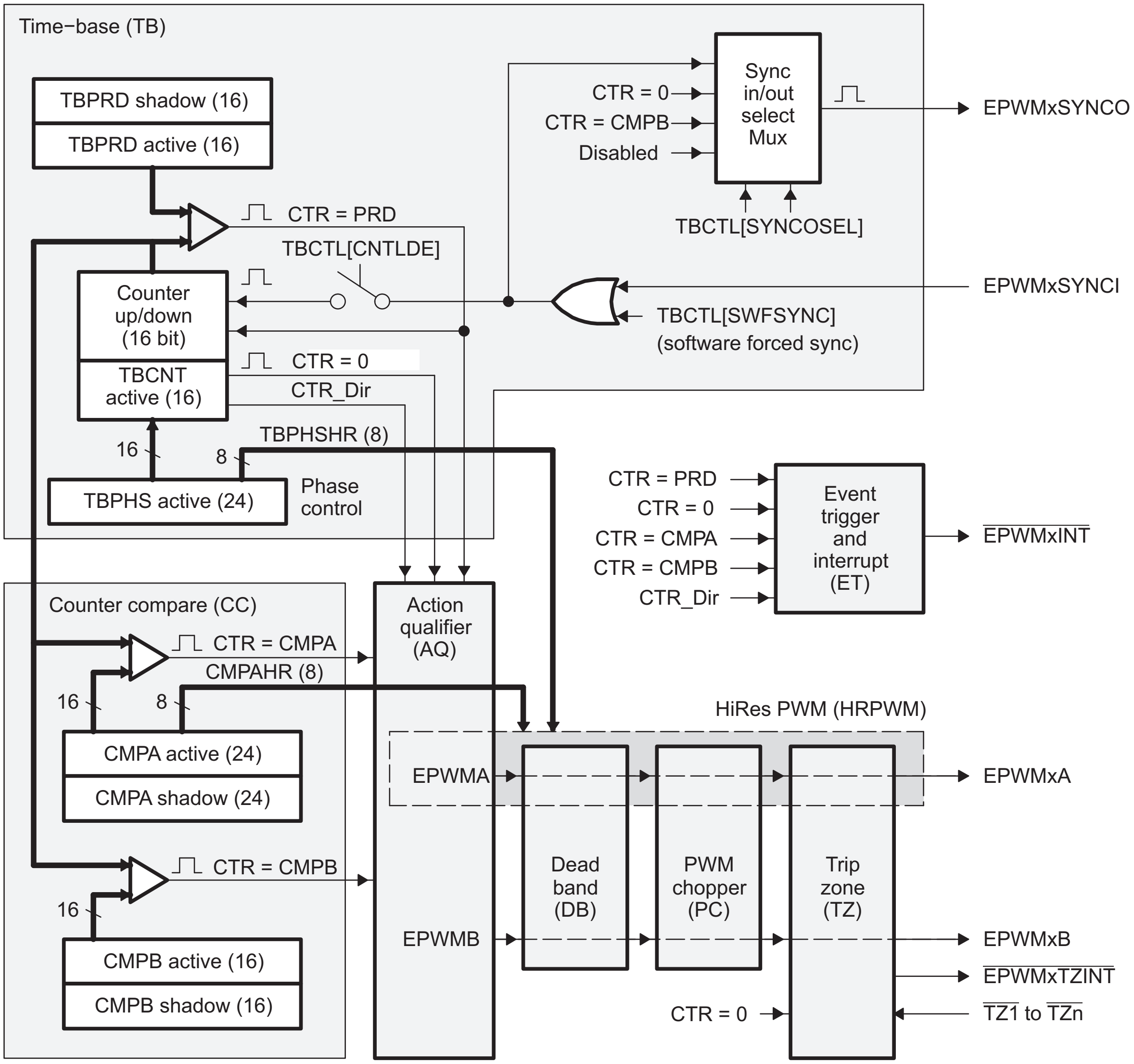


Figure 20-9 shows more internal details of a single ePWM module. The main signals used by the ePWM module are:

- **PWM output signals (EPWMxA and EPWMxB).** The PWM output signals are made available external to the device through the GPIO peripheral described in the system control and interrupts guide for your device.
- **Trip-zone signals ($\overline{\text{TZ1}}$ to $\overline{\text{TZn}}$).** These input signals alert the ePWM module of an external fault condition. Each module on a device can be configured to either use or ignore any of the trip-zone signals. The trip-zone signal can be configured as an asynchronous input through the GPIO peripheral. See Section 20.1.2 to determine how many trip-zone pins are available in the device.
- **Time-base synchronization input (EPWMxSYNCI) and output (EPWMxSYNCO) signals.** The synchronization signals daisy chain the ePWM modules together. Each module can be configured to either use or ignore its synchronization input. The clock synchronization input and output signal are brought out to pins only for ePWM1 (ePWM module #1). The synchronization output for ePWM1 (EPWM1SYNCO) is also connected to the SYNCI of the first enhanced capture module (eCAP1).
- **Peripheral Bus.** The peripheral bus is 32-bits wide and allows both 16-bit and 32-bit writes to the ePWM register file.

Figure 20-9 also shows the key internal submodule interconnect signals. Each submodule is described in Section 20.2.2.

Figure 20-9. ePWM Submodules and Critical Internal Signal Interconnects



20.2.2 Functional Description

Seven submodules are included in every ePWM peripheral. There are some instances that include a high-resolution submodule that allows more precise control of the PWM outputs. Each of these submodules performs specific tasks that can be configured by software.

20.2.2.1 Overview

Table 20-10 lists the eight key submodules together with a list of their main configuration parameters. For example, if you need to adjust or control the duty cycle of a PWM waveform, then you should see the counter-compare submodule in Section 20.2.2.4 for relevant details.

Table 20-10. Submodule Configuration Parameters

Submodule	Configuration Parameter or Option
Time-base (TB)	• Scale the time-base clock (TBCLK) relative to the system clock (SYSCLKOUT). • Configure the PWM time-base counter (TBCNT) frequency or period. • Set the mode for the time-base counter: – count-up mode: used for asymmetric PWM – count-down mode: used for asymmetric PWM – count-up-and-down mode: used for symmetric PWM • Configure the time-base phase relative to another ePWM module. • Synchronize the time-base counter between modules through hardware or software. • Configure the direction (up or down) of the time-base counter after a synchronization event. • Configure how the time-base counter will behave when the device is halted by an emulator. • Specify the source for the synchronization output of the ePWM module: – Synchronization input signal – Time-base counter equal to zero – Time-base counter equal to counter-compare B (CMPB) – No output synchronization signal generated.
Counter-compare (CC)	• Specify the PWM duty cycle for output EPWMxA and/or output EPWMxB • Specify the time at which switching events occur on the EPWMxA or EPWMxB output
Action-qualifier (AQ)	• Specify the type of action taken when a time-base or counter-compare submodule event occurs: – No action taken – Output EPWMxA and/or EPWMxB switched high – Output EPWMxA and/or EPWMxB switched low – Output EPWMxA and/or EPWMxB toggled • Force the PWM output state through software control • Configure and control the PWM dead-band through software
Dead-band (DB)	• Control of traditional complementary dead-band relationship between upper and lower switches • Specify the output rising-edge-delay value • Specify the output falling-edge delay value • Bypass the dead-band module entirely. In this case the PWM waveform is passed through without modification.
PWM-chopper (PC)	• Create a chopping (carrier) frequency. • Pulse width of the first pulse in the chopped pulse train. • Duty cycle of the second and subsequent pulses. • Bypass the PWM-chopper module entirely. In this case the PWM waveform is passed through without modification.

Table 20-10. Submodule Configuration Parameters (continued)

Submodule	Configuration Parameter or Option
Trip-zone (TZ)	• Configure the ePWM module to react to one, all, or none of the trip-zone pins. • Specify the tripping action taken when a fault occurs: – Force EPWMxA and/or EPWMxB high – Force EPWMxA and/or EPWMxB low – Force EPWMxA and/or EPWMxB to a high-impedance state – Configure EPWMxA and/or EPWMxB to ignore any trip condition. • Configure how often the ePWM will react to each trip-zone pin: – One-shot – Cycle-by-cycle • Enable the trip-zone to initiate an interrupt. • Bypass the trip-zone module entirely.
Event-trigger (ET)	• Enable the ePWM events that will trigger an interrupt. • Specify the rate at which events cause triggers (every occurrence or every second or third occurrence) • Poll, set, or clear event flags
High-Resolution PWM (HRPWM)	• Enable extended time resolution capabilities • Configure finer time granularity control or edge positioning

Code examples are provided in the remainder of this chapter that show how to implement various ePWM module configurations. These examples use the constant definitions shown in Example 20-1.

Example 20-1. Constant Definitions Used in the Code Examples

```
// TBCTL (Time-Base Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = =
// TBCNT MODE bits
#define         TB_COUNT_UP        0x0
#define         TB_COUNT_DOWN      0x1
#define         TB_COUNT_UPDOWN    0x2
#define         TB_FREEZE          0x3
// PHSEN bit
#define         TB_DISABLE         0x0
#define         TB_ENABLE          0x1
// PRDLD bit
#define         TB_SHADOW          0x0
#define         TB_IMMEDIATE       0x1
// SYNCOSEL bits
#define         TB_SYNC_IN         0x0
#define         TB_CTR_ZERO        0x1
#define         TB_CTR_CMPB        0x2
#define         TB_SYNC_DISABLE    0x3
// HSPCLKDIV and CLKDIV bits
#define         TB_DIV1            0x0
#define         TB_DIV2            0x1
#define         TB_DIV4            0x2
// PHSDIR bit
#define         TB_DOWN            0x0
#define         TB_UP              0x1

// CMPCTL (Compare Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = =
// LOADAMODE and LOADBMODE bits
#define         CC_CTR_ZERO        0x0
#define         CC_CTR_PRD         0x1
#define         CC_CTR_ZERO_PRD    0x2
#define         CC_LD_DISABLE      0x3
// SHDWAMODE and SHDWBMODE bits
```

Example 20-1. Constant Definitions Used in the Code Examples (continued)

```
#define         CC_SHADOW          0x0
#define         CC_IMMEDIATE       0x1
// AQCTLA and AQCTLB (Action-qualifier Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = =
// ZRO, PRD, CAU, CAD, CBU, CBD bits
#define         AQ_NO_ACTION       0x0
#define         AQ_CLEAR           0x1
#define         AQ_SET             0x2
#define         AQ_TOGGLE          0x3
// DBCTL (Dead-Band Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = =
// MODE bits
#define         DB_DISABLE         0x0
#define         DBA_ENABLE         0x1
#define         DBB_ENABLE         0x2
#define DB_FULL_ENABLE 0x3
// POLSEL bits
#define         DB_ACTV_HI         0x0
#define         DB_ACTV_LOC        0x1
#define         DB_ACTV_HIC        0x2
#define         DB_ACTV_LO         0x3
// PCCTL (chopper control)
// = = = = = = = = = = = = = = = = = = = = = = = = = =
// CHPEN bit
#define         CHP_ENABLE         0x0
#define CHP_DISABLE 0x1
// CHPFREQ bits
#define         CHP_DIV1           0x0
#define         CHP_DIV2           0x1
#define         CHP_DIV3           0x2
#define         CHP_DIV4           0x3
#define         CHP_DIV5           0x4
#define         CHP_DIV6           0x5
#define         CHP_DIV7           0x6
#define         CHP_DIV8           0x7
// CHPDUTY bits
#define         CHP1_8TH           0x0
#define         CHP2_8TH           0x1
#define         CHP3_8TH           0x2
#define         CHP4_8TH           0x3
#define         CHP5_8TH           0x4
#define         CHP6_8TH           0x5
#define         CHP7_8TH           0x6
// TZSEL (Trip-zone Select)
// = = = = = = = = = = = = = = = = = = = = = = = = = =
// CBCn and OSHTn bits
#define         TZ_ENABLE          0x0
#define         TZ_DISABLE         0x1
// TZCTL (Trip-zone Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = =
// TZA and TZB bits
#define         TZ_HIZ             0x0
#define         TZ_FORCE_HI        0x1
#define         TZ_FORCE_LO        0x2
#define         TZ_DISABLE         0x3
// ETSEL (Event-trigger Select)
// = = = = = = = = = = = = = = = = = = = = = = = = = =
// INTSEL, SOCASEL, SOCBSEL bits
#define         ET_CTR_ZERO        0x1
#define         ET_CTR_PRD         0x2
#define         ET_CTRU_CMPA       0x4
#define         ET_CTRD_CMPA       0x5
#define         ET_CTRU_CMPB       0x6
#define         ET_CTRD_CMPB       0x7
```

Example 20-1. Constant Definitions Used in the Code Examples (continued)

```
// ETPS (Event-trigger Prescale)
// = = = = = = = = = = = = = = = = = = = = = = = = = = =
// INTPRD, SOCAPRD, SOCBPRD bits
#define         ET_DISABLE         0x0
#define         ET_1ST             0x1
#define         ET_2ND             0x2
#define         ET_3RD             0x3
```

20.2.2.2 Proper Interrupt Initialization Procedure

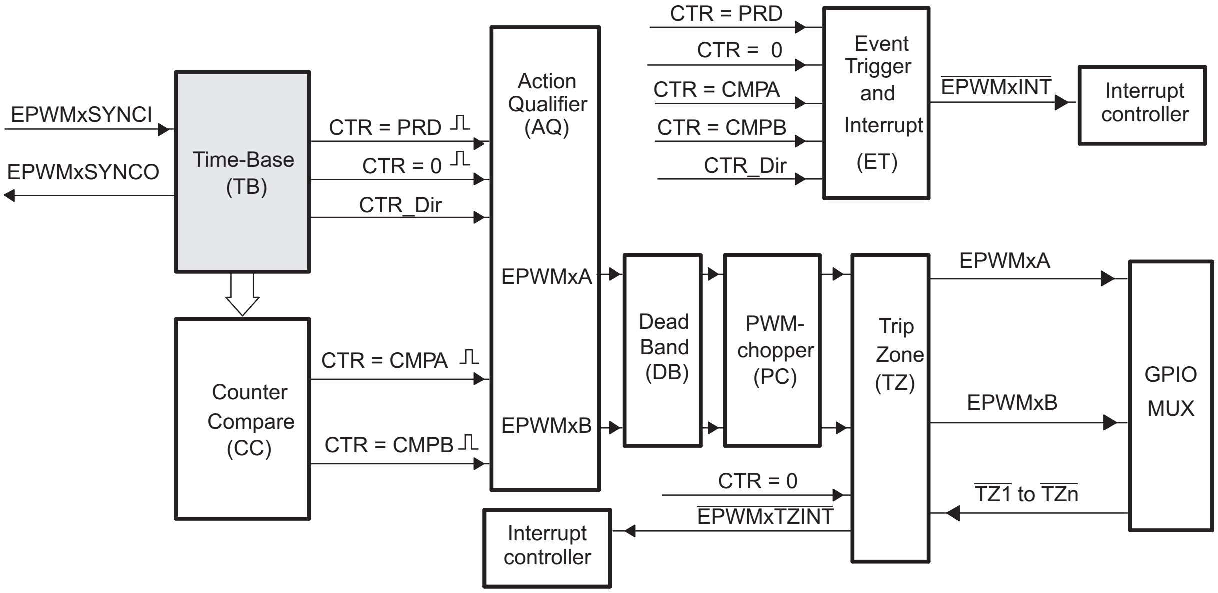
When the ePWM peripheral clock is enabled it may be possible that interrupt flags may be set due to spurious events due to the ePWM registers not being properly initialized. The proper procedure for initializing the ePWM peripheral is:

1. Disable global interrupts (CPU INTM flag)
2. Disable ePWM interrupts
3. Initialize peripheral registers
4. Clear any spurious ePWM flags
5. Enable ePWM interrupts
6. Enable global interrupts

20.2.2.3 Time-Base (TB) Submodule

Each ePWM module has its own time-base submodule that determines all of the event timing for the ePWM module. Built-in synchronization logic allows the time-base of multiple ePWM modules to work together as a single system. Figure 20-10 illustrates the time-base module's place within the ePWM.

Figure 20-10. Time-Base Submodule Block Diagram



20.2.2.3.1 Purpose of the Time-Base Submodule

You can configure the time-base submodule for the following:

- Specify the ePWM time-base counter (TBCNT) frequency or period to control how often events occur.
- Manage time-base synchronization with other ePWM modules.
- Maintain a phase relationship with other ePWM modules.
- Set the time-base counter to count-up, count-down, or count-up-and-down mode.
- Generate the following events:
  - CTR = PRD: Time-base counter equal to the specified period (TBCNT = TBPRD) .
  - CTR = 0: Time-base counter equal to zero (TBCNT = 0000h).
- Configure the rate of the time-base clock; a prescaled version of the CPU system clock (SYSCLKOUT). This allows the time-base counter to increment/decrement at a slower rate.

20.2.2.3.2 Controlling and Monitoring the Time-Base Submodule

Table 20-11 lists the registers used to control and monitor the time-base submodule.

Table 20-11. Time-Base Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
TBCTL	Time-Base Control Register	0h	No
TBSTS	Time-Base Status Register	2h	No
TBPHSHR	HRPWM extension Phase Register (1)	4h	No
TBPHS	Time-Base Phase Register	6h	No
TBCNT	Time-Base Counter Register	8h	No
TBPRD	Time-Base Period Register	Ah	Yes

(1) This register is available only on ePWM instances that include the high-resolution extension (HRPWM). On ePWM modules that do not include the HRPWM, this location is reserved. See Section 20.1.2 to determine which ePWM instances include this feature.

Figure 20-11 shows the critical signals and registers of the time-base submodule. Table 20-12 provides descriptions of the key signals associated with the time-base submodule.

Figure 20-11. Time-Base Submodule Signals and Registers

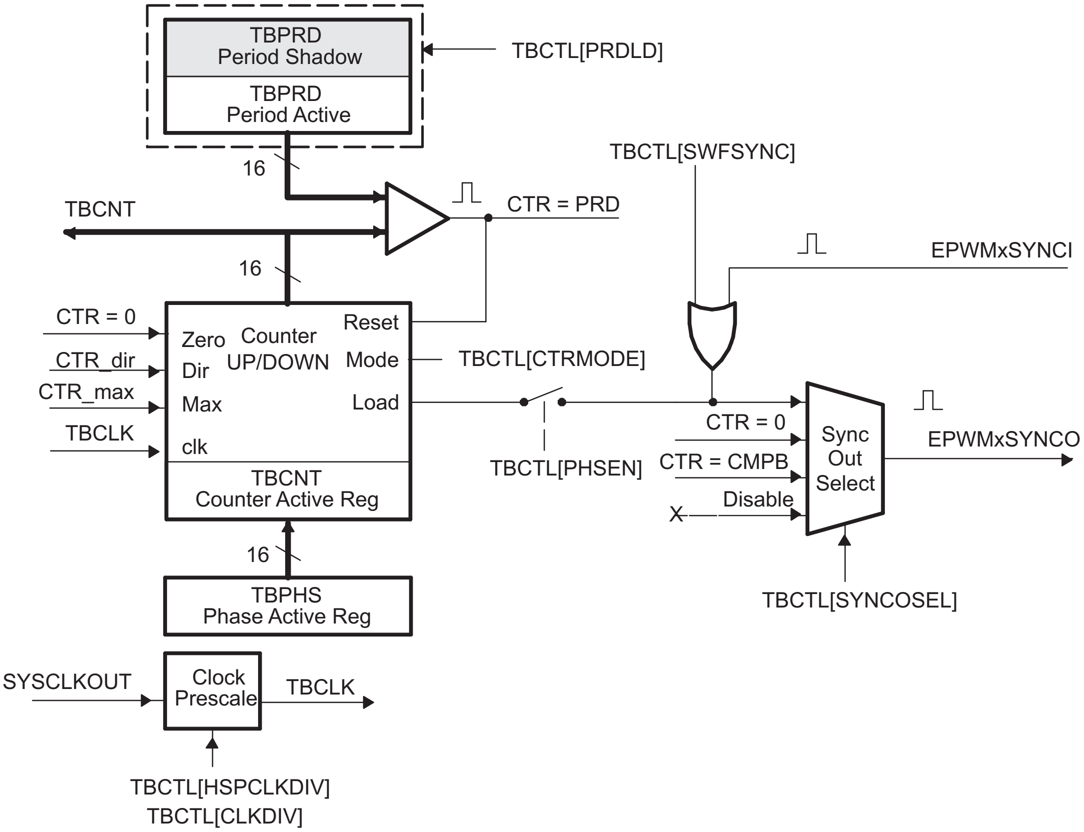


Table 20-12. Key Time-Base Signals

Signal	Description
EPWMxSYNCI	Time-base synchronization input. Input pulse used to synchronize the time-base counter with the counter of ePWM module earlier in the synchronization chain. An ePWM peripheral can be configured to use or ignore this signal. For example, this signal could come from a device pin for the first ePWM module (ePWM0). For subsequent ePWM modules this signal could be passed from another ePWM peripheral, such that EPWM1SYNCI is generated by the ePWM0 peripheral, EPWM2SYNCI is generated by ePWM1, and so forth. See Section 20.1.2 for information on the synchronization order of a particular device.
EPWMxSYNCO	Time-base synchronization output. This output pulse is used to synchronize the counter of an ePWM module later in the synchronization chain. The ePWM module generates this signal from one of three event sources: 1. EPWMxSYNCI (Synchronization input pulse) 2. CTR = 0: The time-base counter equal to zero (TBCNT = 0000h). 3. CTR = CMPB: The time-base counter equal to the counter-compare B (TBCNT = CMPB) register.
CTR = PRD	Time-base counter equal to the specified period. This signal is generated whenever the counter value is equal to the active period register value. That is when TBCNT = TBPRD.
CTR = 0	Time-base counter equal to zero. This signal is generated whenever the counter value is zero. That is when TBCNT equals 0000h.
CTR = CMPB	Time-base counter equal to active counter-compare B register (TBCNT = CMPB). This event is generated by the counter-compare submodule and used by the synchronization out logic.
CTR_dir	Time-base counter direction. Indicates the current direction of the ePWM's time-base counter. This signal is high when the counter is increasing and low when it is decreasing.
CTR_max	Time-base counter equal max value. (TBCNT = FFFFh) Generated event when the TBCNT value reaches its maximum value. This signal is only used only as a status bit.
TBCLK	Time-base clock. This is a prescaled version of the system clock (SYSCLKOUT) and is used by all submodules within the ePWM. This clock determines the rate at which time-base counter increments or decrements.

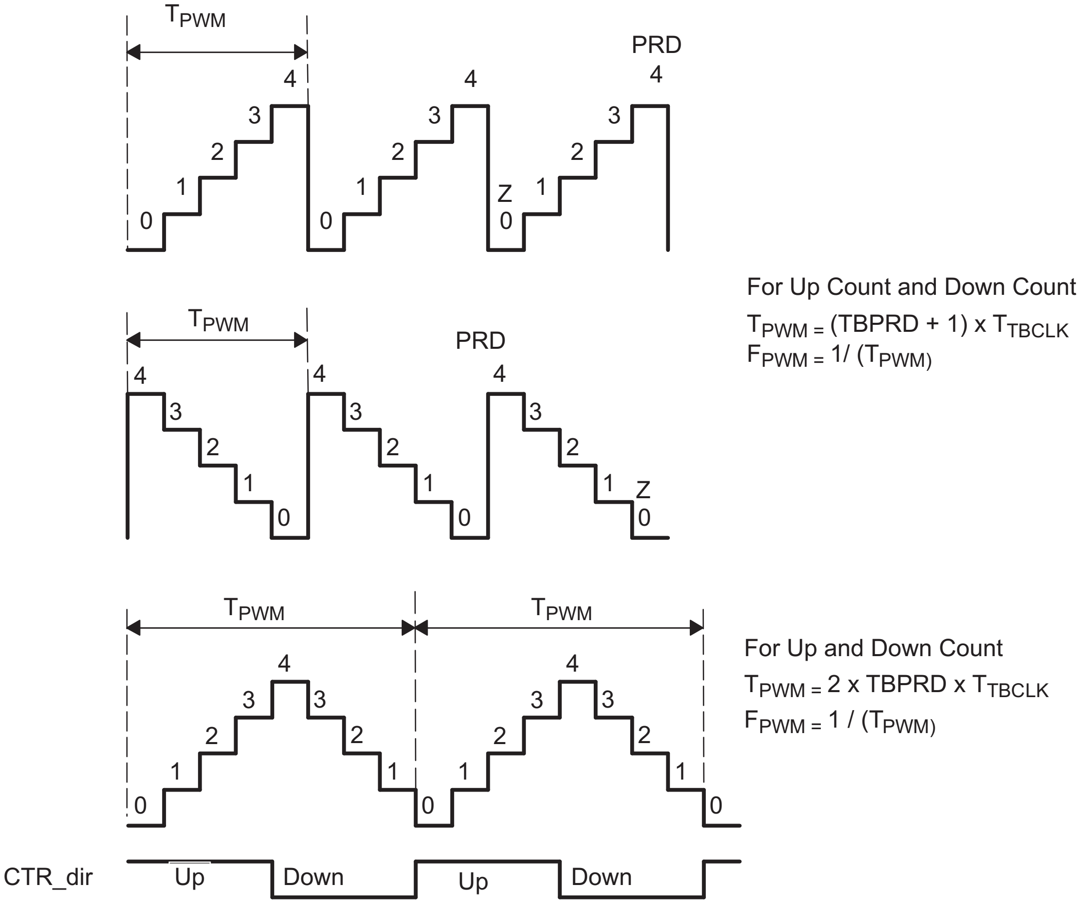
20.2.2.3.3 Calculating PWM Period and Frequency

The frequency of PWM events is controlled by the time-base period (TBPRD) register and the mode of the time-base counter. Figure 20-12 shows the period (T_{pwm}) and frequency (F_{pwm}) relationships for the up-count, down-count, and up-down-count time-base counter modes when when the period is set to 4 (TBPRD = 4). The time increment for each step is defined by the time-base clock (TBCLK) which is a prescaled version of the system clock (SYSCLKOUT).

The time-base counter has three modes of operation selected by the time-base control register (TBCTL):

- **Up-Down-Count Mode:** In up-down-count mode, the time-base counter starts from zero and increments until the period (TBPRD) value is reached. When the period value is reached, the time-base counter then decrements until it reaches zero. At this point the counter repeats the pattern and begins to increment.
- **Up-Count Mode:** In this mode, the time-base counter starts from zero and increments until it reaches the value in the period register (TBPRD). When the period value is reached, the time-base counter resets to zero and begins to increment once again.
- **Down-Count Mode:** In down-count mode, the time-base counter starts from the period (TBPRD) value and decrements until it reaches zero. When it reaches zero, the time-base counter is reset to the period value and it begins to decrement once again.

Figure 20-12. Time-Base Frequency and Period



For Up Count and Down Count

$T_{PWM} = (TBPRD + 1) \times T_{TBCLK}$

$F_{PWM} = 1/(T_{PWM})$

For Up and Down Count

$T_{PWM} = 2 \times TBPRD \times T_{TBCLK}$

$F_{PWM} = 1/(T_{PWM})$

20.2.2.3.3.1 Time-Base Period Shadow Register

The time-base period register (TBPRD) has a shadow register. Shadowing allows the register update to be synchronized with the hardware. The following definitions are used to describe all shadow registers in the ePWM module:

- **Active Register:** The active register controls the hardware and is responsible for actions that the hardware causes or invokes.
- **Shadow Register:** The shadow register buffers or provides a temporary holding location for the active register. It has no direct effect on any control hardware. At a strategic point in time the shadow register's content is transferred to the active register. This prevents corruption or spurious operation due to the register being asynchronously modified by software.

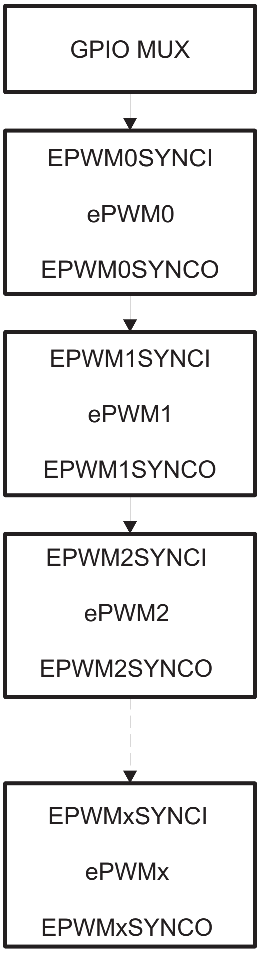
The memory address of the shadow period register is the same as the active register. Which register is written to or read from is determined by the TBCTL[PRDLD] bit. This bit enables and disables the TBPRD shadow register as follows:

- **Time-Base Period Shadow Mode:** The TBPRD shadow register is enabled when TBCTL[PRDLD] = 0. Reads from and writes to the TBPRD memory address go to the shadow register. The shadow register contents are transferred to the active register (TBPRD (Active) ← TBPRD (shadow)) when the time-base counter equals zero (TBCNT = 0000h). By default the TBPRD shadow register is enabled.
- **Time-Base Period Immediate Load Mode:** If immediate load mode is selected (TBCTL[PRDLD] = 1), then a read from or a write to the TBPRD memory address goes directly to the active register.

20.2.2.3.3.2 Time-Base Counter Synchronization

A time-base synchronization scheme connects all of the ePWM modules on a device. Each ePWM module has a synchronization input (EPWMxSYNCI) and a synchronization output (EPWMxSYNCO). The synchronization input can come from an external pin or another ePWM module. An example of synchronization connections for the remaining ePWM modules is shown in Section 20.1.2.

Figure 20-13. Time-Base Counter Synchronization Scheme 1



Each ePWM module can be configured to use or ignore the synchronization input. If the TBCTL[PHSEN] bit is set, then the time-base counter (TBCNT) of the ePWM module will be automatically loaded with the phase register (TBPHS) contents when one of the following conditions occur:

- **EPWMxSYNCI: Synchronization Input Pulse:** The value of the phase register is loaded into the counter register when an input synchronization pulse is detected (TBPHS → TBCNT). This operation occurs on the next valid time-base clock (TBCLK) edge.
- **Software Forced Synchronization Pulse:** Writing a 1 to the TBCTL[SWFSYNC] control bit invokes a software forced synchronization. This pulse is ORed with the synchronization input signal, and therefore has the same effect as a pulse on EPWMxSYNCI.

This feature enables the ePWM module to be automatically synchronized to the time base of another ePWM module. Lead or lag phase control can be added to the waveforms generated by different ePWM modules to synchronize them. In up-down-count mode, the TBCTL[PSHDIR] bit configures the direction of the time-base counter immediately after a synchronization event. The new direction is independent of the direction prior to the synchronization event. The TBPHS bit is ignored in count-up or count-down modes. See Figure 20-14 through Figure 20-17 for examples.

Clearing the TBCTL[PHSEN] bit configures the ePWM to ignore the synchronization input pulse. The synchronization pulse can still be allowed to flow-through to the EPWMxSYNCO and be used to synchronize other ePWM modules. In this way, you can set up a master time-base (for example, ePWM0) and downstream modules (ePWM1 – ePWMx) may elect to run in synchronization with the master.

20.2.2.3.4 Phase Locking the Time-Base Clocks of Multiple ePWM Modules

The TBCLKEN bit in the PWMSS_CTRL register in the Control Module can be used to globally synchronize the time-base clocks of all enabled ePWM modules on a device. The TBCLKEN bit is part of the chip configuration registers and is described in Chapter 7. When TBCLKEN = 0, the time-base clock of all ePWM modules is stopped (default). When TBCLKEN = 1, all ePWM time-base clocks are started with the rising edge of TBCLK aligned. For perfectly synchronized TBCLKs, the prescaler bits in the TBCTL register of each ePWM module must be set identically. The proper procedure for enabling the ePWM clocks is as follows:

1. Enable the ePWM module clocks.
2. Set TBCLKEN = 0. This will stop the time-base clock within any enabled ePWM module.
3. Configure the prescaler values and desired ePWM modes.
4. Set TBCLKEN = 1.

20.2.2.3.5 Time-Base Counter Modes and Timing Waveforms

The time-base counter operates in one of four modes:

- Up-count mode which is asymmetrical.
- Down-count mode which is asymmetrical.
- Up-down-count which is symmetrical.
- Frozen where the time-base counter is held constant at the current value.

To illustrate the operation of the first three modes, Figure 20-14 to Figure 20-17 show when events are generated and how the time-base responds to an EPWMxSYNCI signal.

Figure 20-14. Time-Base Up-Count Mode Waveforms

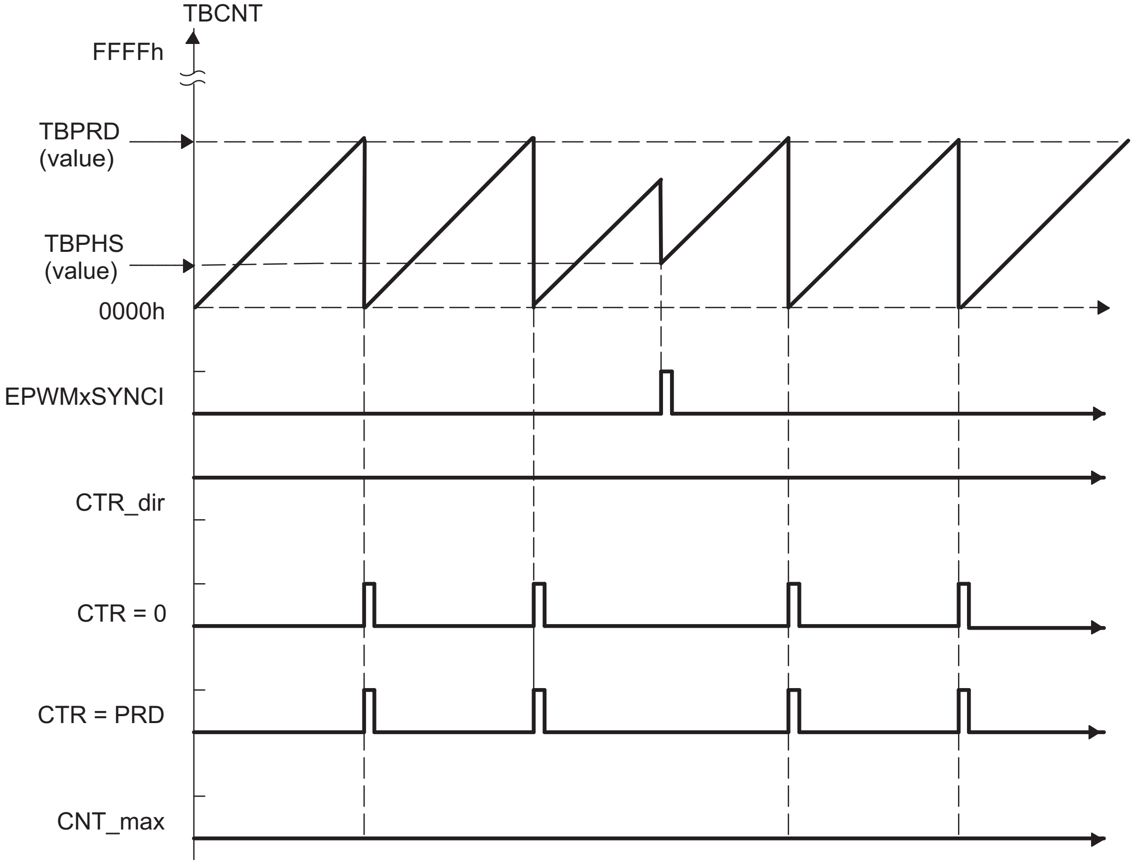


Figure 20-15. Time-Base Down-Count Mode Waveforms

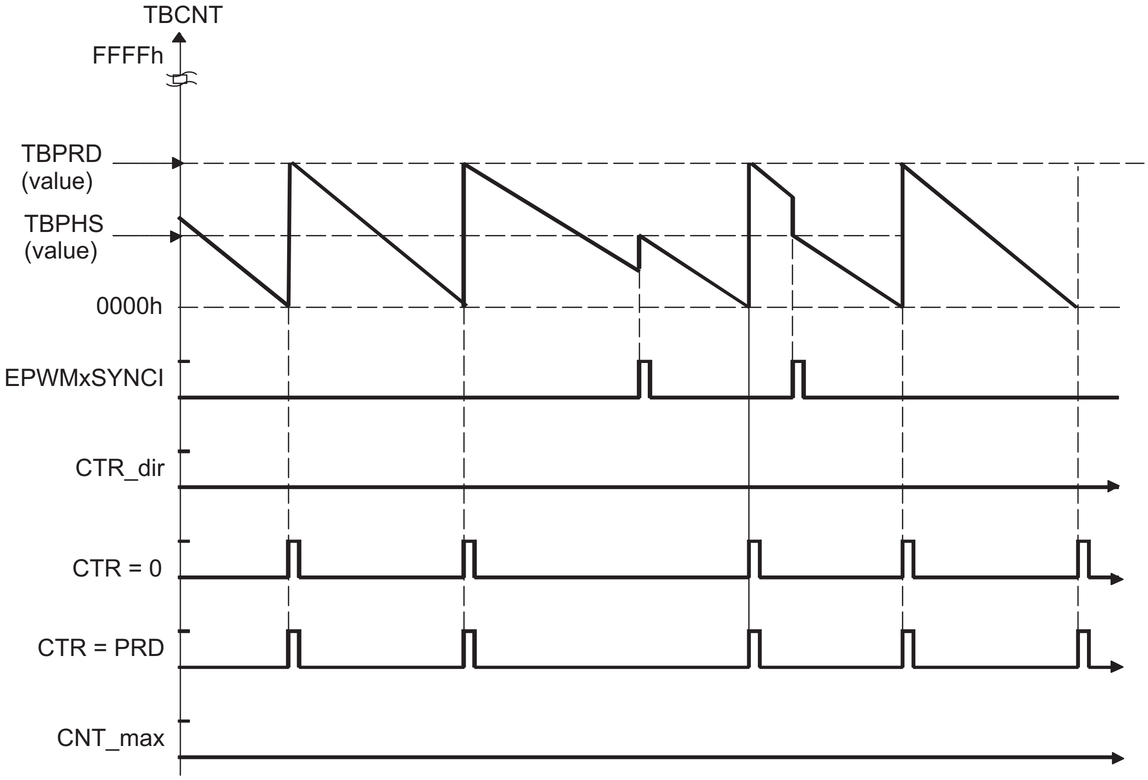


Figure 20-16. Time-Base Up-Down-Count Waveforms, TBCTL[PHSDIR = 0] Count Down on Synchronization Event

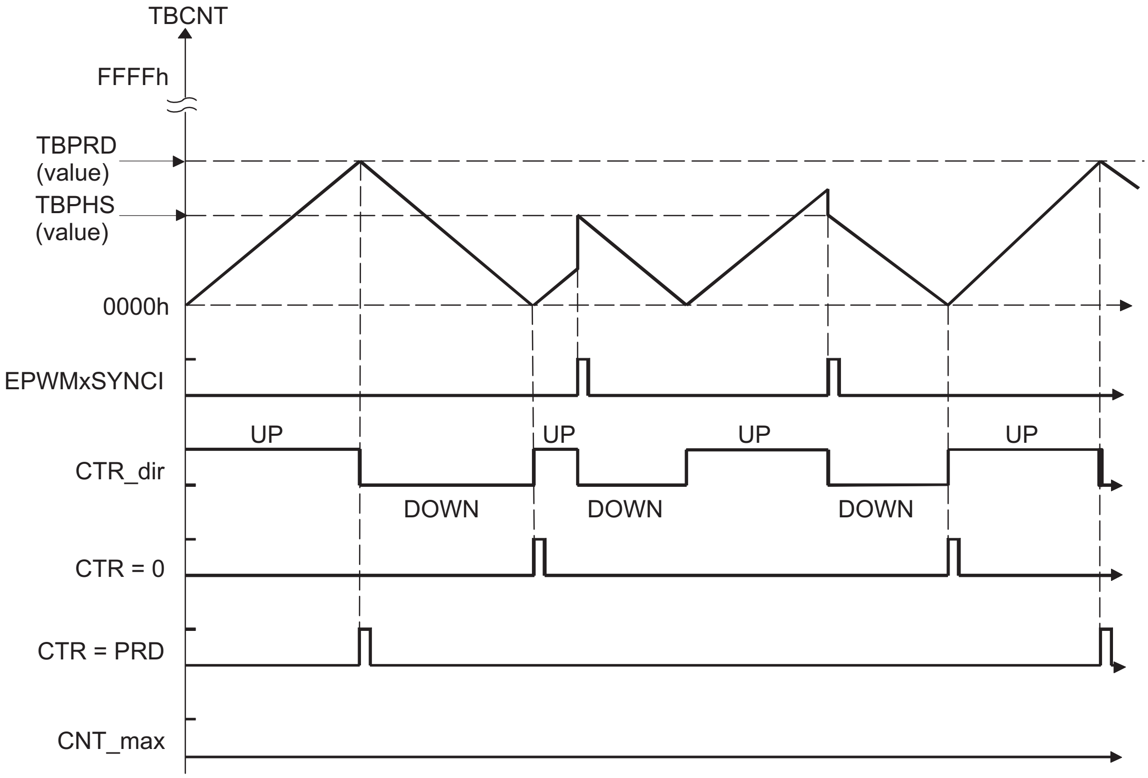
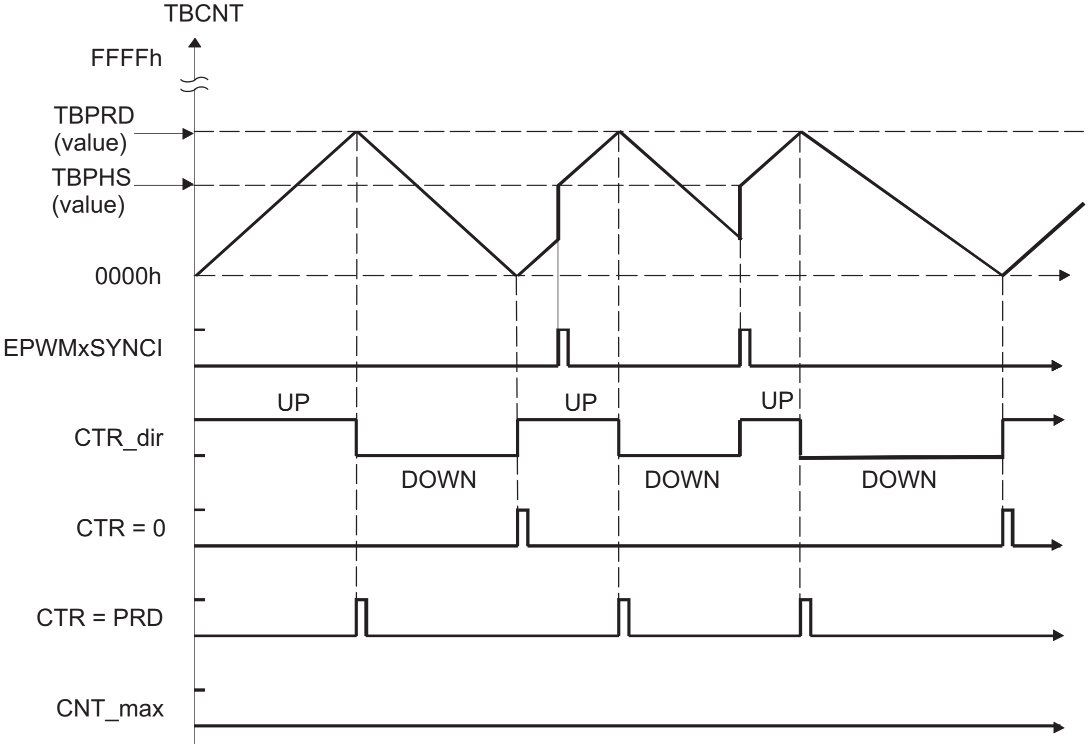


Figure 20-17. Time-Base Up-Down Count Waveforms, TBCTL[PHSDIR = 1] Count Up on Synchronization Event



20.2.2.4 Counter-Compare (CC) Submodule

Figure 20-18 illustrates the counter-compare submodule within the ePWM. Figure 20-19 shows the basic structure of the counter-compare submodule.

Figure 20-18. Counter-Compare Submodule

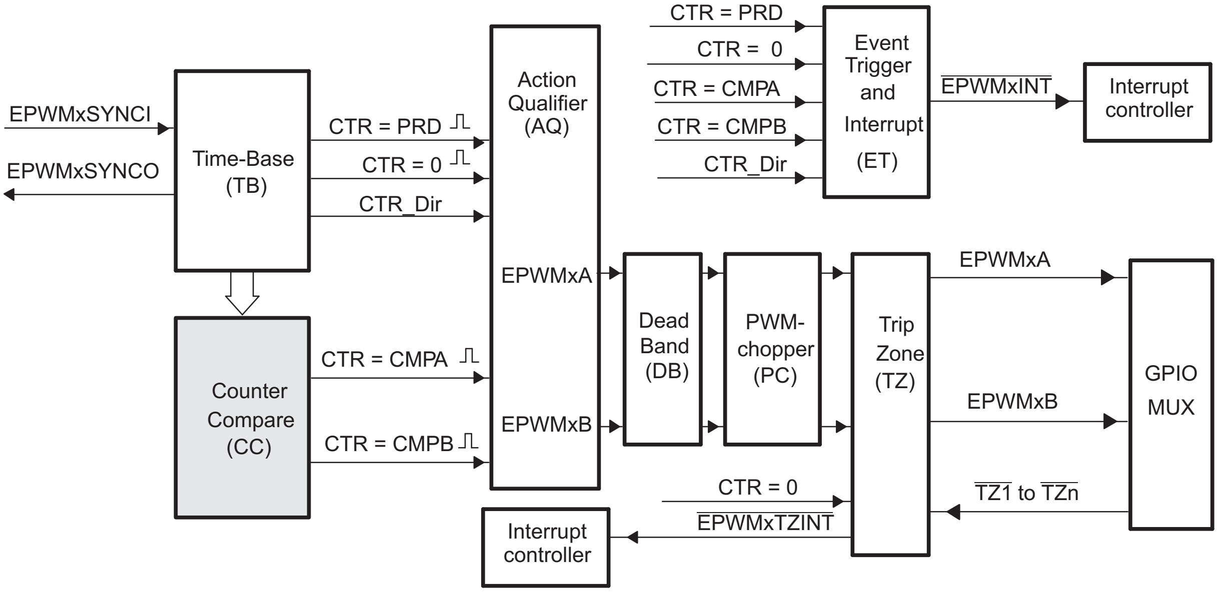
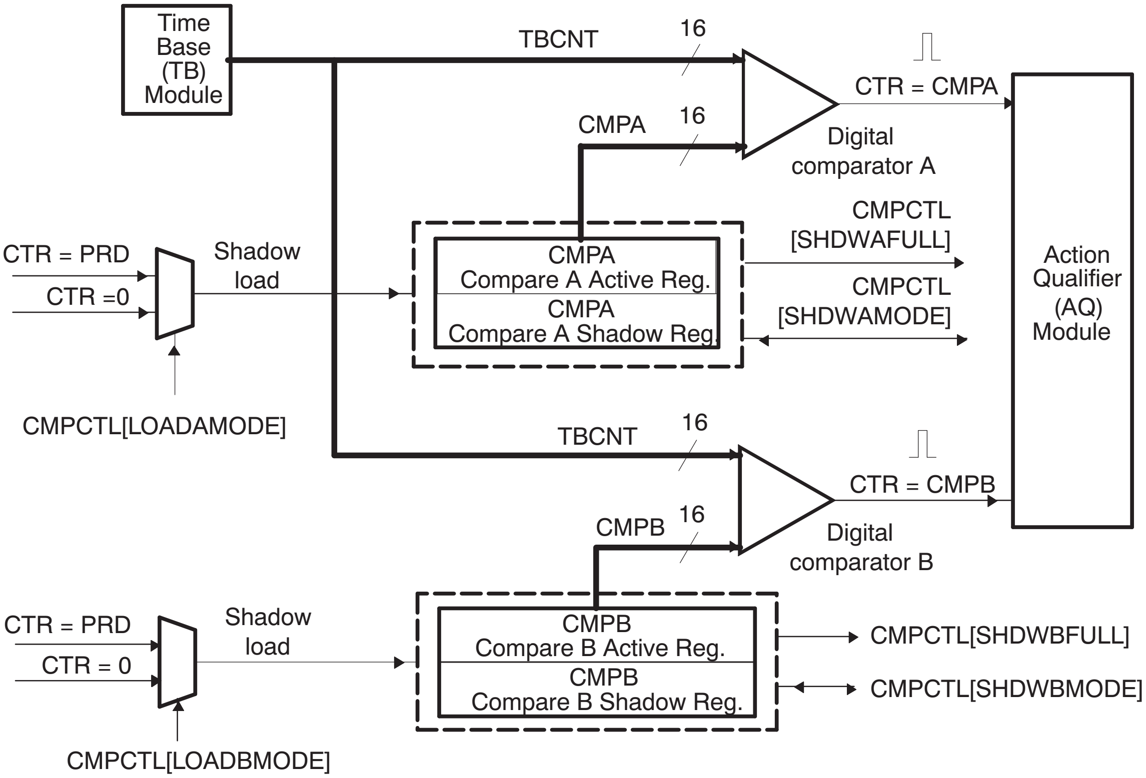


Figure 20-19. Counter-Compare Submodule Signals and Registers



20.2.2.4.1 Purpose of the Counter-Compare Submodule

The counter-compare submodule takes as input the time-base counter value. This value is continuously compared to the counter-compare A (CMPA) and counter-compare B (CMPB) registers. When the time-base counter is equal to one of the compare registers, the counter-compare unit generates an appropriate event.

The counter-compare submodule:

- Generates events based on programmable time stamps using the CMPA and CMPB registers
  - CTR = CMPA: Time-base counter equals counter-compare A register (TBCNT = CMPA).
  - CTR = CMPB: Time-base counter equals counter-compare B register (TBCNT = CMPB)
- Controls the PWM duty cycle if the action-qualifier submodule is configured appropriately
- Shadows new compare values to prevent corruption or glitches during the active PWM cycle

20.2.2.4.2 Controlling and Monitoring the Counter-Compare Submodule

Table 20-13 lists the registers used to control and monitor the counter-compare submodule. Table 20-14 lists the key signals associated with the counter-compare submodule.

Table 20-13. Counter-Compare Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
CMPCTL	Counter-Compare Control Register.	Eh	No
CMPAHR	HRPWM Counter-Compare A Extension Register (1)	10h	Yes
CMPA	Counter-Compare A Register	12h	Yes
CMPB	Counter-Compare B Register	14h	Yes

(1) This register is available only on ePWM modules with the high-resolution extension (HRPWM). On ePWM modules that do not include the HRPWM, this location is reserved. See Section 20.1.2 to determine which ePWM instances include this feature.

Table 20-14. Counter-Compare Submodule Key Signals

Signal	Description of Event	Registers Compared
CTR = CMPA	Time-base counter equal to the active counter-compare A value	TBCNT = CMPA
CTR = CMPB	Time-base counter equal to the active counter-compare B value	TBCNT = CMPB
CTR = PRD	Time-base counter equal to the active period. Used to load active counter-compare A and B registers from the shadow register	TBCNT = TBPRD
CTR = 0	Time-base counter equal to zero. Used to load active counter-compare A and B registers from the shadow register	TBCNT = 0000h

20.2.2.4.3 Operational Highlights for the Counter-Compare Submodule

The counter-compare submodule is responsible for generating two independent compare events based on two compare registers:

1. CTR = CMPA: Time-base counter equal to counter-compare A register (TBCNT = CMPA).
2. CTR = CMPB: Time-base counter equal to counter-compare B register (TBCNT = CMPB).

For up-count or down-count mode, each event occurs only once per cycle. For up-down-count mode each event occurs twice per cycle, if the compare value is between 0000h and TBPRD; and occurs once per cycle, if the compare value is equal to 0000h or equal to TBPRD. These events are fed into the action-qualifier submodule where they are qualified by the counter direction and converted into actions if enabled. Refer to Section 20.2.2.5.1 for more details.

The counter-compare registers CMPA and CMPB each have an associated shadow register. Shadowing provides a way to keep updates to the registers synchronized with the hardware. When shadowing is used, updates to the active registers only occurs at strategic points. This prevents corruption or spurious operation due to the register being asynchronously modified by software. The memory address of the active register and the shadow register is identical. Which register is written to or read from is determined by the CMPCTL[SHDWAMODE] and CMPCTL[SHDWBMODE] bits. These bits enable and disable the CMPA shadow register and CMPB shadow register respectively. The behavior of the two load modes is described below:

- **Shadow Mode:** The shadow mode for the CMPA is enabled by clearing the CMPCTL[SHDWAMODE] bit and the shadow register for CMPB is enabled by clearing the CMPCTL[SHDWBMODE] bit. Shadow mode is enabled by default for both CMPA and CMPB.

  If the shadow register is enabled then the content of the shadow register is transferred to the active register on one of the following events:

  - CTR = PRD: Time-base counter equal to the period (TBCNT = TBPRD).
  - CTR = 0: Time-base counter equal to zero (TBCNT = 0000h)
  - Both CTR = PRD and CTR = 0

  Which of these three events is specified by the CMPCTL[LOADAMODE] and CMPCTL[LOADBMODE] register bits. Only the active register contents are used by the counter-compare submodule to generate events to be sent to the action-qualifier.
- **Immediate Load Mode:** If immediate load mode is selected (TBCTL[SHADWAMODE] = 1 or TBCTL[SHADWBMODE] = 1), then a read from or a write to the register will go directly to the active register.

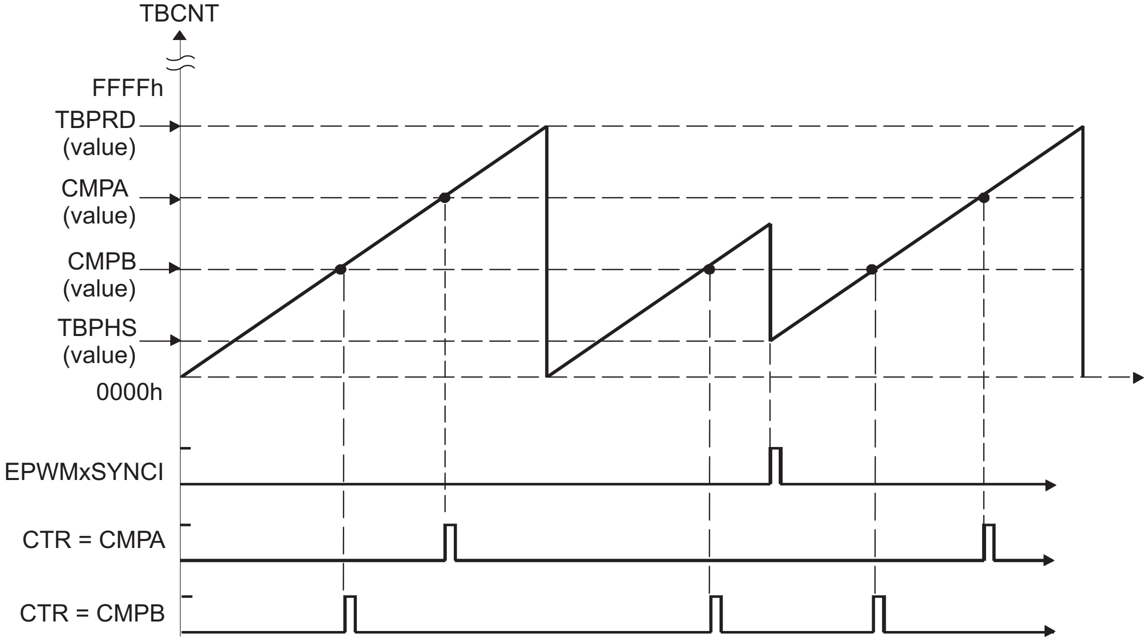
20.2.2.4.4 Count Mode Timing Waveforms

The counter-compare module can generate compare events in all three count modes:

- Up-count mode: used to generate an asymmetrical PWM waveform.
- Down-count mode: used to generate an asymmetrical PWM waveform.
- Up-down-count mode: used to generate a symmetrical PWM waveform.

To best illustrate the operation of the first three modes, the timing diagrams in Figure 20-20 to Figure 20-23 show when events are generated and how the EPWMxSYNCI signal interacts.

Figure 20-20. Counter-Compare Event Waveforms in Up-Count Mode



NOTE: An EPWMxSYNCI external synchronization event can cause a discontinuity in the TBCNT count sequence. This can lead to a compare event being skipped. This skipping is considered normal operation and must be taken into account.

Figure 20-21. Counter-Compare Events in Down-Count Mode

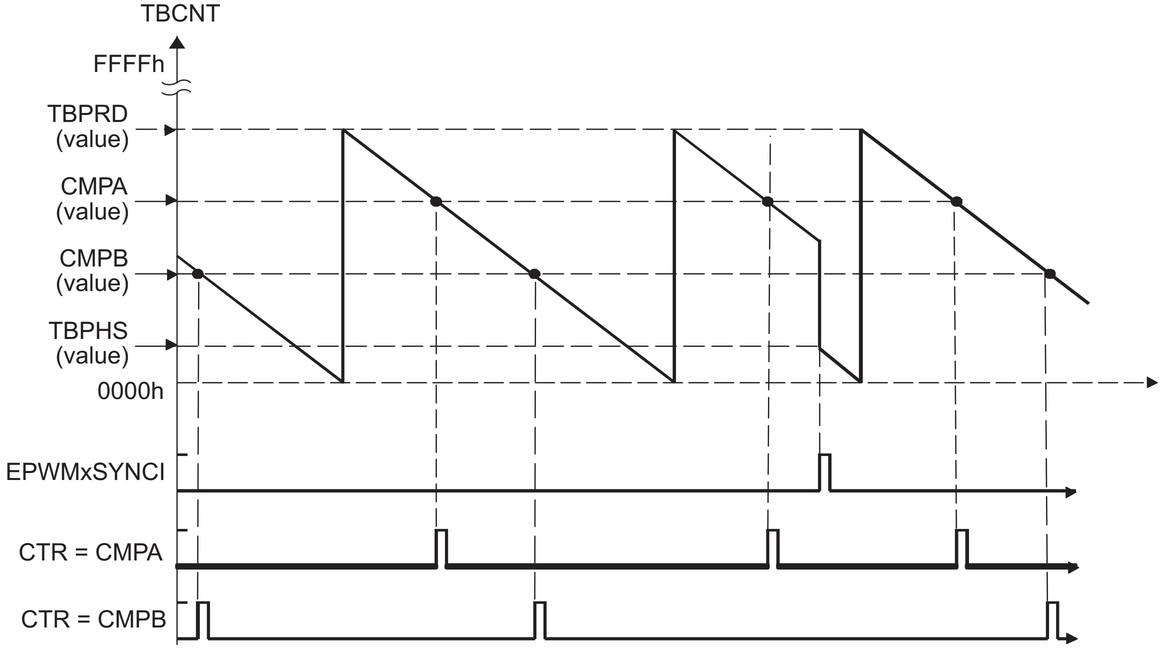


Figure 20-22. Counter-Compare Events in Up-Down-Count Mode, TBCTL[PHSDIR = 0] Count Down on Synchronization Event

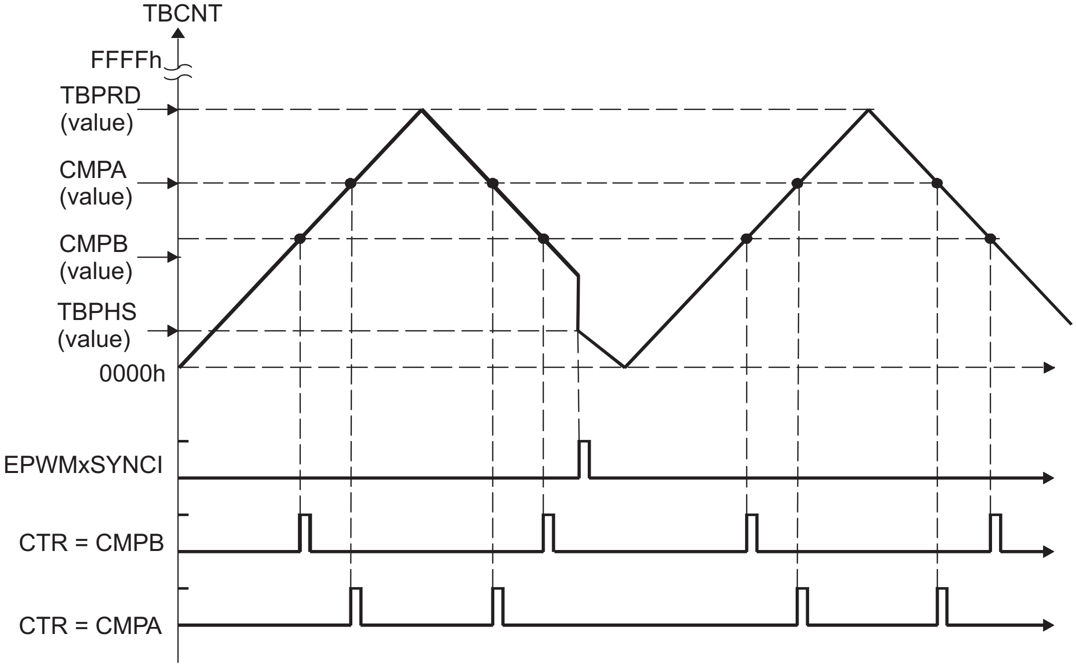
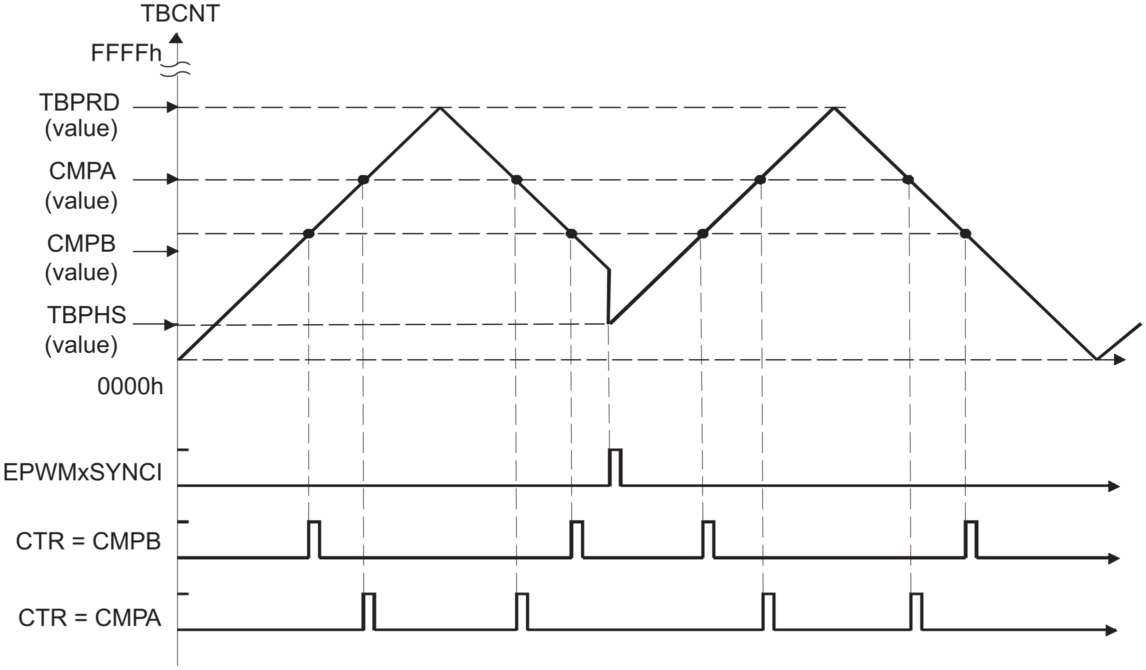


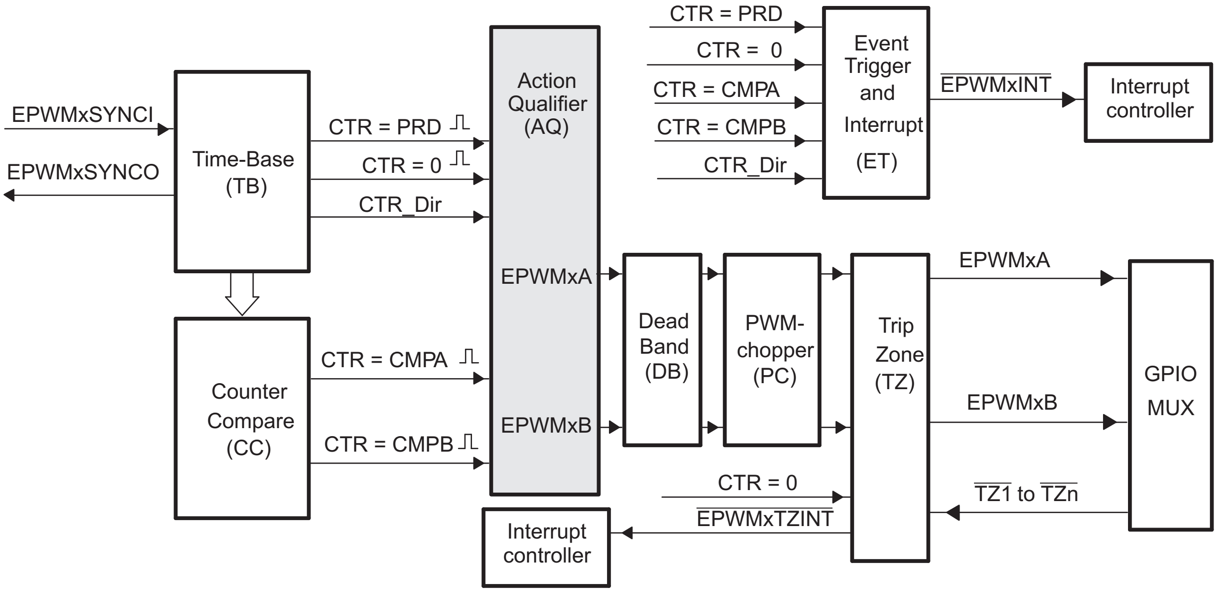
Figure 20-23. Counter-Compare Events in Up-Down-Count Mode, TBCTL[PHSDIR = 1] Count Up on Synchronization Event



20.2.2.5 Action-Qualifier (AQ) Submodule

Figure 20-24 shows the action-qualifier (AQ) submodule (see shaded block) in the ePWM system. The action-qualifier submodule has the most important role in waveform construction and PWM generation. It decides which events are converted into various action types, thereby producing the required switched waveforms at the EPWMxA and EPWMxB outputs.

Figure 20-24. Action-Qualifier Submodule



20.2.2.5.1 *Purpose of the Action-Qualifier Submodule*

The action-qualifier submodule is responsible for the following:

- Qualifying and generating actions (set, clear, toggle) based on the following events:
  - CTR = PRD: Time-base counter equal to the period (TBCNT = TBPRD)
  - CTR = 0: Time-base counter equal to zero (TBCNT = 0000h)
  - CTR = CMPA: Time-base counter equal to the counter-compare A register (TBCNT = CMPA)
  - CTR = CMPB: Time-base counter equal to the counter-compare B register (TBCNT = CMPB)
- Managing priority when these events occur concurrently
- Providing independent control of events when the time-base counter is increasing and when it is decreasing.

20.2.2.5.2 *Controlling and Monitoring the Action-Qualifier Submodule*

Table 20-15 lists the registers used to control and monitor the action-qualifier submodule.

Table 20-15. Action-Qualifier Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
AQCTLA	Action-Qualifier Control Register For Output A (EPWMxA)	16h	No
AQCTLB	Action-Qualifier Control Register For Output B (EPWMxB)	18h	No
AQSFRC	Action-Qualifier Software Force Register	1Ah	No
AQCSFRC	Action-Qualifier Continuous Software Force	1Ch	Yes

The action-qualifier submodule is based on event-driven logic. It can be thought of as a programmable cross switch with events at the input and actions at the output, all of which are software controlled via the set of registers shown in Figure 20-25. The possible input events are summarized again in Table 20-16.

Figure 20-25. Action-Qualifier Submodule Inputs and Outputs

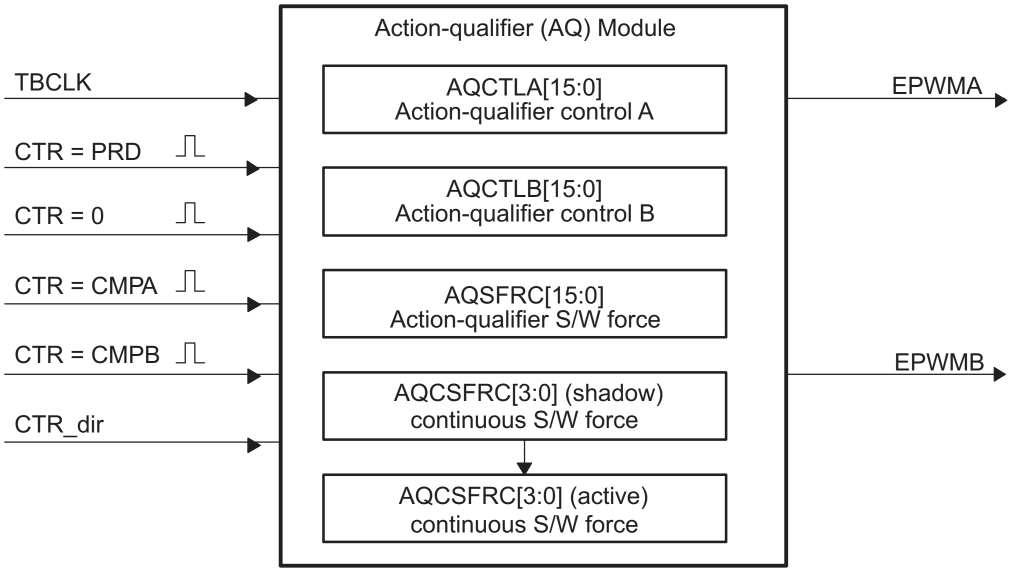


Table 20-16. Action-Qualifier Submodule Possible Input Events

Signal	Description	Registers Compared
CTR = PRD	Time-base counter equal to the period value	TBCNT = TBPRD
CTR = 0	Time-base counter equal to zero	TBCNT = 0000h
CTR = CMPA	Time-base counter equal to the counter-compare A	TBCNT = CMPA
CTR = CMPB	Time-base counter equal to the counter-compare B	TBCNT = CMPB
Software forced event	Asynchronous event initiated by software	

The software forced action is a useful asynchronous event. This control is handled by registers AQSFRC and AQCSFRC.

The action-qualifier submodule controls how the two outputs EPWMxA and EPWMxB behave when a particular event occurs. The event inputs to the action-qualifier submodule are further qualified by the counter direction (up or down). This allows for independent action on outputs on both the count-up and count-down phases.

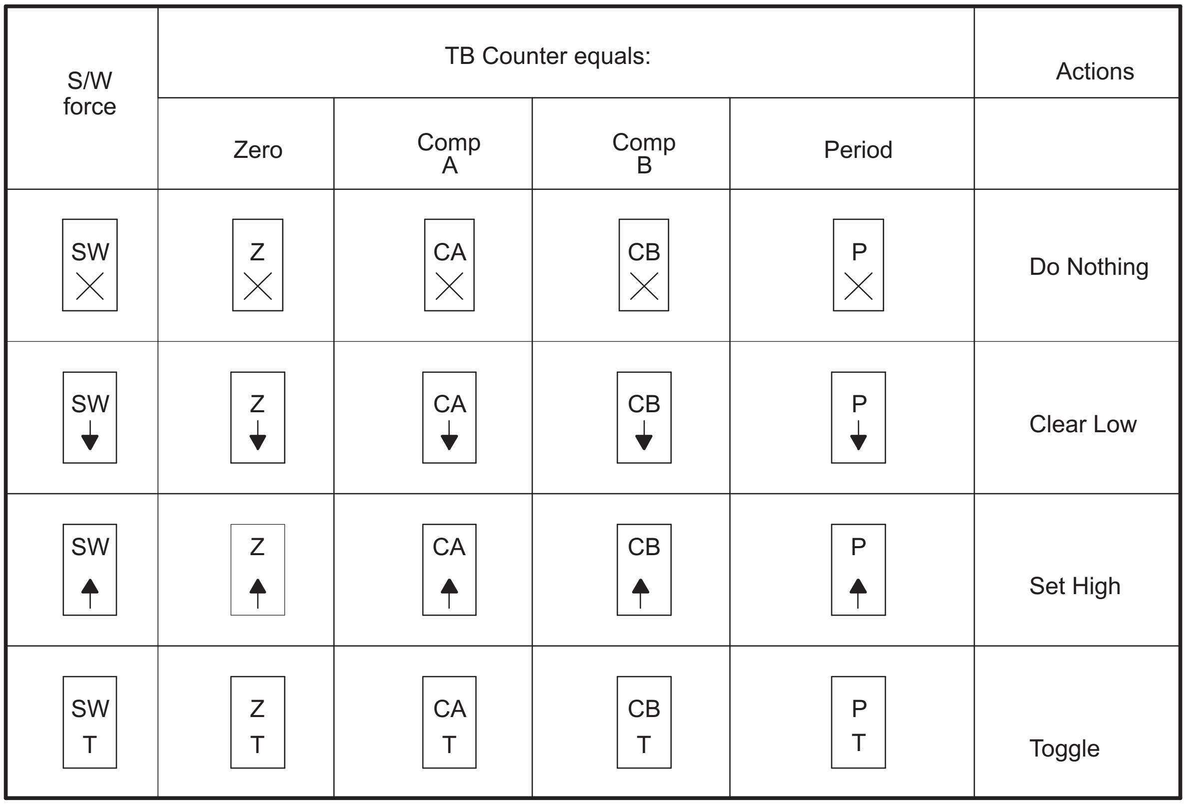
The possible actions imposed on outputs EPWMxA and EPWMxB are:

- **Set High:** Set output EPWMxA or EPWMxB to a high level.
- **Clear Low:** Set output EPWMxA or EPWMxB to a low level.
- **Toggle:** If EPWMxA or EPWMxB is currently pulled high, then pull the output low. If EPWMxA or EPWMxB is currently pulled low, then pull the output high.
- **Do Nothing:** Keep outputs EPWMxA and EPWMxB at same level as currently set. Although the "Do Nothing" option prevents an event from causing an action on the EPWMxA and EPWMxB outputs, this event can still trigger interrupts. See the event-trigger submodule description in Section 20.2.2.9 for details.

Actions are specified independently for either output (EPWMxA or EPWMxB). Any or all events can be configured to generate actions on a given output. For example, both CTR = CMPA and CTR = CMPB can operate on output EPWMxA. All qualifier actions are configured via the control registers found at the end of this section.

For clarity, the drawings in this chapter use a set of symbolic actions. These symbols are summarized in Figure 20-26. Each symbol represents an action as a marker in time. Some actions are fixed in time (zero and period) while the CMPA and CMPB actions are moveable and their time positions are programmed via the counter-compare A and B registers, respectively. To turn off or disable an action, use the "Do Nothing option"; it is the default at reset.

Figure 20-26. Possible Action-Qualifier Actions for EPWMxA and EPWMxB Outputs



S/W force	TB Counter equals:				Actions
	Zero	Comp A	Comp B	Period	
SW ×	Z ×	CA ×	CB ×	P ×	Do Nothing
SW ↓	Z ↓	CA ↓	CB ↓	P ↓	Clear Low
SW ↑	Z ↑	CA ↑	CB ↑	P ↑	Set High
SW T	Z T	CA T	CB T	P T	Toggle

20.2.2.5.3 Action-Qualifier Event Priority

It is possible for the ePWM action qualifier to receive more than one event at the same time. In this case events are assigned a priority by the hardware. The general rule is events occurring later in time have a higher priority and software forced events always have the highest priority. The event priority levels for up-down-count mode are shown in Table 20-17. A priority level of 1 is the highest priority and level 7 is the lowest. The priority changes slightly depending on the direction of TBCNT.

Table 20-17. Action-Qualifier Event Priority for Up-Down-Count Mode

Priority Level	Event if TBCNT is Incrementing TBCNT = 0 up to TBCNT = TBPRD	Event if TBCNT is Decrementing TBCNT = TBPRD down to TBCNT = 1
1 (Highest)	Software forced event	Software forced event
2	Counter equals CMPB on up-count (CBU)	Counter equals CMPB on down-count (CBD)
3	Counter equals CMPA on up-count (CAU)	Counter equals CMPA on down-count (CAD)
4	Counter equals zero	Counter equals period (TBPRD)
5	Counter equals CMPB on down-count (CBD) (1)	Counter equals CMPB on up-count (CBU) (1)
6 (Lowest)	Counter equals CMPA on down-count (CAD) (1)	Counter equals CMPA on up-count (CBU) (1)

(1) To maintain symmetry for up-down-count mode, both up-events (CAU/CBU) and down-events (CAD/CBD) can be generated for TBPRD. Otherwise, up-events can occur only when the counter is incrementing and down-events can occur only when the counter is decrementing.

Table 20-18 shows the action-qualifier priority for up-count mode. In this case, the counter direction is always defined as up and thus down-count events will never be taken.

Table 20-18. Action-Qualifier Event Priority for Up-Count Mode

Priority Level	Event
1 (Highest)	Software forced event
2	Counter equal to period (TBPRD)
3	Counter equal to CMPB on up-count (CBU)
4	Counter equal to CMPA on up-count (CAU)
5 (Lowest)	Counter equal to Zero

Table 20-19 shows the action-qualifier priority for down-count mode. In this case, the counter direction is always defined as down and thus up-count events will never be taken.

Table 20-19. Action-Qualifier Event Priority for Down-Count Mode

Priority Level	Event
1 (Highest)	Software forced event
2	Counter equal to Zero
3	Counter equal to CMPB on down-count (CBD)
4	Counter equal to CMPA on down-count (CAD)
5 (Lowest)	Counter equal to period (TBPRD)

It is possible to set the compare value greater than the period. In this case the action will take place as shown in Table 20-20.

Table 20-20. Behavior if CMPA/CMPB is Greater than the Period

Counter Mode	Compare on Up-Count Event CAU/CBU	Compare on Down-Count Event CAU/CBU
Up-Count Mode	If CMPA/CMPB ≤ TBPRD period, then the event occurs on a compare match (TBCNT = CMPA or CMPB). If CMPA/CMPB > TBPRD, then the event will not occur.	Never occurs.
Down-Count Mode	Never occurs.	If CMPA/CMPB < TBPRD, the event will occur on a compare match (TBCNT = CMPA or CMPB). If CMPA/CMPB ≥ TBPRD, the event will occur on a period match (TBCNT = TBPRD).
Up-Down-Count Mode	If CMPA/CMPB < TBPRD and the counter is incrementing, the event occurs on a compare match (TBCNT = CMPA or CMPB). If CMPA/CMPB is ≥ TBPRD, the event will occur on a period match (TBCNT = TBPRD).	If CMPA/CMPB < TBPRD and the counter is decrementing, the event occurs on a compare match (TBCNT = CMPA or CMPB). If CMPA/CMPB ≥ TBPRD, the event occurs on a period match (TBCNT = TBPRD).

20.2.2.5.4 Waveforms for Common Configurations

NOTE: The waveforms in this chapter show the ePWMs behavior for a static compare register value. In a running system, the active compare registers (CMPA and CMPB) are typically updated from their respective shadow registers once every period. The user specifies when the update will take place; either when the time-base counter reaches zero or when the time-base counter reaches period. There are some cases when the action based on the new value can be delayed by one period or the action based on the old value can take effect for an extra period. Some PWM configurations avoid this situation. These include, but are not limited to, the following:

Use up-down-count mode to generate a symmetric PWM:

- If you load CMPA/CMPB on zero, then use CMPA/CMPB values greater than or equal to 1.
- If you load CMPA/CMPB on period, then use CMPA/CMPB values less than or equal to TBPRD - 1. This means there will always be a pulse of at least one TBCLK cycle in a PWM period which, when very short, tend to be ignored by the system.

Use up-down-count mode to generate an asymmetric PWM:

- To achieve 50%-0% asymmetric PWM use the following configuration: Load CMPA/CMPB on period and use the period action to clear the PWM and a compare-up action to set the PWM. Modulate the compare value from 0 to TBPRD to achieve 50%-0% PWM duty.

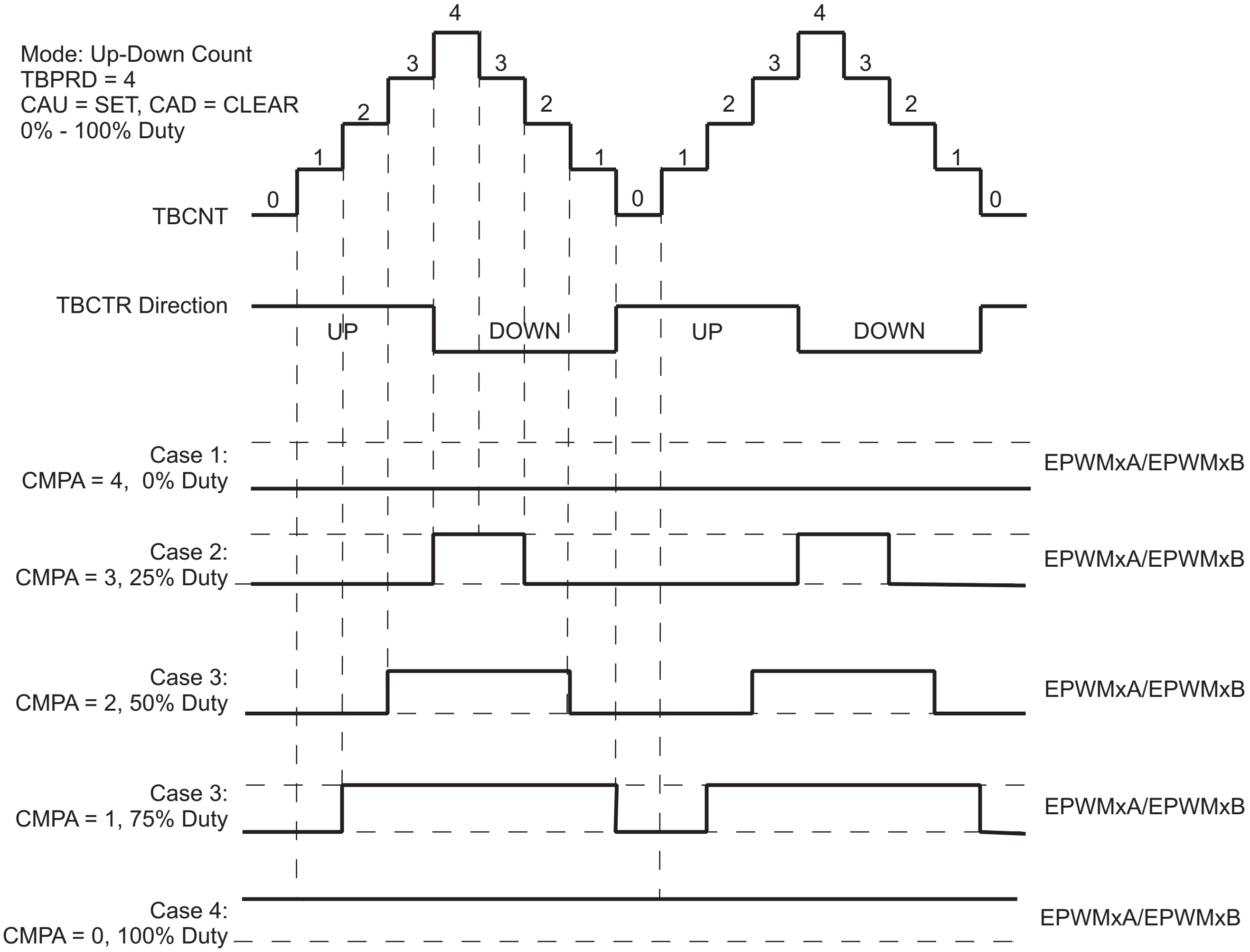
When using up-count mode to generate an asymmetric PWM:

- To achieve 0-100% asymmetric PWM use the following configuration: Load CMPA/CMPB on TBPRD. Use the Zero action to set the PWM and a compare-up action to clear the PWM. Modulate the compare value from 0 to TBPRD+1 to achieve 0-100% PWM duty.

Figure 20-27 shows how a symmetric PWM waveform can be generated using the up-down-count mode of the TBCNT. In this mode 0%-100% DC modulation is achieved by using equal compare matches on the up count and down count portions of the waveform. In the example shown, CMPA is used to make the comparison. When the counter is incrementing the CMPA match will pull the PWM output high. Likewise, when the counter is decrementing the compare match will pull the PWM signal low. When CMPA = 0, the PWM signal is low for the entire period giving the 0% duty waveform. When CMPA = TBPRD, the PWM signal is high achieving 100% duty.

When using this configuration in practice, if you load CMPA/CMPB on zero, then use CMPA/CMPB values greater than or equal to 1. If you load CMPA/CMPB on period, then use CMPA/CMPB values less than or equal to TBPRD-1. This means there will always be a pulse of at least one TBCLK cycle in a PWM period which, when very short, tend to be ignored by the system.

Figure 20-27. Up-Down-Count Mode Symmetrical Waveform

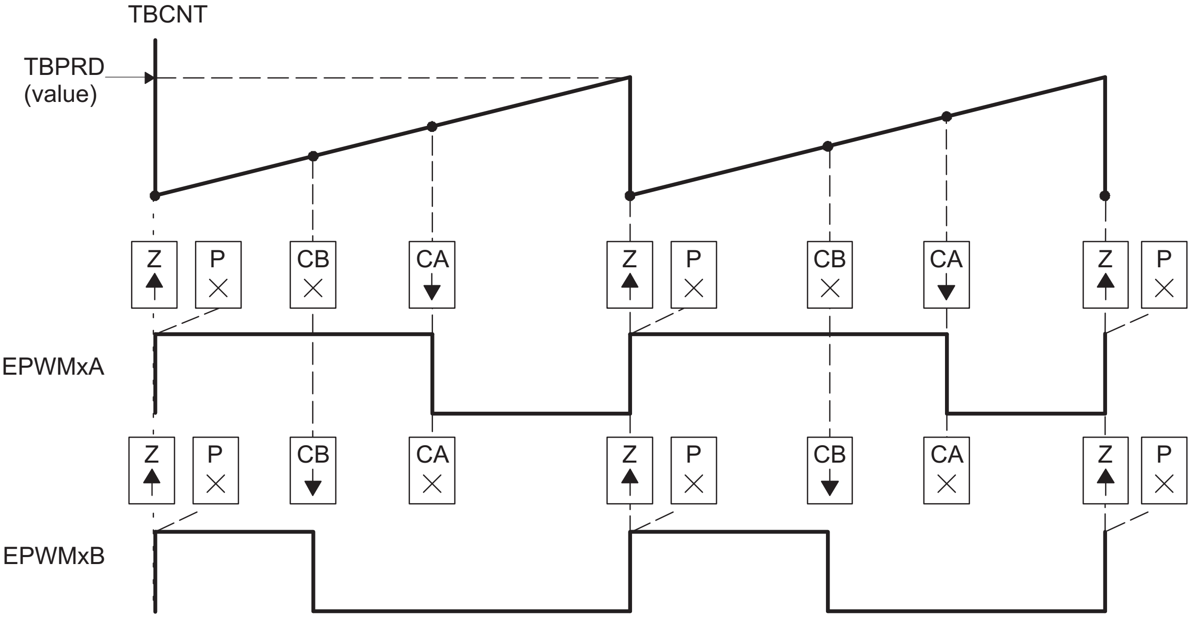


The PWM waveforms in Figure 20-28 through Figure 20-33 show some common action-qualifier configurations. Some conventions used in the figures are as follows:

- TBPRD, CMPA, and CMPB refer to the value written in their respective registers. The active register, not the shadow register, is used by the hardware.
- CMPx, refers to either CMPA or CMPB.
- EPWMxA and EPWMxB refer to the output signals from ePWMx
- Up-Down means Count-up-and-down mode, Up means up-count mode and Dwn means down-count mode
- Sym = Symmetric, Asym = Asymmetric

Table 20-21 and Table 20-22 contains initialization and runtime register configurations for the waveforms in Figure 20-28.

Figure 20-28. Up, Single Edge Asymmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB—Active High



(1) PWM period = (TBPRD + 1) × T_{TBCLK}

(2) Duty modulation for EPWMxA is set by CMPA, and is active high (that is, high time duty proportional to CMPA).

(3) Duty modulation for EPWMxB is set by CMPB and is active high (that is, high time duty proportional to CMPB).

(4) The "Do Nothing" actions (X) are shown for completeness, but will not be shown on subsequent diagrams.

(5) Actions at zero and period, although appearing to occur concurrently, are actually separated by one TBCLK period. TBCNT wraps from period to 0000h.

Table 20-21. EPWMx Initialization for Figure 20-28

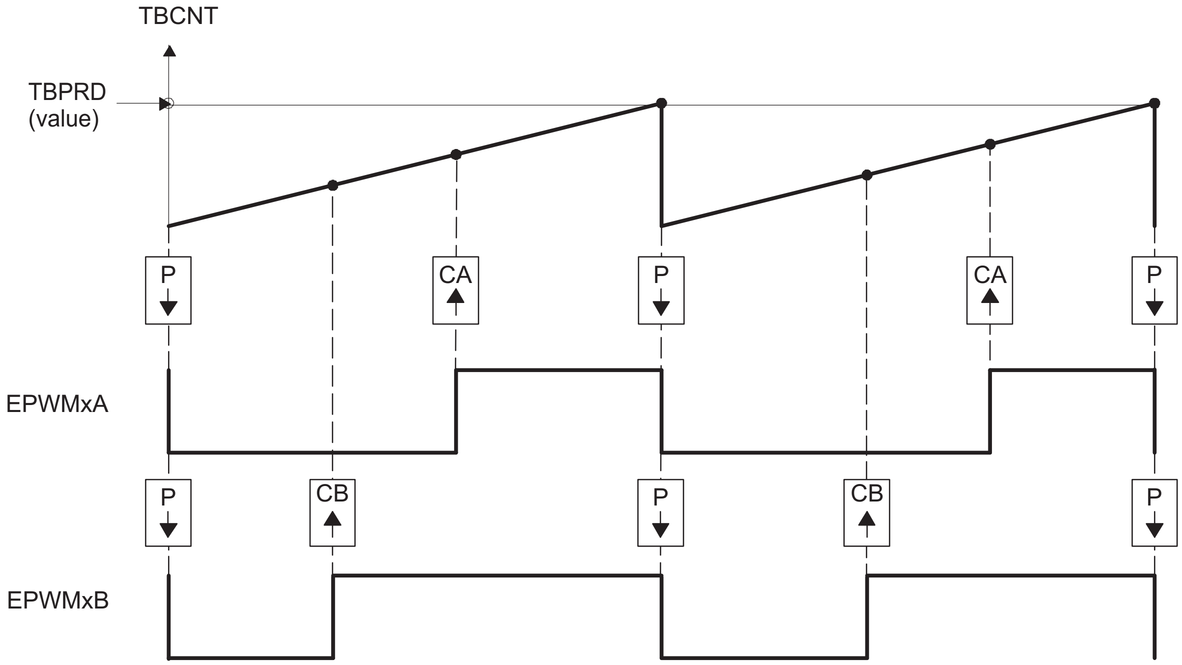
Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCNT	TBCNT	0	Clear TB counter
TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLK
	CLKDIV	TB_DIV1	
CMPA	CMPA	350 (15Eh)	Compare A = 350 TBCLK counts
CMPB	CMPB	200 (C8h)	Compare B = 200 TBCLK counts
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	ZRO	AQ_SET	
	CAU	AQ_CLEAR	
AQCTLB	ZRO	AQ_SET	
	CBU	AQ_CLEAR	

Table 20-22. EPWMx Run Time Changes for Figure 20-28

Register	Bit	Value	Comments
CMPA	CMPA	Duty1A	Adjust duty for output EPWM1A
CMPB	CMPB	Duty1B	Adjust duty for output EPWM1B

Table 20-23 and Table 20-24 contains initialization and runtime register configurations for the waveforms in Figure 20-29.

Figure 20-29. Up, Single Edge Asymmetric Waveform With Independent Modulation on EPWMxA and EPWMxB—Active Low



(1) PWM period = (TBPRD + 1) × T_{TBCLK}

(2) Duty modulation for EPWMxA is set by CMPA, and is active low (that is, the low time duty is proportional to CMPA).

(3) Duty modulation for EPWMxB is set by CMPB and is active low (that is, the low time duty is proportional to CMPB).

(4) The Do Nothing actions (X) are shown for completeness here, but will not be shown on subsequent diagrams.

(5) Actions at zero and period, although appearing to occur concurrently, are actually separated by one TBCLK period. TBCNT wraps from period to 0000h.

Table 20-23. EPWMx Initialization for Figure 20-29

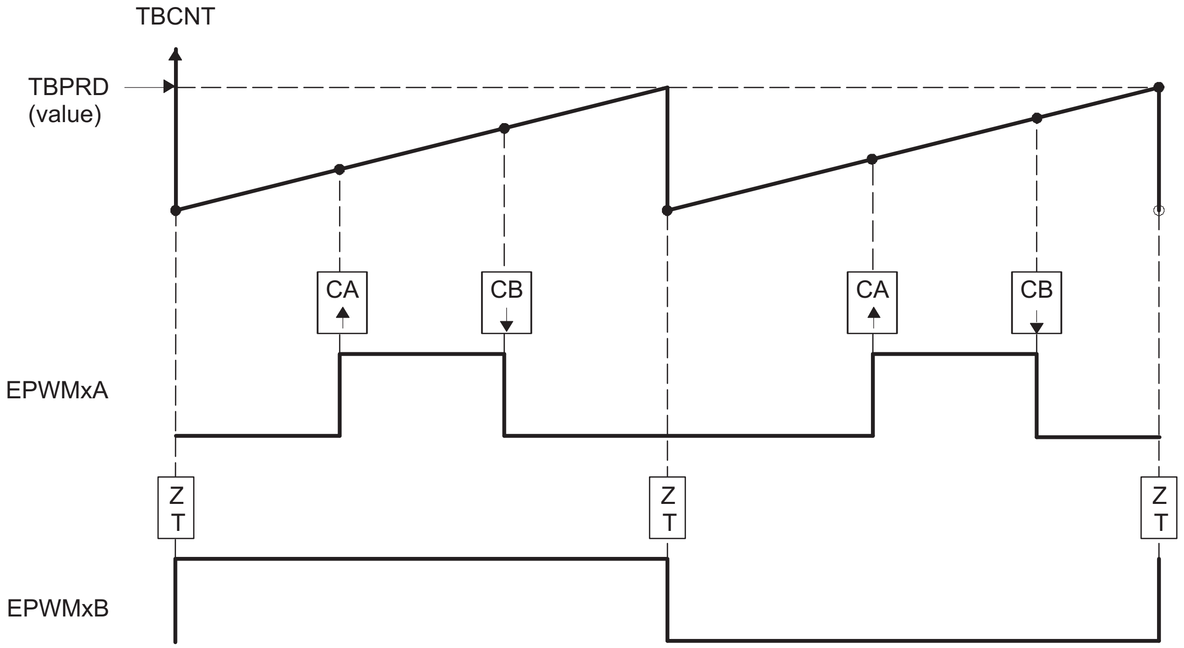
Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCNT	TBCNT	0	Clear TB counter
TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLK
	CLKDIV	TB_DIV1	
CMPA	CMPA	350 (15Eh)	Compare A = 350 TBCLK counts
CMPB	CMPB	200 (C8h)	Compare B = 200 TBCLK counts
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	PRD	AQ_CLEAR	
	CAU	AQ_SET	
AQCTLB	PRD	AQ_CLEAR	
	CBU	AQ_SET	

Table 20-24. EPWMx Run Time Changes for Figure 20-29

Register	Bit	Value	Comments
CMPA	CMPA	Duty1A	Adjust duty for output EPWM1A
CMPB	CMPB	Duty1B	Adjust duty for output EPWM1B

Table 20-25 and Table 20-26 contains initialization and runtime register configurations for the waveforms Figure 20-30. Use the code in Example 20-1 to define the headers.

Figure 20-30. Up-Count, Pulse Placement Asymmetric Waveform With Independent Modulation on EPWMxA



(1) PWM frequency = 1/((TBPRD + 1) × T_{TBCLK})

(2) Pulse can be placed anywhere within the PWM cycle (0000h - TBPRD)

(3) High time duty proportional to (CMPB - CMPA)

(4) EPWMxB can be used to generate a 50% duty square wave with frequency = 1/2 × ((TBPRD + 1) × TBCLK)

Table 20-25. EPWMx Initialization for Figure 20-30

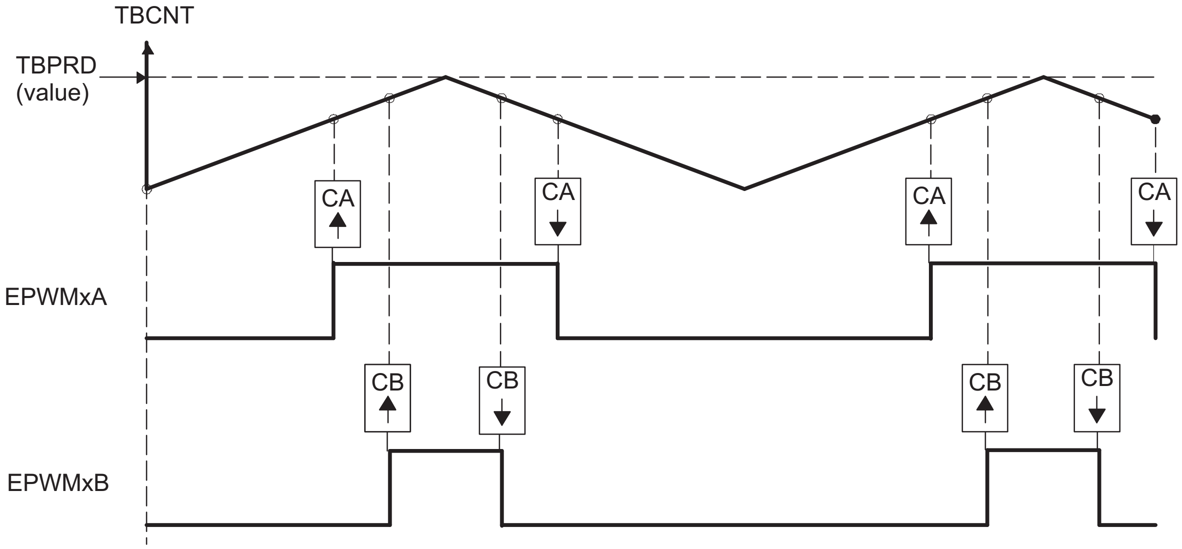
Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCNT	TBCNT	0	Clear TB counter
TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLK
	CLKDIV	TB_DIV1	
CMPA	CMPA	200 (C8h)	Compare A = 200 TBCLK counts
CMPB	CMPB	400 (190h)	Compare B = 400 TBCLK counts
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	
	CBU	AQ_CLEAR	
AQCTLB	ZRO	AQ_TOGGLE	

Table 20-26. EPWMx Run Time Changes for Figure 20-30

Register	Bit	Value	Comments
CMPA	CMPA	EdgePosA	Adjust duty for output EPWM1A
CMPB	CMPB	EdgePosB	Adjust duty for output EPWM1B

Table 20-27 and Table 20-28 contains initialization and runtime register configurations for the waveforms in Figure 20-31. Use the code in Example 20-1 to define the headers.

Figure 20-31. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB — Active Low



(1) PWM period = 2 x TBPRD × T_{TBCLK}

(2) Duty modulation for EPWMxA is set by CMPA, and is active low (that is, the low time duty is proportional to CMPA).

(3) Duty modulation for EPWMxB is set by CMPB and is active low (that is, the low time duty is proportional to CMPB).

(4) Outputs EPWMxA and EPWMxB can drive independent power switches

Table 20-27. EPWMx Initialization for Figure 20-31

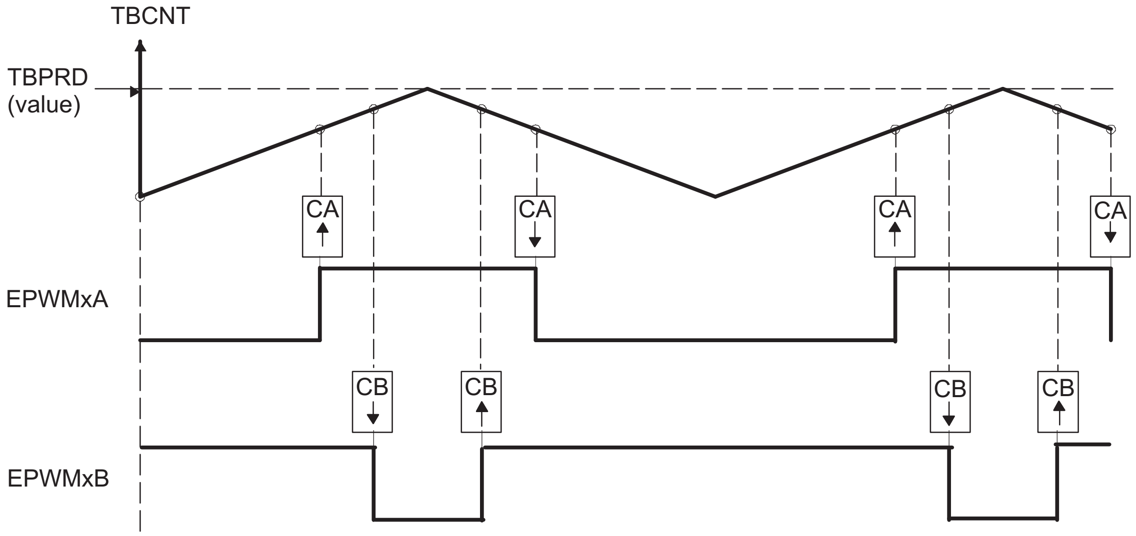
Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCNT	TBCNT	0	Clear TB counter
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLK
	CLKDIV	TB_DIV1	
CMPA	CMPA	400 (190h)	Compare A = 400 TBCLK counts
CMPB	CMPB	500 (1F4h)	Compare B = 500 TBCLK counts
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	
	CAD	AQ_CLEAR	
AQCTLB	CBU	AQ_SET	
	CBD	AQ_CLEAR	

Table 20-28. EPWMx Run Time Changes for Figure 20-31

Register	Bit	Value	Comments
CMPA	CMPA	Duty1A	Adjust duty for output EPWM1A
CMPB	CMPB	Duty1B	Adjust duty for output EPWM1B

Table 20-29 and Table 20-30 contains initialization and runtime register configurations for the waveforms in Figure 20-32. Use the code in Example 20-1 to define the headers.

Figure 20-32. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB — Complementary



(1) PWM period = 2 × TBPRD × T_{TBCLK}

(2) Duty modulation for EPWMxA is set by CMPA, and is active low, i.e., low time duty proportional to CMPA

(3) Duty modulation for EPWMxB is set by CMPB and is active high, i.e., high time duty proportional to CMPB

(4) Outputs EPWMx can drive upper/lower (complementary) power switches

(5) Dead-band = CMPB - CMPA (fully programmable edge placement by software). Note the dead-band module is also available if the more classical edge delay method is required.

Table 20-29. EPWMx Initialization for Figure 20-32

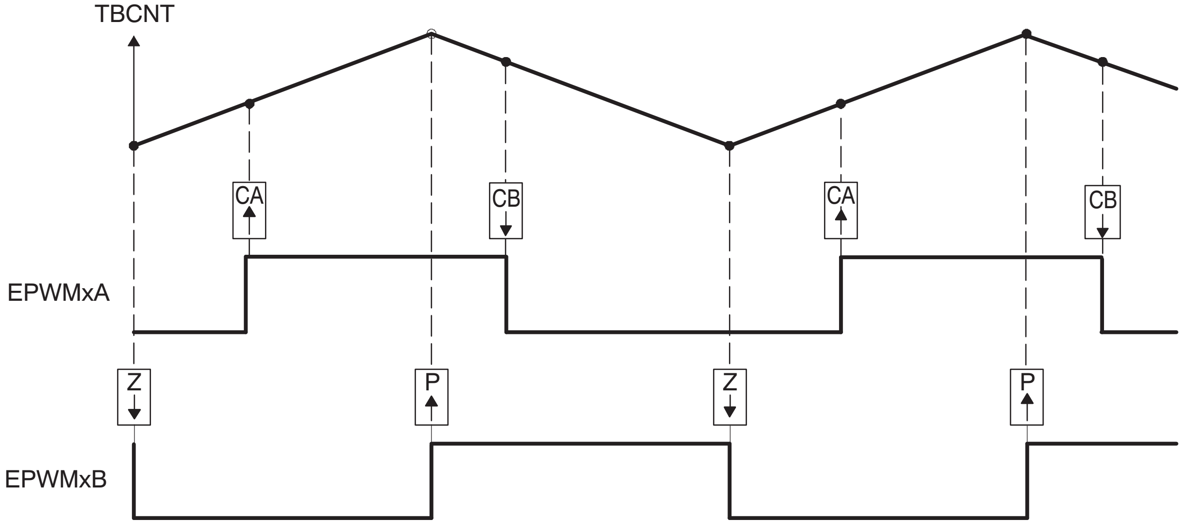
Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCNT	TBCNT	0	Clear TB counter
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLK
	CLKDIV	TB_DIV1	
CMPA	CMPA	350 (15Eh)	Compare A = 350 TBCLK counts
CMPB	CMPB	400 (190h)	Compare B = 400 TBCLK counts
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	
	CAD	AQ_CLEAR	
AQCTLB	CBU	AQ_CLEAR	
	CBD	AQ_SET	

Table 20-30. EPWMx Run Time Changes for Figure 20-32

Register	Bit	Value	Comments
CMPA	CMPA	Duty1A	Adjust duty for output EPWM1A
CMPB	CMPB	Duty1B	Adjust duty for output EPWM1B

Table 20-31 and Table 20-32 contains initialization and runtime register configurations for the waveforms in Figure 20-33. Use the code in Example 20-1 to define the headers.

Figure 20-33. Up-Down-Count, Dual Edge Asymmetric Waveform, With Independent Modulation on EPWMxA—Active Low



(1) PWM period = 2 × TBPRD × TBCLK

(2) Rising edge and falling edge can be asymmetrically positioned within a PWM cycle. This allows for pulse placement techniques.

(3) Duty modulation for EPWMxA is set by CMPA and CMPB.

(4) Low time duty for EPWMxA is proportional to (CMPA + CMPB).

(5) To change this example to active high, CMPA and CMPB actions need to be inverted (i.e., Set ! Clear and Clear Set).

(6) Duty modulation for EPWMxB is fixed at 50% (utilizes spare action resources for EPWMxB)

Table 20-31. EPWMx Initialization for Figure 20-33

Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCNT	TBCNT	0	Clear TB counter
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLK
	CLKDIV	TB_DIV1	
CMPA	CMPA	250 (FAh)	Compare A = 250 TBCLK counts
CMPB	CMPB	450 (1C2h)	Compare B = 450 TBCLK counts
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	
	CBD	AQ_CLEAR	
AQCTLB	ZRO	AQ_CLEAR	
	PRD	AQ_SET	

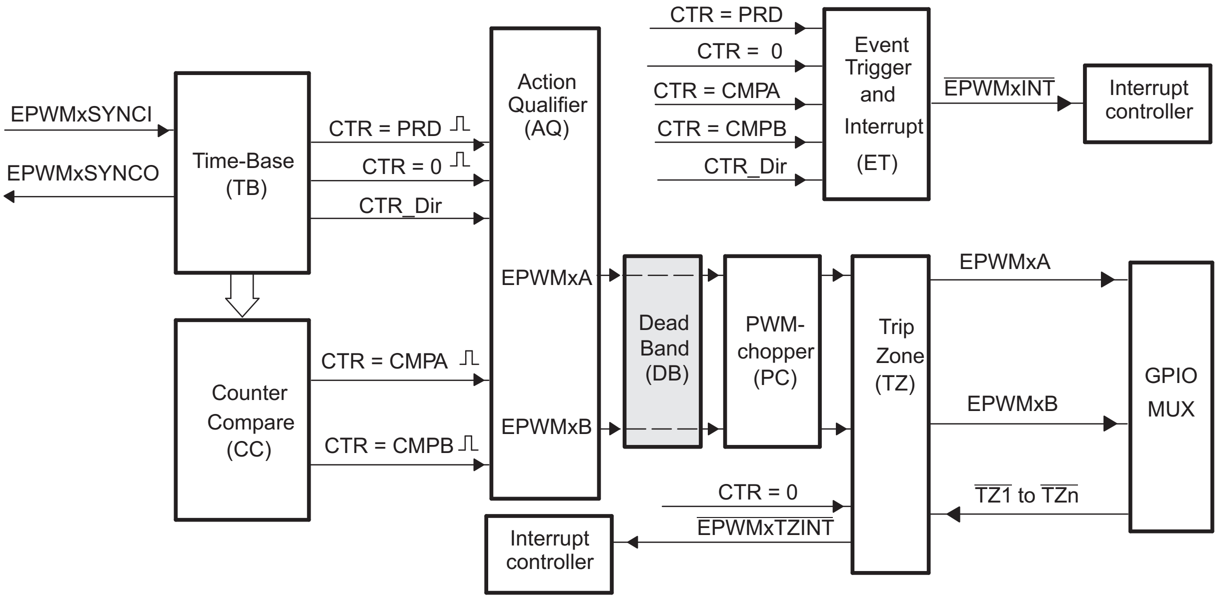
Table 20-32. EPWMx Run Time Changes for Figure 20-33

Register	Bit	Value	Comments
CMPA	CMPA	EdgePosA	Adjust duty for output EPWM1A
CMPB	CMPB	EdgePosB	

20.2.2.6 Dead-Band Generator (DB) Submodule

Figure 20-34 illustrates the dead-band generator submodule within the ePWM module.

Figure 20-34. Dead-Band Generator Submodule



20.2.2.6.1 Purpose of the Dead-Band Submodule

The "Action-qualifier (AQ) Module" section discussed how it is possible to generate the required dead-band by having full control over edge placement using both the CMPA and CMPB resources of the ePWM module. However, if the more classical edge delay-based dead-band with polarity control is required, then the dead-band generator submodule should be used.

The key functions of the dead-band generator submodule are:

- Generating appropriate signal pairs (EPWMxA and EPWMxB) with dead-band relationship from a single EPWMxA input
- Programming signal pairs for:
  - Active high (AH)
  - Active low (AL)
  - Active high complementary (AHC)
  - Active low complementary (ALC)
- Adding programmable delay to rising edges (RED)
- Adding programmable delay to falling edges (FED)
- Can be totally bypassed from the signal path (note dotted lines in diagram)

20.2.2.6.2 Controlling and Monitoring the Dead-Band Submodule

The dead-band generator submodule operation is controlled and monitored via the following registers:

Table 20-33. Dead-Band Generator Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
DBCTL	Dead-Band Control Register	1Eh	No
DBRED	Dead-Band Rising Edge Delay Count Register	20h	No
DBFED	Dead-Band Falling Edge Delay Count Register	22h	No

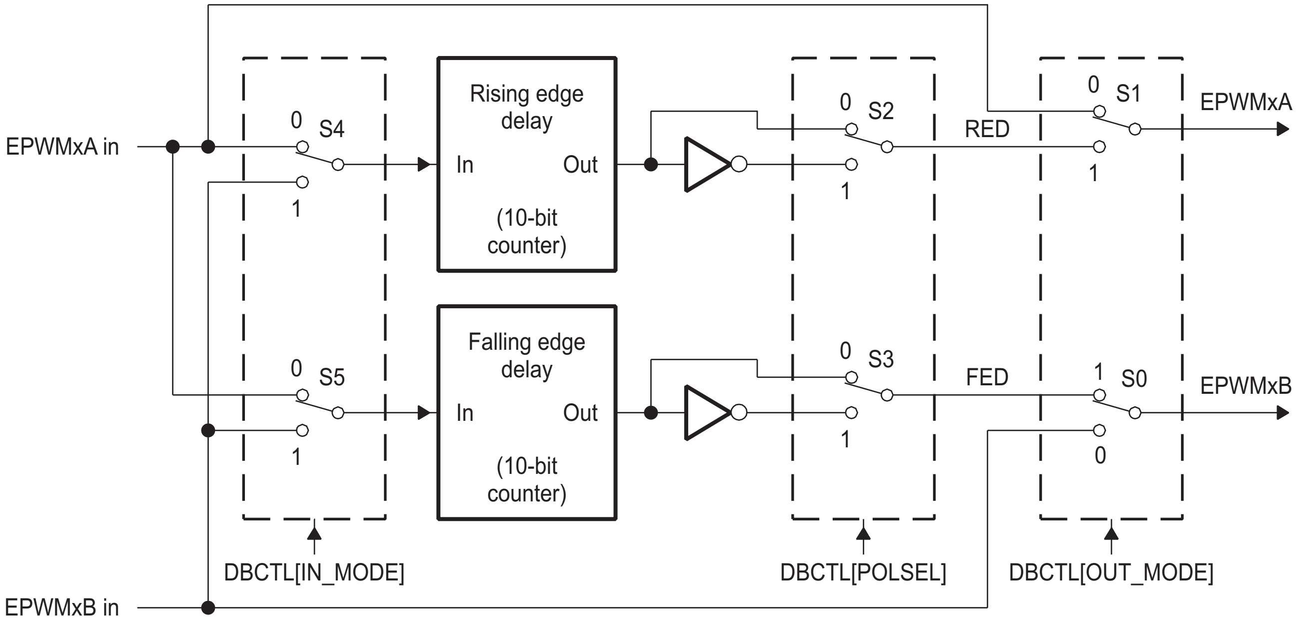
20.2.2.6.3 Operational Highlights for the Dead-Band Generator Submodule

The following sections provide the operational highlights.

The dead-band submodule has two groups of independent selection options as shown in Figure 20-35.

- **Input Source Selection:** The input signals to the dead-band module are the EPWMxA and EPWMxB output signals from the action-qualifier. In this section they will be referred to as EPWMxA In and EPWMxB In. Using the DBCTL[IN_MODE) control bits, the signal source for each delay, falling-edge or rising-edge, can be selected:
  - EPWMxA In is the source for both falling-edge and rising-edge delay. This is the default mode.
  - EPWMxA In is the source for falling-edge delay, EPWMxB In is the source for rising-edge delay.
  - EPWMxA In is the source for rising edge delay, EPWMxB In is the source for falling-edge delay.
  - EPWMxB In is the source for both falling-edge and rising-edge delay.
- **Output Mode Control:** The output mode is configured by way of the DBCTL[OUT_MODE] bits. These bits determine if the falling-edge delay, rising-edge delay, neither, or both are applied to the input signals.
- **Polarity Control:** The polarity control (DBCTL[POLSEL]) allows you to specify whether the rising-edge delayed signal and/or the falling-edge delayed signal is to be inverted before being sent out of the dead-band submodule.

Figure 20-35. Configuration Options for the Dead-Band Generator Submodule



Although all combinations are supported, not all are typical usage modes. Table 20-34 lists some classical dead-band configurations. These modes assume that the DBCTL[IN_MODE] is configured such that EPWMxA In is the source for both falling-edge and rising-edge delay. Enhanced, or non-traditional modes can be achieved by changing the input signal source. The modes shown in Table 20-34 fall into the following categories:

- **Mode 1: Bypass both falling-edge delay (FED) and rising-edge delay (RED)** Allows you to fully disable the dead-band submodule from the PWM signal path.
- **Mode 2-5: Classical Dead-Band Polarity Settings** These represent typical polarity configurations that should address all the active high/low modes required by available industry power switch gate drivers. The waveforms for these typical cases are shown in Figure 20-36. Note that to generate equivalent waveforms to Figure 20-36, configure the action-qualifier submodule to generate the signal as shown for EPWMxA.
- **Mode 6: Bypass rising-edge-delay and Mode 7: Bypass falling-edge-delay** Finally the last two entries in Table 20-34 show combinations where either the falling-edge-delay (FED) or rising-edge-delay (RED) blocks are bypassed.

Table 20-34. Classical Dead-Band Operating Modes

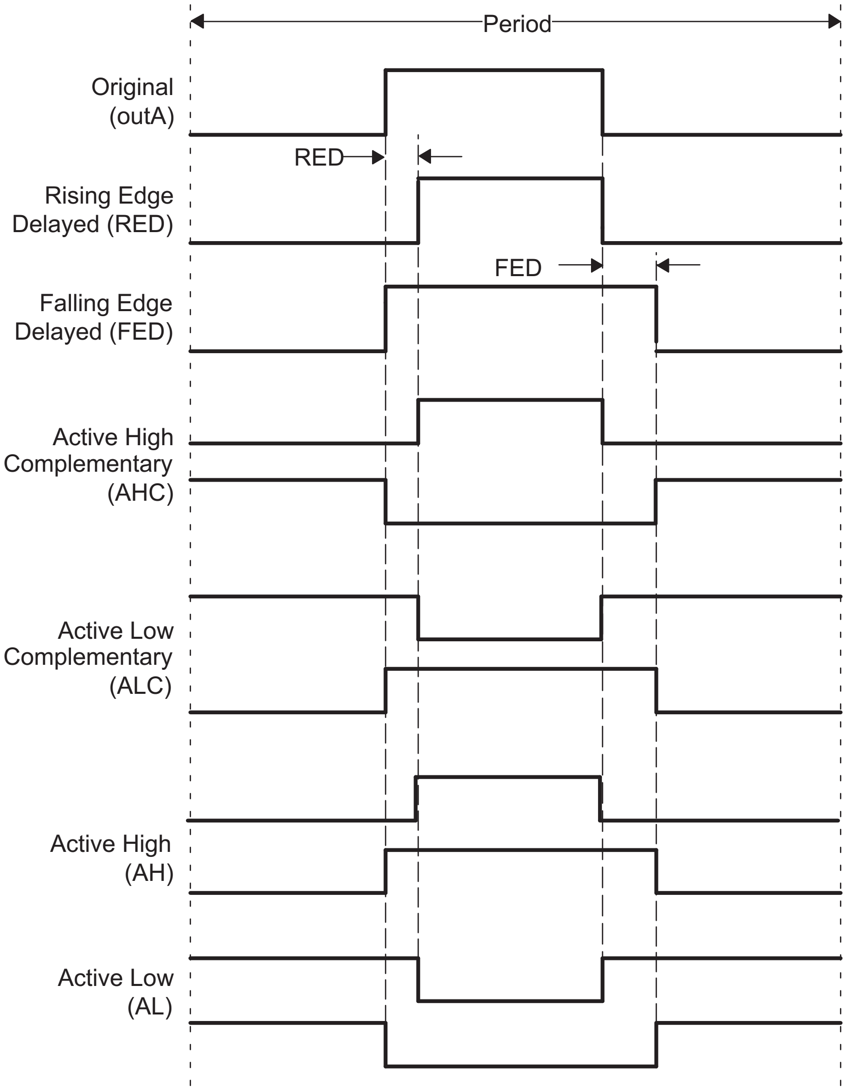
		DBCTL[POLSEL]		DBCTL[OUT_MODE]	
Mode	**Mode Description** [1]	**S3**	**S2**	**S1**	**S0**
1	EPWMxA and EPWMxB Passed Through (No Delay)	x	x	0	0
2	Active High Complementary (AHC)	1	0	1	1
3	Active Low Complementary (ALC)	0	1	1	1
4	Active High (AH)	0	0	1	1
5	Active Low (AL)	1	1	1	1
6	EPWMxA Out = EPWMxA In (No Delay) EPWMxB Out = EPWMxA In with Falling Edge Delay	0 or 1	0 or 1	0	1
7	EPWMxA Out = EPWMxA In with Rising Edge Delay EPWMxB Out = EPWMxB In with No Delay	0 or 1	0 or 1	1	0

---

[1] These are classical dead-band modes and assume that DBCTL[IN_MODE] = 0,0. That is, EPWMxA in is the source for both the falling-edge and rising-edge delays. Enhanced, non-traditional modes can be achieved by changing the IN_MODE configuration.

Figure 20-36 shows waveforms for typical cases where 0% < duty < 100%.

Figure 20-36. Dead-Band Waveforms for Typical Cases (0% < Duty < 100%)



The dead-band submodule supports independent values for rising-edge (RED) and falling-edge (FED) delays. The amount of delay is programmed using the DBRED and DBFED registers. These are 10-bit registers and their value represents the number of time-base clock, TBCLK, periods a signal edge is delayed by. For example, the formula to calculate falling-edge-delay and rising-edge-delay are:

$FED = DBFED \times T_{TBCLK}$

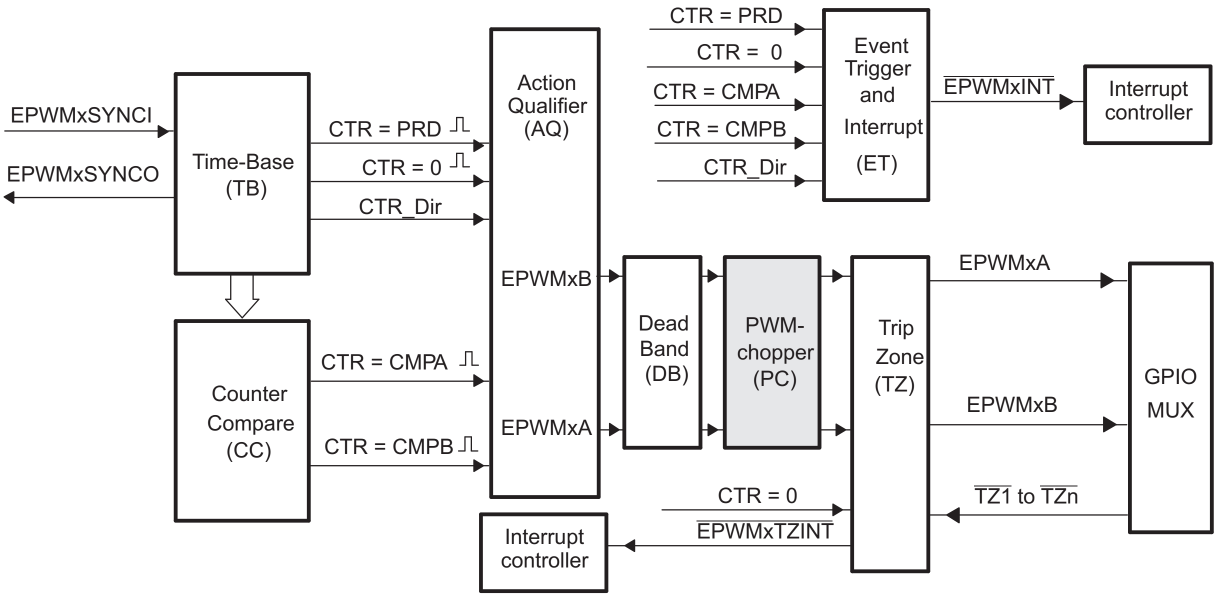
$RED = DBRED \times T_{TBCLK}$

Where T_{TBCLK} is the period of TBCLK, the prescaled version of SYSCLKOUT.

20.2.2.7 PWM-Chopper (PC) Submodule

Figure 20-37 illustrates the PWM-chopper (PC) submodule within the ePWM module. The PWM-chopper submodule allows a high-frequency carrier signal to modulate the PWM waveform generated by the action-qualifier and dead-band submodules. This capability is important if you need pulse transformer-based gate drivers to control the power switching elements.

Figure 20-37. PWM-Chopper Submodule



20.2.2.7.1 *Purpose of the PWM-Chopper Submodule*

The key functions of the PWM-chopper submodule are:

- Programmable chopping (carrier) frequency
- Programmable pulse width of first pulse
- Programmable duty cycle of second and subsequent pulses
- Can be fully bypassed if not required

20.2.2.7.2 *Controlling the PWM-Chopper Submodule*

The PWM-chopper submodule operation is controlled via the register in Table 20-35.

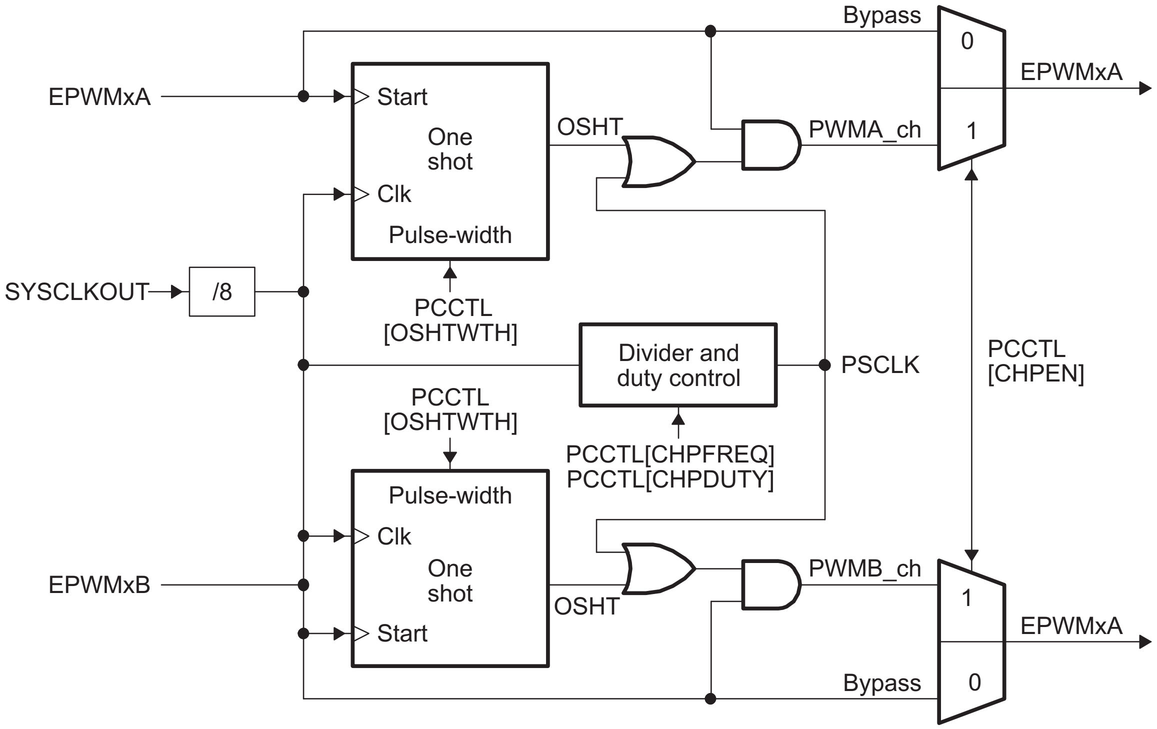
Table 20-35. PWM-Chopper Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
PCCTL	PWM-chopper Control Register	3Ch	No

20.2.2.7.3 Operational Highlights for the PWM-Chopper Submodule

Figure 20-38 shows the operational details of the PWM-chopper submodule. The carrier clock is derived from SYSCLKOUT. Its frequency and duty cycle are controlled via the CHPFREQ and CHPDUTY bits in the PCCTL register. The one-shot block is a feature that provides a high energy first pulse to ensure hard and fast power switch turn on, while the subsequent pulses sustain pulses, ensuring the power switch remains on. The one-shot width is programmed via the OSHTWTH bits. The PWM-chopper submodule can be fully disabled (bypassed) via the CHPEN bit.

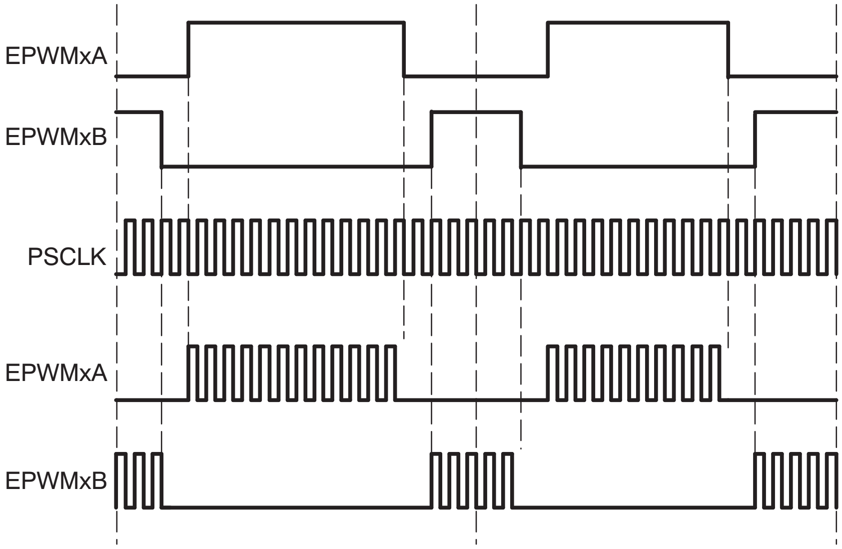
Figure 20-38. PWM-Chopper Submodule Signals and Registers



20.2.2.7.4 Waveforms

Figure 20-39 shows simplified waveforms of the chopping action only; one-shot and duty-cycle control are not shown. Details of the one-shot and duty-cycle control are discussed in the following sections.

Figure 20-39. Simple PWM-Chopper Submodule Waveforms Showing Chopping Action Only



20.2.2.7.4.1 One-Shot Pulse

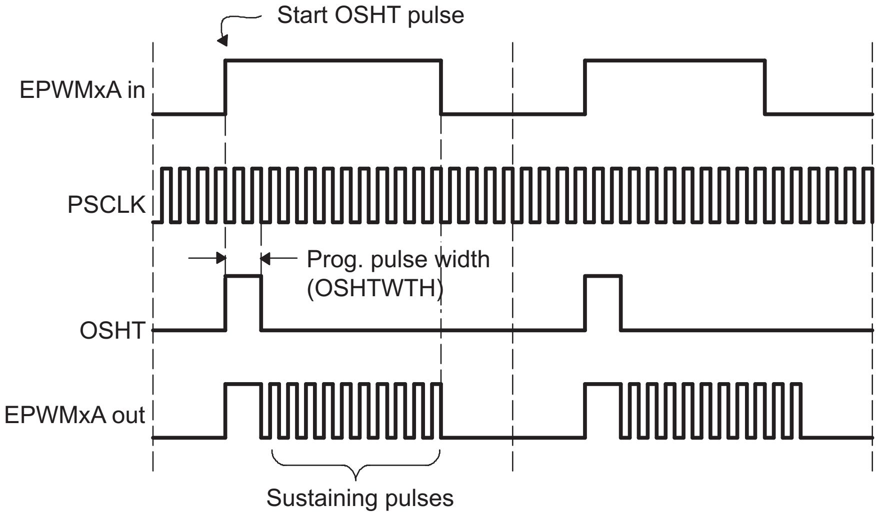
The width of the first pulse can be programmed to any of 16 possible pulse width values. The width or period of the first pulse is given by:

$$T_{1stpulse} = T_{SYSCLKOUT} \times 8 \times OSHTWTH$$

Where $T_{SYSCLKOUT}$ is the period of the system clock (SYSCLKOUT) and OSHTWTH is the four control bits (value from 1 to 16)

Figure 20-40 shows the first and subsequent sustaining pulses.

Figure 20-40. PWM-Chopper Submodule Waveforms Showing the First Pulse and Subsequent Sustaining Pulses

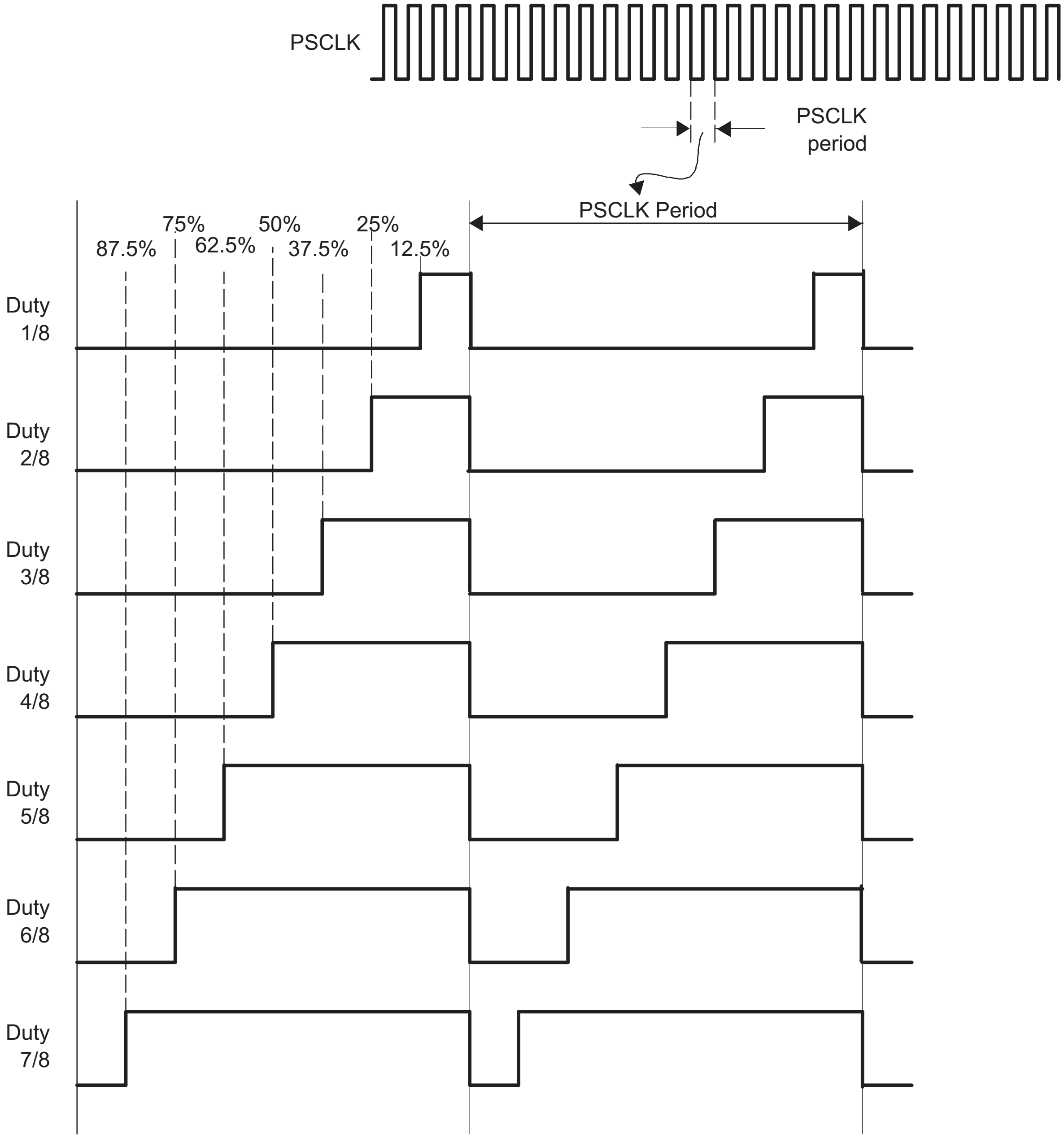


20.2.2.7.4.2 Duty Cycle Control

Pulse transformer-based gate drive designs need to comprehend the magnetic properties or characteristics of the transformer and associated circuitry. Saturation is one such consideration. To assist the gate drive designer, the duty cycles of the second and subsequent pulses have been made programmable. These sustaining pulses ensure the correct drive strength and polarity is maintained on the power switch gate during the on period, and hence a programmable duty cycle allows a design to be tuned or optimized via software control.

Figure 20-41 shows the duty cycle control that is possible by programming the CHPDUTY bits. One of seven possible duty ratios can be selected ranging from 12.5% to 87.5%.

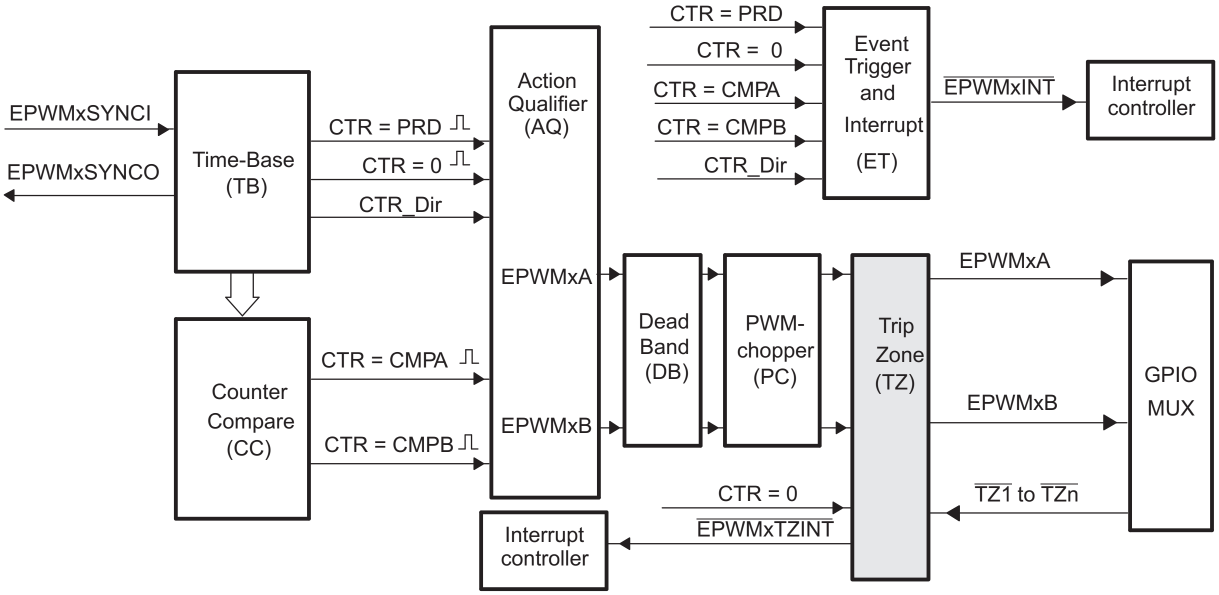
Figure 20-41. PWM-Chopper Submodule Waveforms Showing the Pulse Width (Duty Cycle) Control of Sustaining Pulses



20.2.2.8 Trip-Zone (TZ) Submodule

Figure 20-42 shows how the trip-zone (TZ) submodule fits within the ePWM module. Each ePWM module is connected to every $\overline{\text{TZ}}$ signal that are sourced from the GPIO MUX. These signals indicates external fault or trip conditions, and the ePWM outputs can be programmed to respond accordingly when faults occur. See Section 20.1.2 to determine the number of trip-zone pins available for the device.

Figure 20-42. Trip-Zone Submodule



20.2.2.8.1 *Purpose of the Trip-Zone Submodule*

The key functions of the trip-zone submodule are:

- Trip inputs $\overline{\text{TZ1}}$ to $\overline{\text{TZn}}$ can be flexibly mapped to any ePWM module.
- Upon a fault condition, outputs EPWMxA and EPWMxB can be forced to one of the following:
  - High
  - Low
  - High-impedance
  - No action taken
- Support for one-shot trip (OSHT) for major short circuits or over-current conditions.
- Support for cycle-by-cycle tripping (CBC) for current limiting operation.
- Each trip-zone input pin can be allocated to either one-shot or cycle-by-cycle operation.
- Interrupt generation is possible on any trip-zone pin.
- Software-forced tripping is also supported.
- The trip-zone submodule can be fully bypassed if it is not required.

20.2.2.8.2 Controlling and Monitoring the Trip-Zone Submodule

The trip-zone submodule operation is controlled and monitored through the following registers:

Table 20-36. Trip-Zone Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
TZSEL	Trip-Zone Select Register	24h	No
TZCTL	Trip-Zone Control Register	28h	No
TZEINT	Trip-Zone Enable Interrupt Register	2Ah	No
TZFLG	Trip-Zone Flag Register	2Ch	No
TZCLR	Trip-Zone Clear Register	2Eh	No
TZFRC	Trip-Zone Force Register	30h	No

20.2.2.8.3 Operational Highlights for the Trip-Zone Submodule

The following sections describe the operational highlights and configuration options for the trip-zone submodule.

The trip-zone signals at pin $\overline{\text{TZ1}}$ to $\overline{\text{TZn}}$ is an active-low input signal. When the pin goes low, it indicates that a trip event has occurred. Each ePWM module can be individually configured to ignore or use each of the trip-zone pins. Which trip-zone pins are used by a particular ePWM module is determined by the TZSEL register for that specific ePWM module. The trip-zone signal may or may not be synchronized to the system clock (SYSCLKOUT). A minimum of 1 SYSCLKOUT low pulse on the $\overline{\text{TZ}}$ $\overline{\text{n}}$ inputs is sufficient to trigger a fault condition in the ePWM module. The asynchronous trip makes sure that if clocks are missing for any reason, the outputs can still be tripped by a valid event present on the $\overline{\text{TZn}}$ inputs.

The $\overline{\text{TZ}}$ $\overline{\text{n}}$ input can be individually configured to provide either a cycle-by-cycle or one-shot trip event for a ePWM module. The configuration is determined by the TZSEL[CBCn] and TZSEL[OSHTn] bits (where n corresponds to the trip pin) respectively.

- **Cycle-by-Cycle (CBC):** When a cycle-by-cycle trip event occurs, the action specified in the TZCTL register is carried out immediately on the EPWMxA and/or EPWMxB output. Table 20-37 lists the possible actions. In addition, the cycle-by-cycle trip event flag (TZFLG[CBC]) is set and a EPWMxTZINT interrupt is generated if it is enabled in the TZEINT register.

  The specified condition on the pins is automatically cleared when the ePWM time-base counter reaches zero (TBCNT = 0000h) if the trip event is no longer present. Therefore, in this mode, the trip event is cleared or reset every PWM cycle. The TZFLG[CBC] flag bit will remain set until it is manually cleared by writing to the TZCLR[CBC] bit. If the cycle-by-cycle trip event is still present when the TZFLG[CBC] bit is cleared, then it will again be immediately set.
- **One-Shot (OSHT):** When a one-shot trip event occurs, the action specified in the TZCTL register is carried out immediately on the EPWMxA and/or EPWMxB output. Table 20-37 lists the possible actions. In addition, the one-shot trip event flag (TZFLG[OST]) is set and a EPWMxTZINT interrupt is generated if it is enabled in the TZEINT register. The one-shot trip condition must be cleared manually by writing to the TZCLR[OST] bit.

The action taken when a trip event occurs can be configured individually for each of the ePWM output pins by way of the TZCTL[TZA] and TZCTL[TZB] register bits. One of four possible actions, shown in Table 20-37, can be taken on a trip event.

Table 20-37. Possible Actions On a Trip Event

TZCTL[TZA] and/or TZCTL[TZB]	EPWMxA and/or EPWMxB	Comment
0	High-Impedance	Tripped
1h	Force to High State	Tripped
2h	Force to Low State	Tripped
3h	No Change	Do Nothing. No change is made to the output.

Example 20-2. Trip-Zone Configurations

Scenario A:
A one-shot trip event on $\overline{\text{TZ1}}$ pulls both EPWM1A, EPWM1B low and also forces EPWM2A and EPWM2B high.

- Configure the ePWM1 registers as follows:
  - TZSEL[OSHT1] = 1: enables $\overline{\text{TZ}}$ as a one-shot event source for ePWM1
  - TZCTL[TZA] = 2: EPWM1A will be forced low on a trip event.
  - TZCTL[TZB] = 2: EPWM1B will be forced low on a trip event.
- Configure the ePWM2 registers as follows:
  - TZSEL[OSHT1] = 1: enables $\overline{\text{TZ}}$ as a one-shot event source for ePWM2
  - TZCTL[TZA] = 1: EPWM2A will be forced high on a trip event.
  - TZCTL[TZB] = 1: EPWM2B will be forced high on a trip event.

Scenario B:
A cycle-by-cycle event on $\overline{\text{TZ5}}$ pulls both EPWM1A, EPWM1B low.

A one-shot event on $\overline{\text{TZ1}}$ or $\overline{\text{TZ6}}$ puts EPWM2A into a high impedance state.

- Configure the ePWM1 registers as follows:
  - TZSEL[CBC5] = 1: enables $\overline{\text{TZ5}}$ as a one-shot event source for ePWM1
  - TZCTL[TZA] = 2: EPWM1A will be forced low on a trip event.
  - TZCTL[TZB] = 2: EPWM1B will be forced low on a trip event.
- Configure the ePWM2 registers as follows:
  - TZSEL[OSHT1] = 1: enables $\overline{\text{TZ1}}$ as a one-shot event source for ePWM2
  - TZSEL[OSHT6] = 1: enables $\overline{\text{TZ6}}$ as a one-shot event source for ePWM1
  - TZCTL[TZA] = 0: EPWM1A will be put into a high-impedance state on a trip event.
  - TZCTL[TZB] = 3: EPWM1B will ignore the trip event.

20.2.2.8.4 Generating Trip Event Interrupts

Figure 20-43 and Figure 20-44 illustrate the trip-zone submodule control and interrupt logic, respectively.

Figure 20-43. Trip-Zone Submodule Mode Control Logic

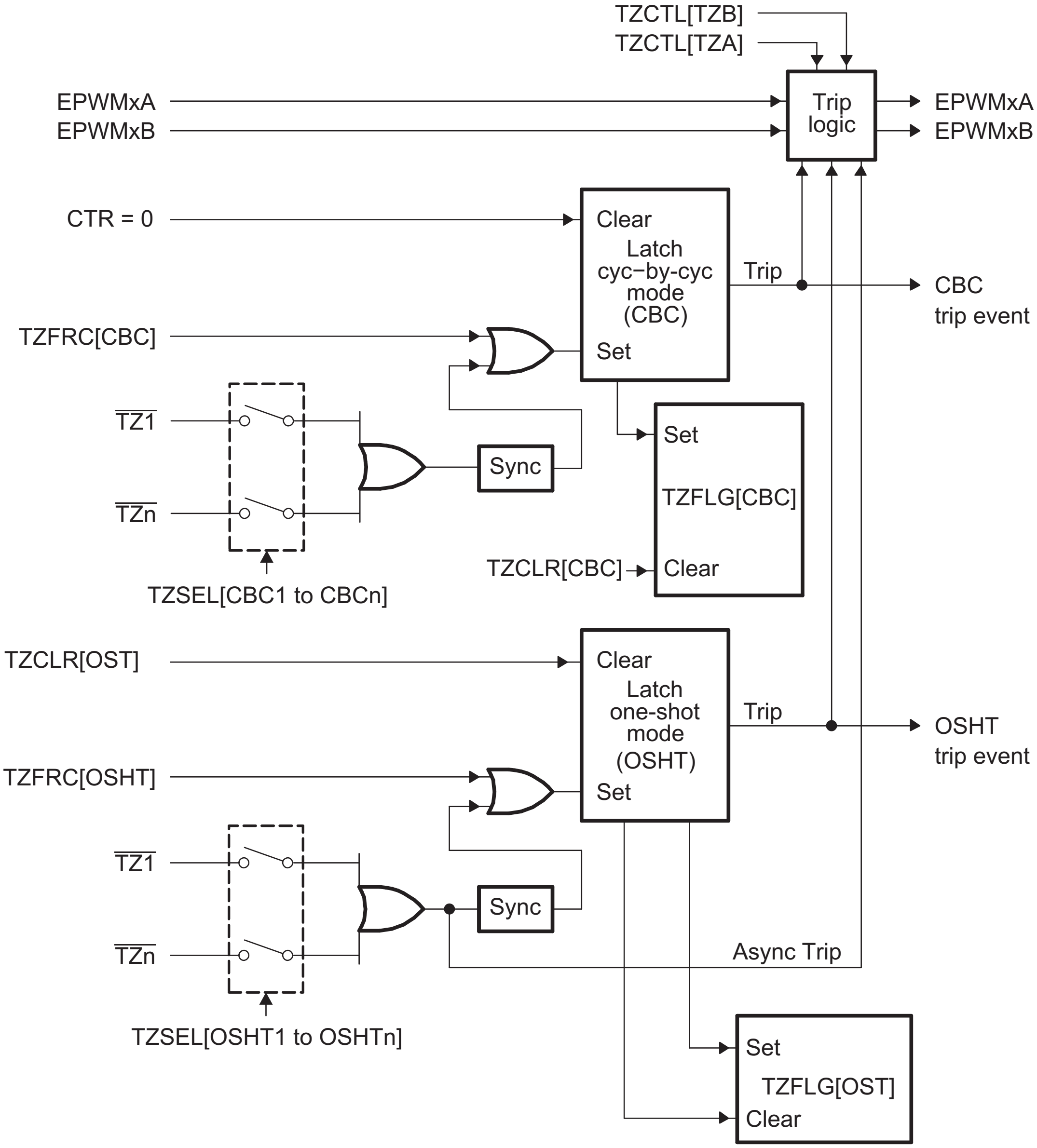
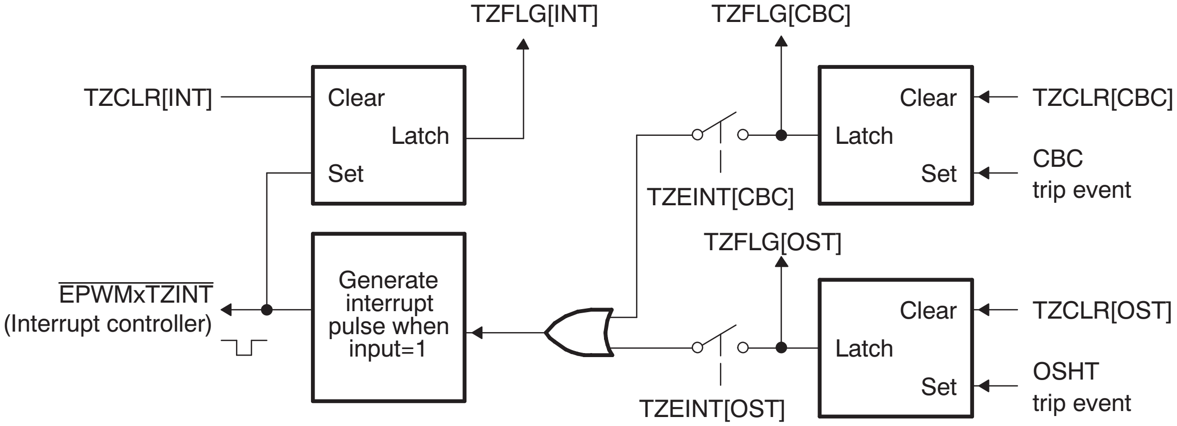


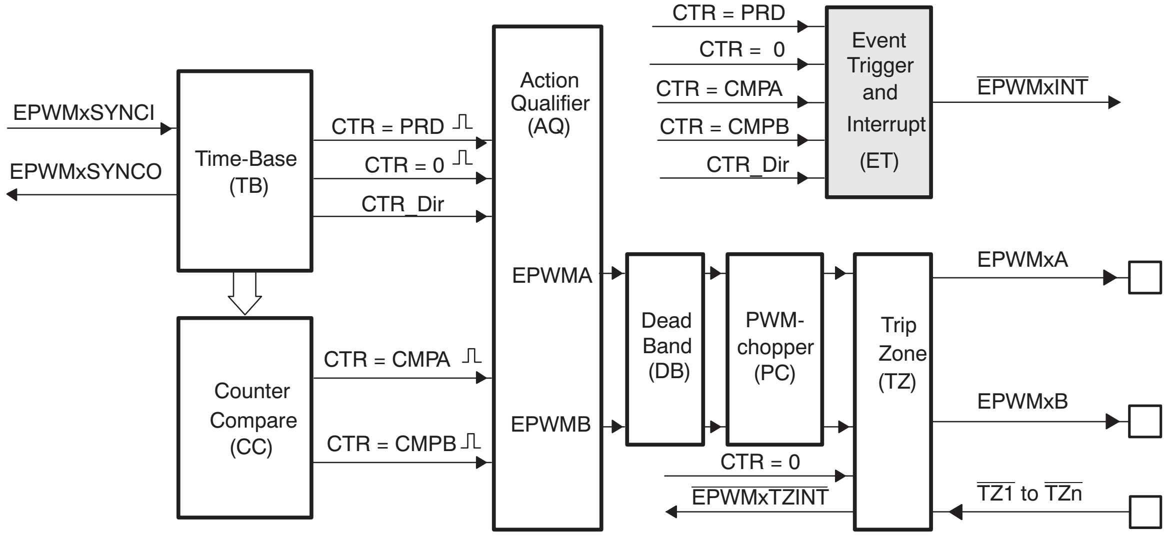
Figure 20-44. Trip-Zone Submodule Interrupt Logic



20.2.2.9 Event-Trigger (ET) Submodule

Figure 20-45 shows the event-trigger (ET) submodule in the ePWM system. The event-trigger submodule manages the events generated by the time-base submodule and the counter-compare submodule to generate an interrupt to the CPU.

Figure 20-45. Event-Trigger Submodule



20.2.2.9.1 Purpose of the Event-Trigger Submodule

The key functions of the event-trigger submodule are:

- Receives event inputs generated by the time-base and counter-compare submodules
- Uses the time-base direction information for up/down event qualification
- Uses prescaling logic to issue interrupt requests at:
  - Every event
  - Every second event
  - Every third event
- Provides full visibility of event generation via event counters and flags

20.2.2.9.2 Controlling and Monitoring the Event-Trigger Submodule

The key registers used to configure the event-trigger submodule are shown in Table 20-38:

Table 20-38. Event-Trigger Submodule Registers

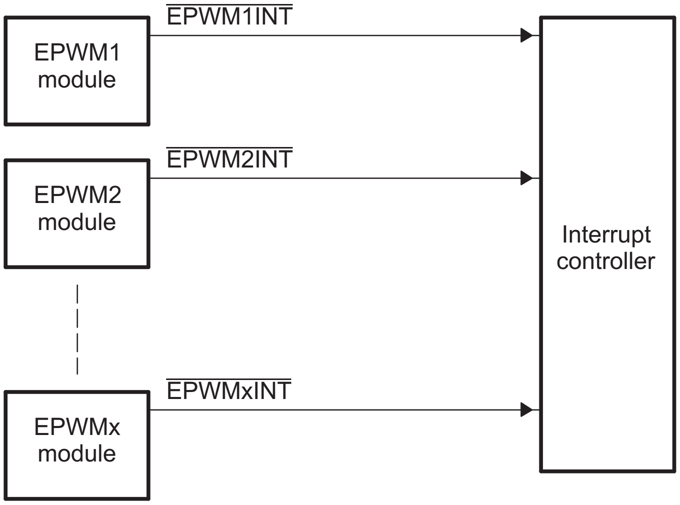
Acronym	Register Description	Address Offset	Shadowed
ETSEL	Event-Trigger Selection Register	32h	No
ETPS	Event-Trigger Prescale Register	34h	No
ETFLG	Event-Trigger Flag Register	36h	No
ETCLR	Event-Trigger Clear Register	38h	No
ETFRC	Event-Trigger Force Register	3Ah	No

20.2.2.9.3 Operational Overview of the Event-Trigger Submodule

The following sections describe the event-trigger submodule's operational highlights.

Each ePWM module has one interrupt request line connected to the interrupt controller as shown in Figure 20-46.

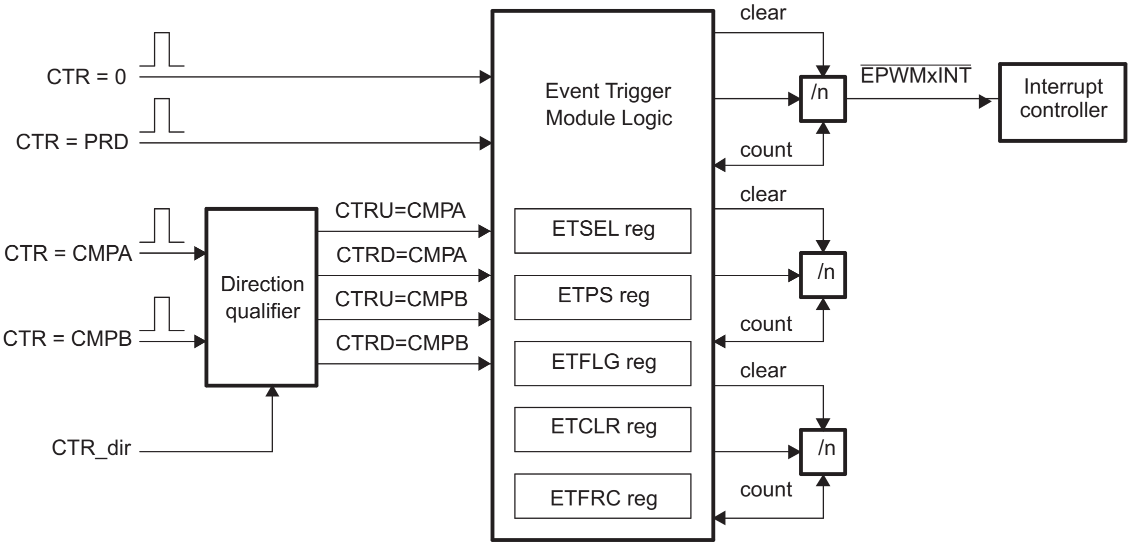
Figure 20-46. Event-Trigger Submodule Inter-Connectivity to Interrupt Controller



The event-trigger submodule monitors various event conditions (the left side inputs to event-trigger submodule shown in Figure 20-47) and can be configured to prescale these events before issuing an Interrupt request. The event-trigger prescaling logic can issue Interrupt requests at:

- Every event
- Every second event
- Every third event

Figure 20-47. Event-Trigger Submodule Showing Event Inputs and Prescaled Outputs



- ETSEL—This selects which of the possible events will trigger an interrupt.
- ETPS—This programs the event prescaling options previously mentioned.
- ETFLG—These are flag bits indicating status of the selected and prescaled events.
- ETCLR—These bits allow you to clear the flag bits in the ETFLG register via software.
- ETFRC—These bits allow software forcing of an event. Useful for debugging or software intervention.

A more detailed look at how the various register bits interact with the Interrupt is shown in Figure 20-48.

Figure 20-48 shows the event-trigger's interrupt generation logic. The interrupt-period (ETPS[INTPRD]) bits specify the number of events required to cause an interrupt pulse to be generated. The choices available are:

- Do not generate an interrupt
- Generate an interrupt on every event
- Generate an interrupt on every second event
- Generate an interrupt on every third event

An interrupt cannot be generated on every fourth or more events.

Which event can cause an interrupt is configured by the interrupt selection (ETSEL[INTSEL]) bits. The event can be one of the following:

- Time-base counter equal to zero (TBCNT = 0000h).
- Time-base counter equal to period (TBCNT = TBPRD).
- Time-base counter equal to the compare A register (CMPA) when the timer is incrementing.
- Time-base counter equal to the compare A register (CMPA) when the timer is decrementing.
- Time-base counter equal to the compare B register (CMPB) when the timer is incrementing.
- Time-base counter equal to the compare B register (CMPB) when the timer is decrementing.

The number of events that have occurred can be read from the interrupt event counter (ETPS[INTCNT]) register bits. That is, when the specified event occurs the ETPS[INTCNT] bits are incremented until they reach the value specified by ETPS[INTPRD]. When ETPS[INTCNT] = ETPS[INTPRD] the counter stops counting and its output is set. The counter is only cleared when an interrupt is sent to the interrupt controller.

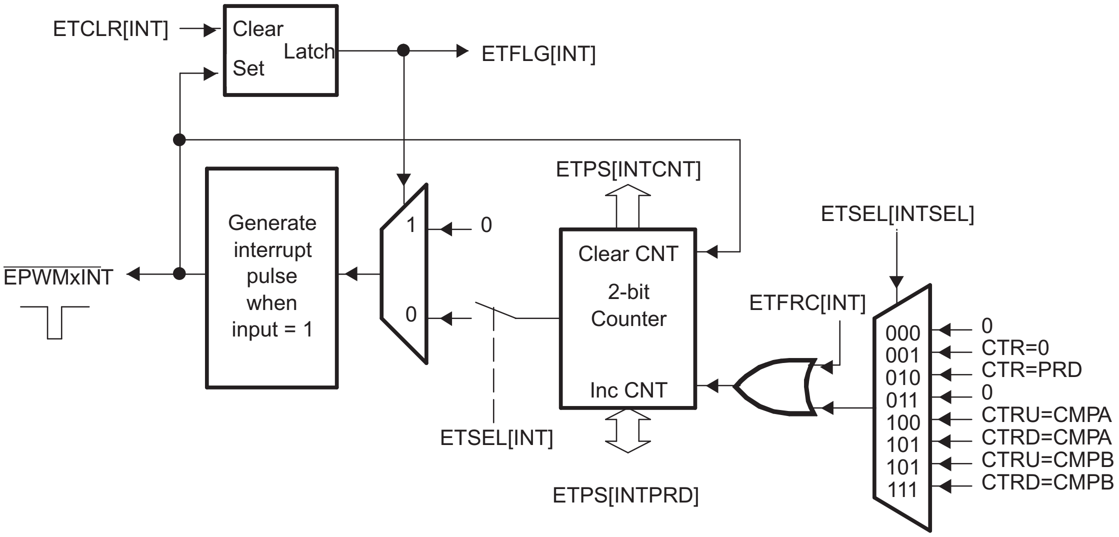
When ETPS[INTCNT] reaches ETPS[INTPRD], one of the following behaviors will occur:

- If interrupts are enabled, ETSEL[INTEN] = 1 and the interrupt flag is clear, ETFLG[INT] = 0, then an interrupt pulse is generated and the interrupt flag is set, ETFLG[INT] = 1, and the event counter is cleared ETPS[INTCNT] = 0. The counter will begin counting events again.
- If interrupts are disabled, ETSEL[INTEN] = 0, or the interrupt flag is set, ETFLG[INT] = 1, the counter stops counting events when it reaches the period value ETPS[INTCNT] = ETPS[INTPRD].
- If interrupts are enabled, but the interrupt flag is already set, then the counter will hold its output high until the ENTFLG[INT] flag is cleared. This allows for one interrupt to be pending while one is serviced.

Writing to the INTPRD bits will automatically clear the counter INTCNT = 0 and the counter output will be reset (so no interrupts are generated). Writing a 1 to the ETFRC[INT] bit will increment the event counter INTCNT. The counter will behave as described above when INTCNT = INTPRD. When INTPRD = 0, the counter is disabled and hence no events will be detected and the ETFRC[INT] bit is also ignored.

Note that the interrupts coming from the ePWM module are also used as DMA events. The interrupt registers should be used to enable and clear the current DMA event in order for the ePWM module to generate subsequent DMA events.

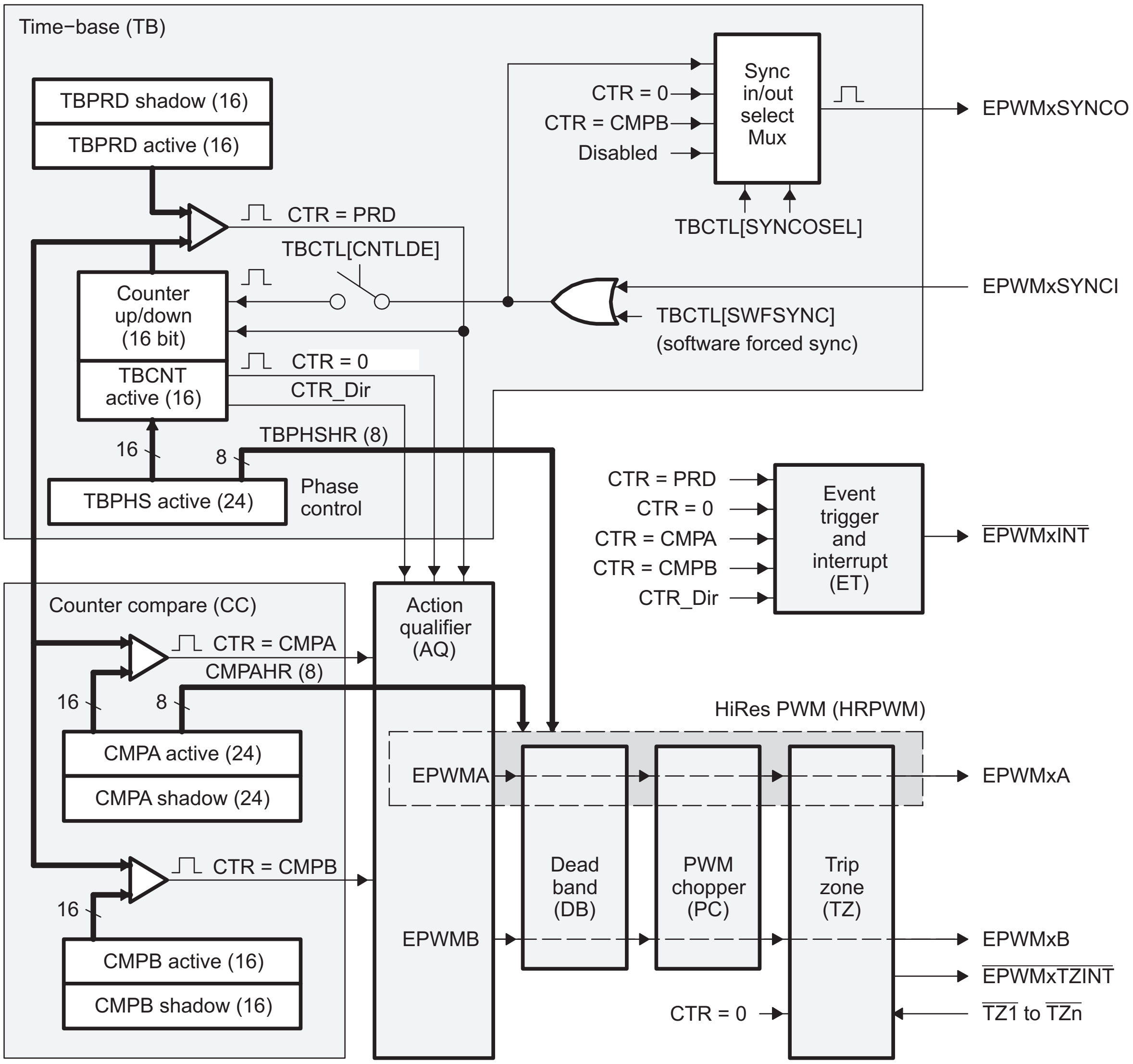
Figure 20-48. Event-Trigger Interrupt Generator



20.2.2.10 High-Resolution PWM (HRPWM) Submodule

Figure 20-49 shows the high-resolution PWM (HRPWM) submodule in the ePWM system. Some devices include the high-resolution PWM submodule, see Section 20.1.2 to determine which ePWM instances include this feature.

Figure 20-49. HRPWM System Interface



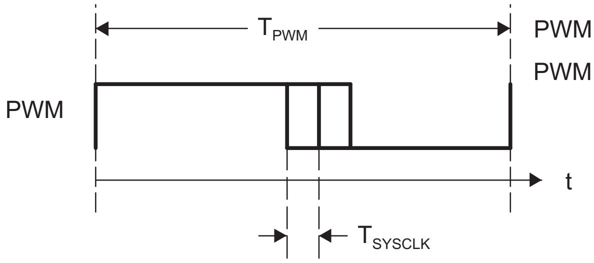
20.2.2.10.1 Purpose of the High-Resolution PWM Submodule

The enhanced high-resolution pulse-width modulator (eHRPWM) extends the time resolution capabilities of the conventionally derived digital pulse-width modulator (PWM). HRPWM is typically used when PWM resolution falls below ~9-10 bits. The key features of HRPWM are:

- Extended time resolution capability
- Used in both duty cycle and phase-shift control methods
- Finer time granularity control or edge positioning using extensions to the Compare A and Phase registers
- Implemented using the A signal path of PWM, that is, on the EPWMxA output. EPWMxB output has conventional PWM capabilities

The ePWM peripheral is used to perform a function that is mathematically equivalent to a digital-to-analog converter (DAC). As shown in Figure 20-50, the effective resolution for conventionally generated PWM is a function of PWM frequency (or period) and system clock frequency.

Figure 20-50. Resolution Calculations for Conventionally Generated PWM



PWM resolution (%) = $F_{PWM}/F_{SYSCLKOUT}$ x 100%

PWM resolution (bits) = $\text{Log}_2\,(F_{PWM}/F_{SYSCLKOUT})$

If the required PWM operating frequency does not offer sufficient resolution in PWM mode, you may want to consider HRPWM. As an example of improved performance offered by HRPWM, Table 20-39 shows resolution in bits for various PWM frequencies. Table 20-39 values assume a MEP step size of 180 ps. See your device-specific data manual for typical and maximum performance specifications for the MEP.

Table 20-39. Resolution for PWM and HRPWM

	Regular Resolution (PWM)		High Resolution (HRPWM)	
PWM Frequency (kHz)	**Bits**	**%**	**Bits**	**%**
20	12.3	0.0	18.1	0.000
50	11.0	0.0	16.8	0.001
100	10.0	0.1	15.8	0.002
150	9.4	0.2	15.2	0.003
200	9.0	0.2	14.8	0.004
250	8.6	0.3	14.4	0.005
500	7.6	0.5	13.8	0.007
1000	6.6	1.0	12.4	0.018
1500	6.1	1.5	11.9	0.027
2000	5.6	2.0	11.4	0.036

Although each application may differ, typical low-frequency PWM operation (below 250 kHz) may not require HRPWM. HRPWM capability is most useful for high-frequency PWM requirements of power conversion topologies such as:

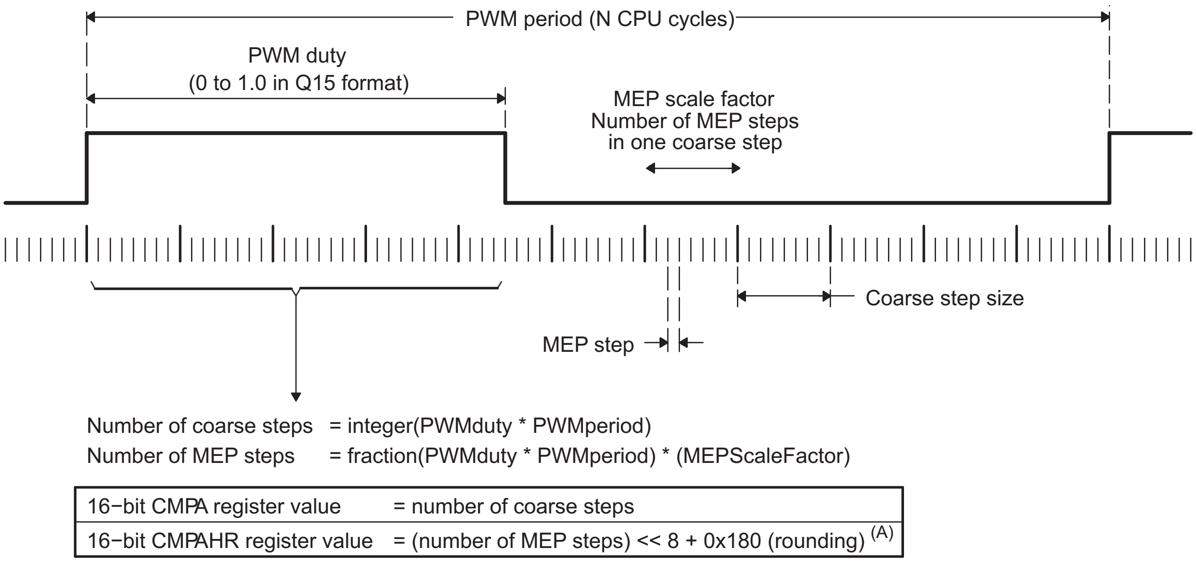
- Single-phase buck, boost, and flyback
- Multi-phase buck, boost, and flyback
- Phase-shifted full bridge
- Direct modulation of D-Class power amplifiers

20.2.2.10.2 Architecture of the High-Resolution PWM Submodule

The HRPWM is based on micro edge positioner (MEP) technology. MEP logic is capable of positioning an edge very finely by sub-dividing one coarse system clock of a conventional PWM generator. The time step accuracy is on the order of 150 ps. The HRPWM also has a self-check software diagnostics mode to check if the MEP logic is running optimally, under all operating conditions.

Figure 20-51 shows the relationship between one coarse system clock and edge position in terms of MEP steps, which are controlled via an 8-bit field in the Compare A extension register (CMPAHR).

Figure 20-51. Operating Logic Using MEP



A For MEP range and rounding adjustment.

To generate an HRPWM waveform, configure the TBM, CCM, and AQM registers as you would to generate a conventional PWM of a given frequency and polarity. The HRPWM works together with the TBM, CCM, and AQM registers to extend edge resolution, and should be configured accordingly. Although many programming combinations are possible, only a few are needed and practical.

20.2.2.10.3 Controlling and Monitoring the High-Resolution PWM Submodule

The MEP of the HRPWM is controlled by two extension registers, each 8-bits wide. These two HRPWM registers are concatenated with the 16-bit TBPHS and CMPA registers used to control PWM operation.

- TBPHSHR - Time-Base Phase High-Resolution Register
- CMPAHR - Counter-Compare A High-Resolution Register

Table 20-40 lists the registers used to control and monitor the high-resolution PWM submodule.

Table 20-40. HRPWM Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
TBPHSHR	Extension Register for HRPWM Phase	4h	No
CMPAHR	Extension Register for HRPWM Duty	10h	Yes
HRCTL	HRPWM Configuration Register	C0h	No

20.2.2.10.4 Configuring the High-Resolution PWM Submodule

Once the ePWM has been configured to provide conventional PWM of a given frequency and polarity, the HRPWM is configured by programming the HRCTL register located at offset address C0h. This register provides configuration options for the following key operating modes:

- **Edge Mode:** The MEP can be programmed to provide precise position control on the rising edge (RE), falling edge (FE), or both edges (BE) at the same time. FE and RE are used for power topologies requiring duty cycle control, while BE is used for topologies requiring phase shifting, for example, phase shifted full bridge.
- **Control Mode:** The MEP is programmed to be controlled either from the CMPAHR register (duty cycle control) or the TBPHSHR register (phase control). RE or FE control mode should be used with CMPAHR register. BE control mode should be used with TBPHSHR register.
- **Shadow Mode:** This mode provides the same shadowing (double buffering) option as in regular PWM mode. This option is valid only when operating from the CMPAHR register and should be chosen to be the same as the regular load option for the CMPA register. If TBPHSHR is used, then this option has no effect.

20.2.2.10.5 Operational Highlights for the High-Resolution PWM Submodule

The MEP logic is capable of placing an edge in one of 255 (8 bits) discrete time steps, each of which has a time resolution on the order of 150 ps. The MEP works with the TBM and CCM registers to be certain that time steps are optimally applied and that edge placement accuracy is maintained over a wide range of PWM frequencies, system clock frequencies and other operating conditions. Table 20-41 shows the typical range of operating frequencies supported by the HRPWM.

Table 20-41. Relationship Between MEP Steps, PWM Frequency and Resolution

System (MHz)	MEP Steps Per SYSCLKOUT[1] [2] [3]	PWM Minimum (Hz)[4]	PWM Maximum (MHz)	Resolution at Maximum (Bits)[5]
50.0	111	763	2.50	11.1
60.0	93	916	3.00	10.9
70.0	79	1068	3.50	10.6
80.0	69	1221	4.00	10.4
90.0	62	1373	4.50	10.3
100.0	56	1526	5.00	10.1

[1] System frequency = SYSCLKOUT, that is, CPU clock. TBCLK = SYSCLKOUT

[2] Table data based on a MEP time resolution of 180 ps (this is an example value)

[3] MEP steps applied = $T_{SYSCLKOUT}$/180 ps in this example.

[4] PWM minimum frequency is based on a maximum period value, TBPRD = 65 535. PWM mode is asymmetrical up-count.

[5] Resolution in bits is given for the maximum PWM frequency stated.

20.2.2.10.5.1 Edge Positioning

In a typical power control loop (switch modes, digital motor control (DMC), uninterruptible power supply (UPS)), a digital controller (PID, 2pole/2zero, lag/lead, etc.) issues a duty command, usually expressed in a per unit or percentage terms.

In the following example, assume that for a particular operating point, the demanded duty cycle is 0.405 or 40.5% on-time and the required converter PWM frequency is 1.25 MHz. In conventional PWM generation with a system clock of 100 MHz, the duty cycle choices are in the vicinity of 40.5%. In Figure 20-52, a compare value of 32 counts (duty = 40%) is the closest to 40.5% that you can attain. This is equivalent to an edge position of 320 ns instead of the desired 324 ns. This data is shown in Table 20-42.

By utilizing the MEP, you can achieve an edge position much closer to the desired point of 324 ns. Table 20-42 shows that in addition to the CMPA value, 22 steps of the MEP (CMPAHR register) will position the edge at 323.96 ns, resulting in almost zero error. In this example, it is assumed that the MEP has a step resolution of 180 ns.

Figure 20-52. Required PWM Waveform for a Requested Duty = 40.5%

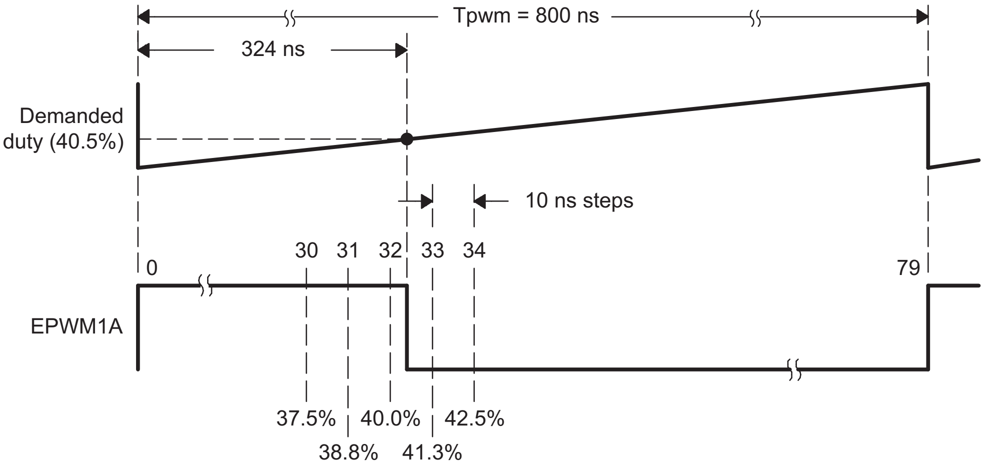


Table 20-42. CMPA vs Duty (left), and [CMPA:CMPAHR] vs Duty (right)

CMPA (count)(1) (2) (3)	DUTY (%)	High Time (ns)	CMPA (count)	CMPAHR (count)	Duty (%)	High Time (ns)
28	35.0	280	32	18	40.405	323.24
29	36.3	290	32	19	40.428	323.42
30	37.5	300	32	20	40.450	323.60
31	38.8	310	32	21	40.473	323.78
32	40.0	320	32	22	40.495	323.96
33	41.3	330	32	23	40.518	324.14
34	42.5	340	32	24	40.540	324.32
			32	25	40.563	324.50
Required			32	26	40.585	324.68
32.40	40.5	324	32	27	40.608	324.86

(1) System clock, SYSCLKOUT and TBCLK = 100 MHz, 10 ns

(2) For a PWM Period register value of 80 counts, PWM Period = 80 × 10 ns = 800 ns, PWM frequency = 1/800 ns = 1.25 MHz

(3) Assumed MEP step size for the above example = 180 ps

20.2.2.10.5.2 Scaling Considerations

The mechanics of how to position an edge precisely in time has been demonstrated using the resources of the standard (CMPA) and MEP (CMPAHR) registers. In a practical application, however, it is necessary to seamlessly provide the CPU a mapping function from a per-unit (fractional) duty cycle to a final integer (non-fractional) representation that is written to the [CMPA:CMPAHR] register combination.

To do this, first examine the scaling or mapping steps involved. It is common in control software to express duty cycle in a per-unit or percentage basis. This has the advantage of performing all needed math calculations without concern for the final absolute duty cycle, expressed in clock counts or high time in ns. Furthermore, it makes the code more transportable across multiple converter types running different PWM frequencies.

To implement the mapping scheme, a two-step scaling procedure is required.

Assumptions for this example:

System clock, SYSCLKOUT = 10 ns (100 MHz)
PWM frequency = 1.25 MHz (1/800 ns)
Required PWM duty cycle, **PWMDuty** = 0.405 (40.5%)
PWM period in terms of coarse steps, **PWMperiod** (800 ns/10 ns) = 80
Number of MEP steps per coarse step at 180 ps (10 ns/180 ps), **MEP_SF** = 55
Value to keep CMPAHR within the range of 1-255 and fractional rounding constant (default value) = 180h

Step 1: Percentage Integer Duty value conversion for CMPA register

CMPA register value = int(**PWMDuty** × **PWMperiod**); int means integer part
= int(0.405 × 80)
= int(32.4)
CMPA register value = 32 (20h)

Step 2: Fractional value conversion for CMPAHR register

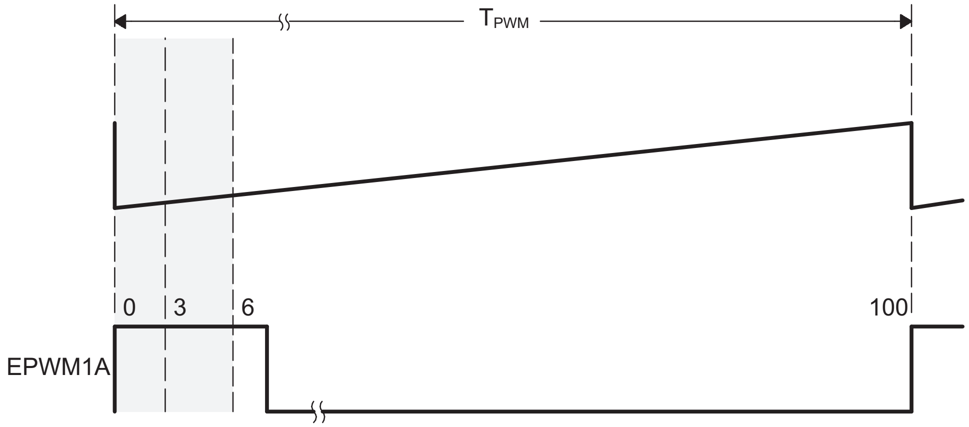
CMPAHR register value = (frac(**PWMDuty** × **PWMperiod**) × **MEP_SF**) << 8) + 180h; frac means fractional part
= (frac(32.4) × 55 <<8) + 180h; Shift is to move the value as CMPAHR high byte
= ((0.4 × 55) <<8) + 180h
= (22 <<8) + 180h
= 22 × 256 + 180h ; Shifting left by 8 is the same multiplying by 256.
= 5632 + 180h
= 1600h + 180h
CMPAHR value = 1780h; CMPAHR value = 1700h, lower 8 bits will be ignored by hardware.

20.2.2.10.5.3 Duty Cycle Range Limitation

In high resolution mode, the MEP is not active for 100% of the PWM period. It becomes operational 3 SYSCLK cycles after the period starts.

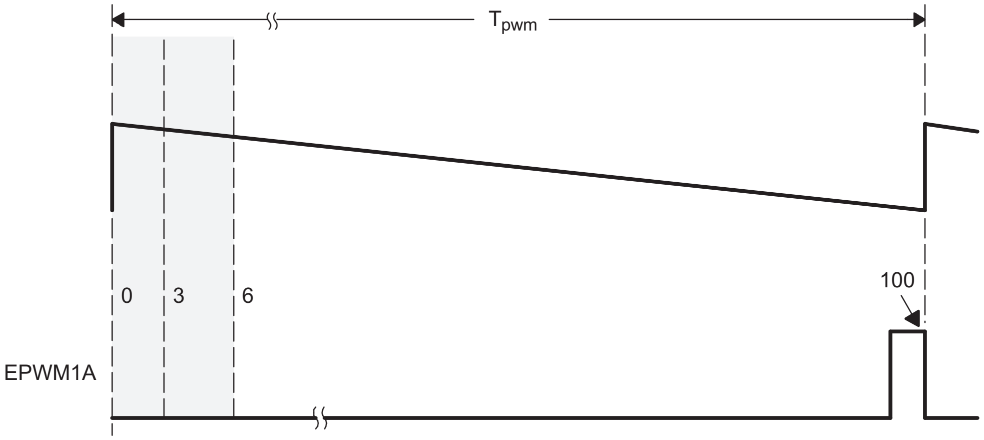
Duty cycle range limitations are illustrated in Figure 20-53. This limitation imposes a lower duty cycle limit on the MEP. For example, precision edge control is not available all the way down to 0% duty cycle. Although for the first 3 or 6 cycles, the HRPWM capabilities are not available, regular PWM duty control is still fully operational down to 0% duty. In most applications this should not be an issue as the controller regulation point is usually not designed to be close to 0% duty cycle.

Figure 20-53. Low % Duty Cycle Range Limitation Example When PWM Frequency = 1 MHz



If the application demands HRPWM operation in the low percent duty cycle region, then the HRPWM can be configured to operate in count-down mode with the rising edge position (REP) controlled by the MEP. This is illustrated in Figure 20-54. In this case low percent duty limitation is no longer an issue.

Figure 20-54. High % Duty Cycle Range Limitation Example when PWM Frequency = 1 MHz



20.2.2.11 ePWM Behavior During Emulation

To configure the ePWM to stop during emulation suspend events (for example, debugger breakpoints), set up the ePWM and the Debug Subsystem:

1. Set TBCTL.FREE_SOFT= 0 or 1 (see register description for more details). This will allow the Suspend_Control signal from the Debug Subsystem (Chapter 31) to stop and start the ePWM. Note that if FREE_SOFT = 2 or 3, the Suspend_Control signal is ignored and the ePWM is free running regardless of any debug suspend event. This FREE_SOFT bit gives local control from a module perspective to gate the suspend signal coming from the Debug Subsystem.
2. Set the appropriate xxx_Suspend_Control register = 0x9, as described in Section 31.1.1.1, *Debug Suspend Support for Peripherals*. Choose the register appropriate to the peripheral you want to suspend during a suspend event.

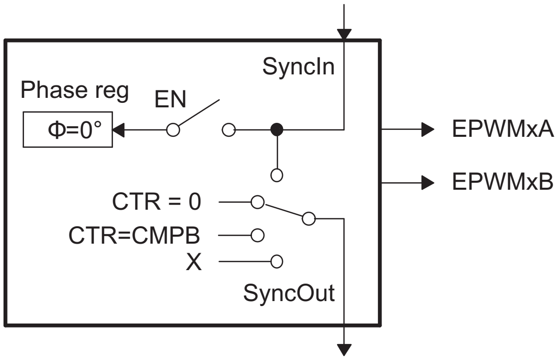
20.2.3 Use Cases

An ePWM module has all the local resources necessary to operate completely as a standalone module or to operate in synchronization with other identical ePWM modules.

20.2.3.1 Overview of Multiple Modules

Previously in this user's guide, all discussions have described the operation of a single module. To facilitate the understanding of multiple modules working together in a system, the ePWM module described in reference is represented by the more simplified block diagram shown in Figure 20-55. This simplified ePWM block shows only the key resources needed to explain how a multiswitch power topology is controlled with multiple ePWM modules working together.

Figure 20-55. Simplified ePWM Module



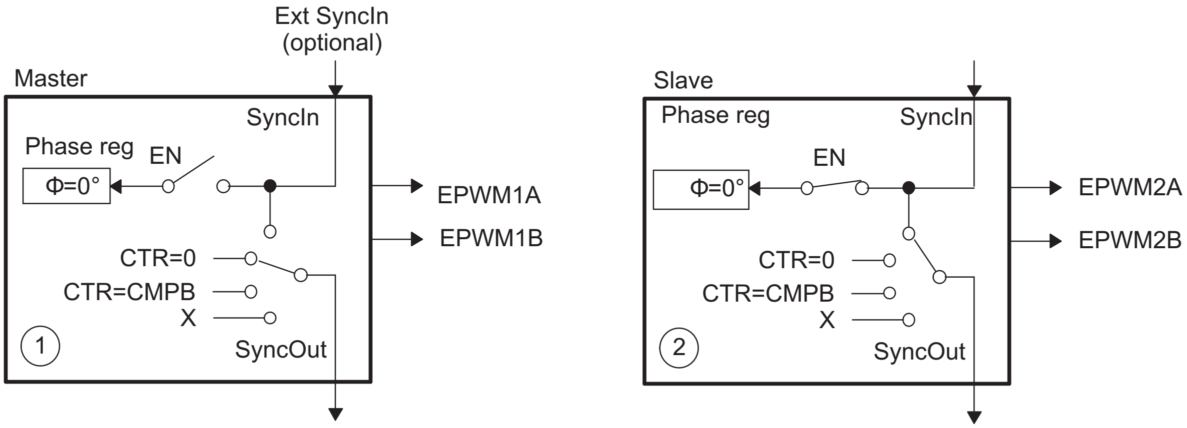
20.2.3.2 Key Configuration Capabilities

The key configuration choices available to each module are as follows:

- Options for SyncIn
  - Load own counter with phase register on an incoming sync strobe—enable (EN) switch closed
  - Do nothing or ignore incoming sync strobe—enable switch open
  - Sync flow-through - SyncOut connected to SyncIn
  - Master mode, provides a sync at PWM boundaries—SyncOut connected to CTR = PRD
  - Master mode, provides a sync at any programmable point in time—SyncOut connected to CTR = CMPB
  - Module is in standalone mode and provides No sync to other modules—SyncOut connected to X (disabled)
- Options for SyncOut
  - Sync flow-through - SyncOut connected to SyncIn
  - Master mode, provides a sync at PWM boundaries—SyncOut connected to CTR = PRD
  - Master mode, provides a sync at any programmable point in time—SyncOut connected to CTR = CMPB
  - Module is in standalone mode and provides No sync to other modules—SyncOut connected to X (disabled)

For each choice of SyncOut, a module may also choose to load its own counter with a new phase value on a SyncIn strobe input or choose to ignore it, i.e., via the enable switch. Although various combinations are possible, the two most common—master module and slave module modes—are shown in Figure 20-56.

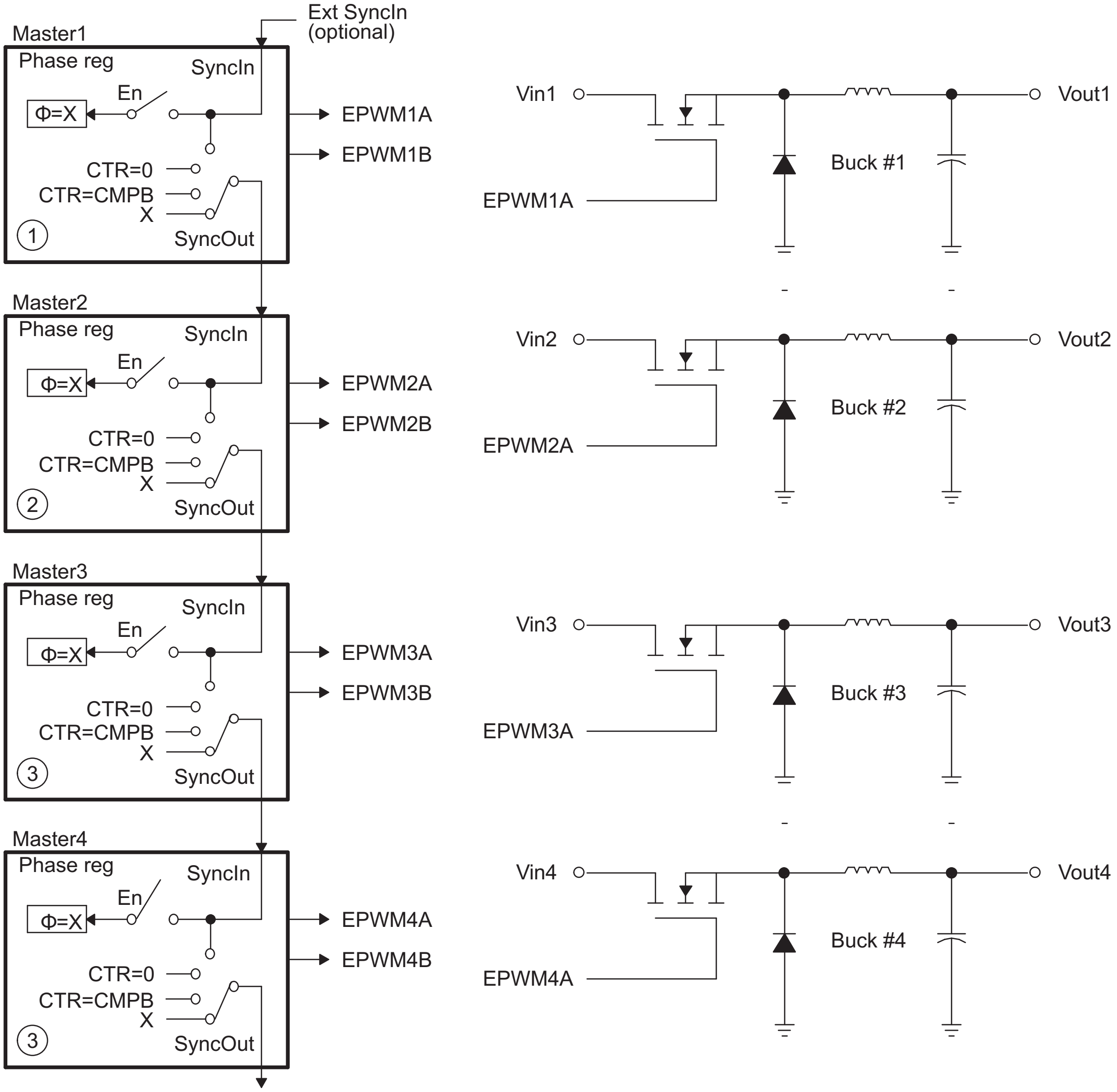
Figure 20-56. EPWM1 Configured as a Typical Master, EPWM2 Configured as a Slave



20.2.3.3 Controlling Multiple Buck Converters With Independent Frequencies

One of the simplest power converter topologies is the buck. A single ePWM module configured as a master can control two buck stages with the same PWM frequency. If independent frequency control is required for each buck converter, then one ePWM module must be allocated for each converter stage. Figure 20-57 shows four buck stages, each running at independent frequencies. In this case, all four ePWM modules are configured as Masters and no synchronization is used. Figure 20-58 shows the waveforms generated by the setup shown in Figure 20-57; note that only three waveforms are shown, although there are four stages.

Figure 20-57. Control of Four Buck Stages. Here $F_{PWM1} \neq F_{PWM2} \neq F_{PWM3} \neq F_{PWM4}$



NOTE: Θ = X indicates value in phase register is a "don't care"

Figure 20-58. Buck Waveforms for Figure 20-57 (Note: Only three bucks shown here)

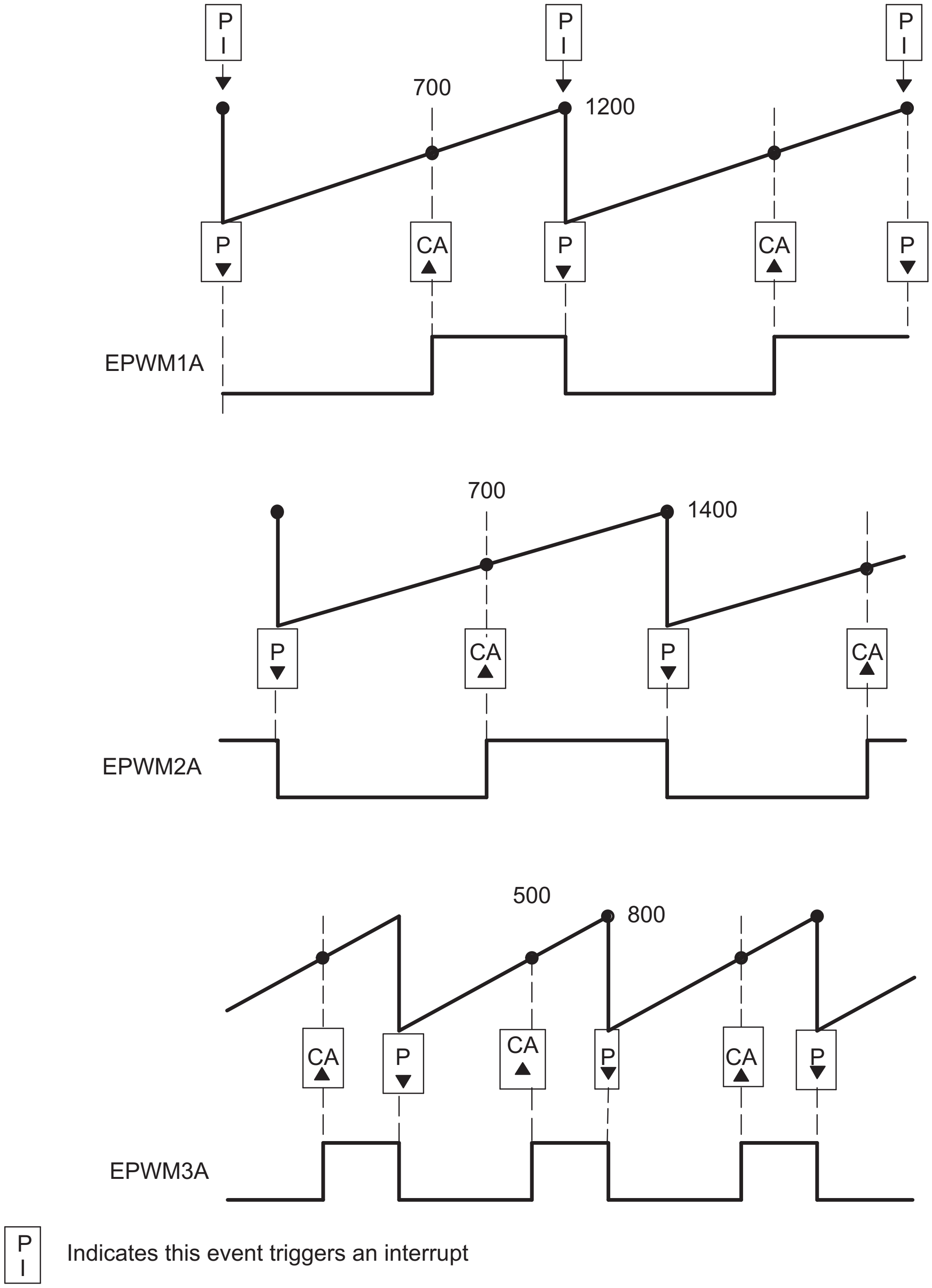


Table 20-43. EPWM1 Initialization for Figure 20-58

Register	Bit	Value	Comments
TBPRD	TBPRD	1200 (4B0h)	Period = 1201 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	PRD	AQ_CLEAR	
	CAU	AQ_SET	

Table 20-44. EPWM2 Initialization for Figure 20-58

Register	Bit	Value	Comments
TBPRD	TBPRD	1400 (578h)	Period = 1401 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	PRD	AQ_CLEAR	
	CAU	AQ_SET	

Table 20-45. EPWM3 Initialization for Figure 20-58

Register	Bit	Value	Comments
TBPRD	TBPRD	800 (320h)	Period = 801 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	PRD	AQ_CLEAR	
	CAU	AQ_SET	

Example 20-3. Configuration for Example in Figure 20-58

```
// Run Time (Note: Example execution of one run-time instance)
//===========================================================
    EPwm1Regs.CMPA.half.CMPA = 700;               // adjust duty for output EPWM1A
    EPwm2Regs.CMPA.half.CMPA = 700;               // adjust duty for output EPWM2A
    EPwm3Regs.CMPA.half.CMPA = 500;               // adjust duty for output EPWM3A
```

20.2.3.4 Controlling Multiple Buck Converters With Same Frequencies

If synchronization is a requirement, ePWM module 2 can be configured as a slave and can operate at integer multiple (N) frequencies of module 1. The sync signal from master to slave ensures these modules remain locked. Figure 20-59 shows such a configuration; Figure 20-60 shows the waveforms generated by the configuration.

Figure 20-59. Control of Four Buck Stages. (Note: $F_{PWM2} = N \times F_{PWM1}$)

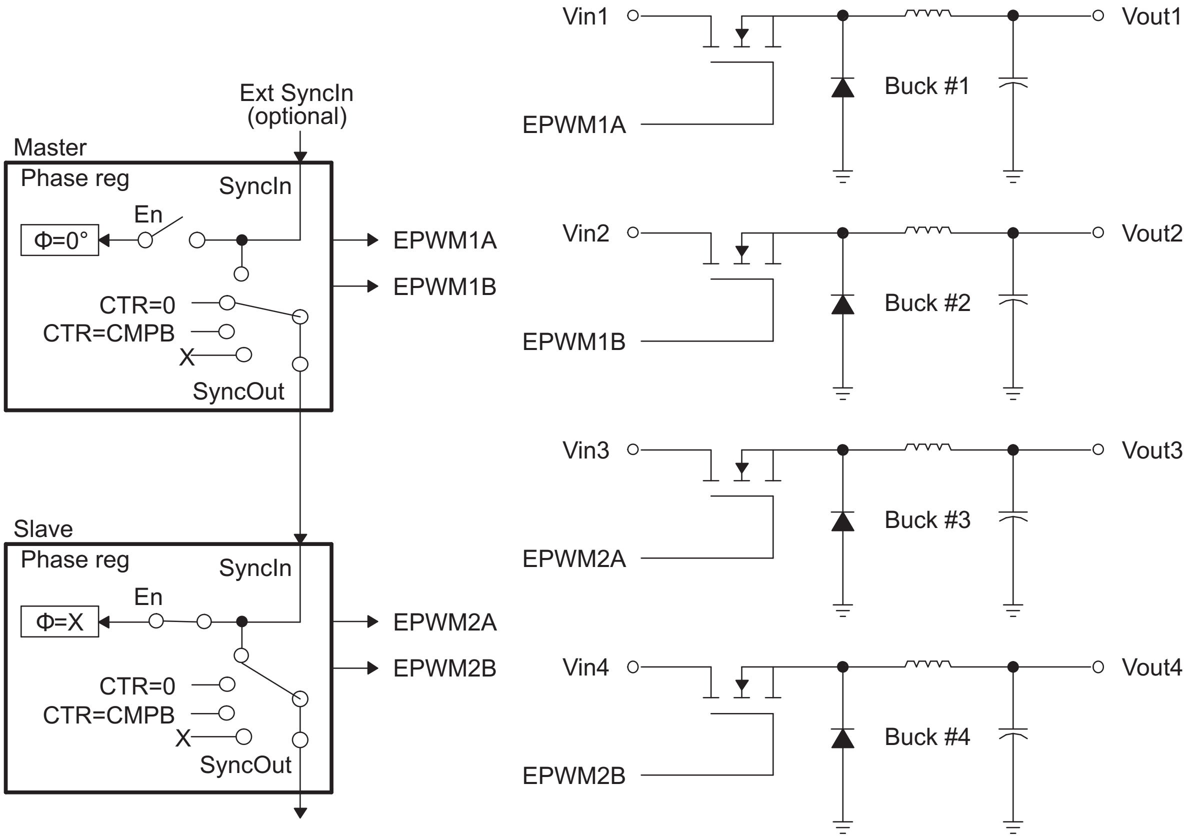


Figure 20-60. Buck Waveforms for Figure 20-59 (Note: $F_{PWM2} = F_{PWM1}$)

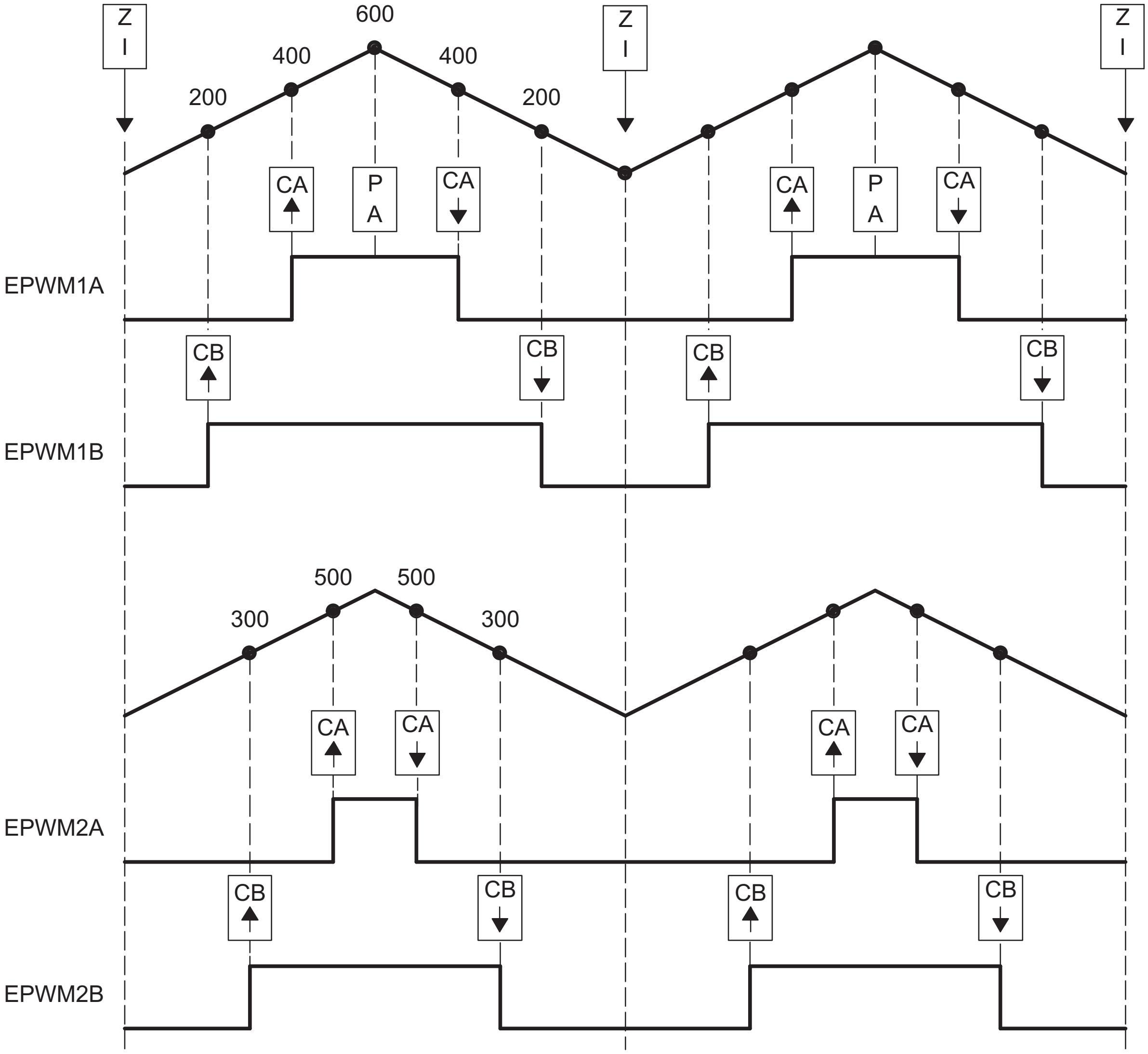


Table 20-46. EPWM1 Initialization for Figure 20-59

Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 1200 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_CTR_ZERO	Sync down-stream module
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	Set actions for EPWM1A
	CAD	AQ_CLEAR	
AQCTLB	CBU	AQ_SET	Set actions for EPWM1B
	CBD	AQ_CLEAR	

Table 20-47. EPWM2 Initialization for Figure 20-59

Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 1200 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_ENABLE	Phase loading enabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_IN	Sync flow-through
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	Set actions for EPWM2A
	CAD	AQ_CLEAR	
AQCTLB	CBU	AQ_SET	Set actions for EPWM2B
	CBD	AQ_CLEAR	

Example 20-4. Code Snippet for Configuration in Figure 20-59

```
// Run Time (Note: Example execution of one run-time instance)
//=========================================================
     EPwm1Regs.CMPA.half.CMPA = 400;       // adjust duty for output EPWM1A
     EPwm1Regs.CMPB = 200;                 // adjust duty for output EPWM1B
     EPwm2Regs.CMPA.half.CMPA = 500;       // adjust duty for output EPWM2A
     EPwm2Regs.CMPB = 300;                 // adjust duty for output EPWM2B
```

20.2.3.5 Controlling Multiple Half H-Bridge (HHB) Converters

Topologies that require control of multiple switching elements can also be addressed with these same ePWM modules. It is possible to control a Half-H bridge stage with a single ePWM module. This control can be extended to multiple stages. Figure 20-61 shows control of two synchronized Half-H bridge stages where stage 2 can operate at integer multiple (N) frequencies of stage 1. Figure 20-62 shows the waveforms generated by the configuration shown in Figure 20-61.

Module 2 (slave) is configured for Sync flow-through; if required, this configuration allows for a third Half-H bridge to be controlled by PWM module 3 and also, most importantly, to remain in synchronization with master module 1.

Figure 20-61. Control of Two Half-H Bridge Stages ($F_{PWM2} = N \times F_{PWM1}$)

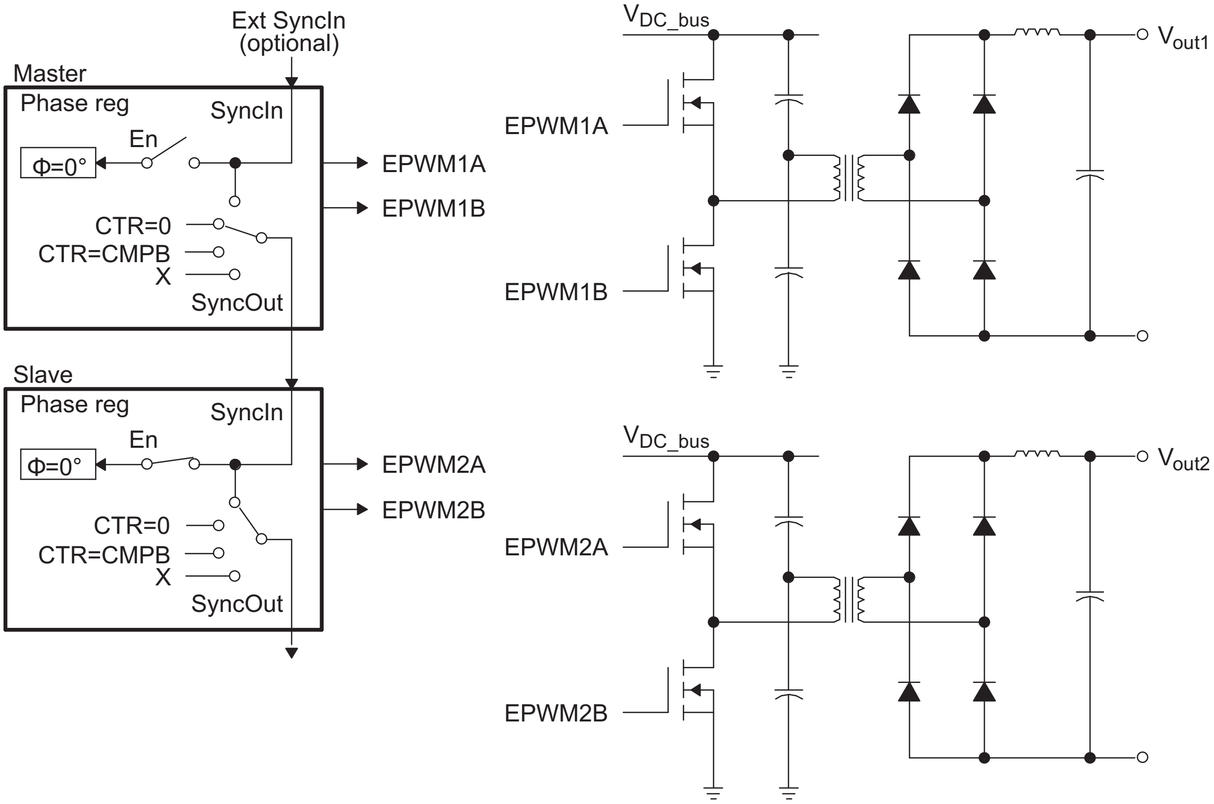


Figure 20-62. Half-H Bridge Waveforms for Figure 20-61 (Note: Here $F_{PWM2} = F_{PWM1}$)

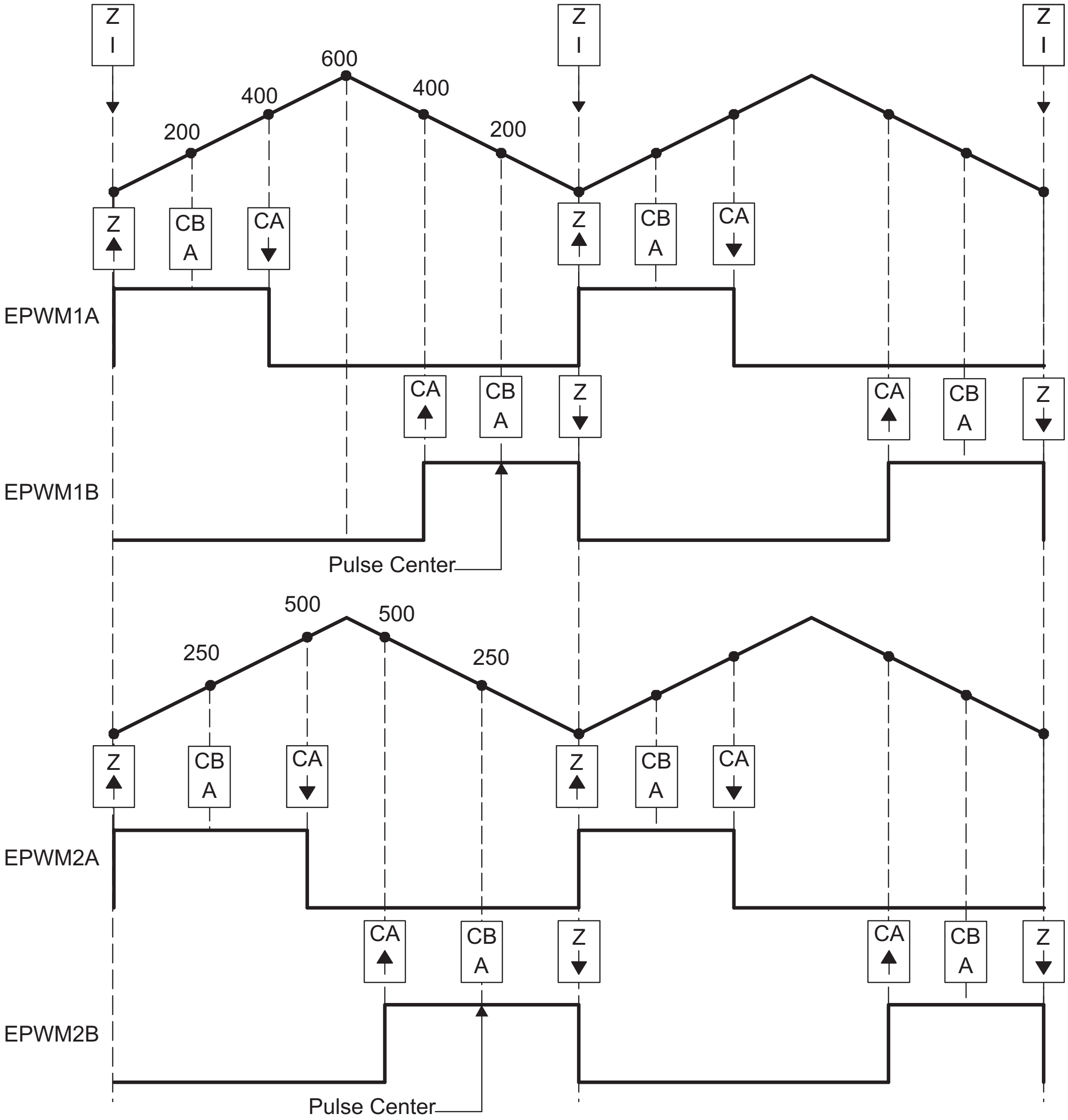


Table 20-48. EPWM1 Initialization for Figure 20-61

Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 1200 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_CTR_ZERO	Sync down-stream module
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	ZRO	AQ_SET	Set actions for EPWM1A
	CAU	AQ_CLEAR	
AQCTLB	ZRO	AQ_CLEAR	Set actions for EPWM1B
	CAD	AQ_SET	

Table 20-49. EPWM2 Initialization for Figure 20-61

Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 1200 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_ENABLE	Phase loading enabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_IN	Sync flow-through
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	ZRO	AQ_SET	Set actions for EPWM2A
	CAU	AQ_CLEAR	
AQCTLB	ZRO	AQ_CLEAR	Set actions for EPWM2B
	CAD	AQ_SET	

Example 20-5. Code Snippet for Configuration in Figure 20-61

```
// Run Time (Note: Example execution of one run-time instance)
//=========================================================
     EPwm1Regs.CMPA.half.CMPA = 400; // adjust duty for output EPWM1A
     EPwm1Regs.CMPB = 200;           // adjust duty for output EPWM1B
     EPwm2Regs.CMPA.half.CMPA = 500; // adjust duty for output EPWM2A
     EPwm2Regs.CMPB = 250;           // adjust duty for output EPWM2B
```

20.2.3.6 Controlling Dual 3-Phase Inverters for Motors (ACI and PMSM)

The idea of multiple modules controlling a single power stage can be extended to the 3-phase Inverter case. In such a case, six switching elements can be controlled using three PWM modules, one for each leg of the inverter. Each leg must switch at the same frequency and all legs must be synchronized. A master + two slaves configuration can easily address this requirement. Figure 20-63 shows how six PWM modules can control two independent 3-phase Inverters; each running a motor.

As in the cases shown in the previous sections, we have a choice of running each inverter at a different frequency (module 1 and module 4 are masters as in Figure 20-63), or both inverters can be synchronized by using one master (module 1) and five slaves. In this case, the frequency of modules 4, 5, and 6 (all equal) can be integer multiples of the frequency for modules 1, 2, 3 (also all equal).

Figure 20-63. Control of Dual 3-Phase Inverter Stages as Is Commonly Used in Motor Control

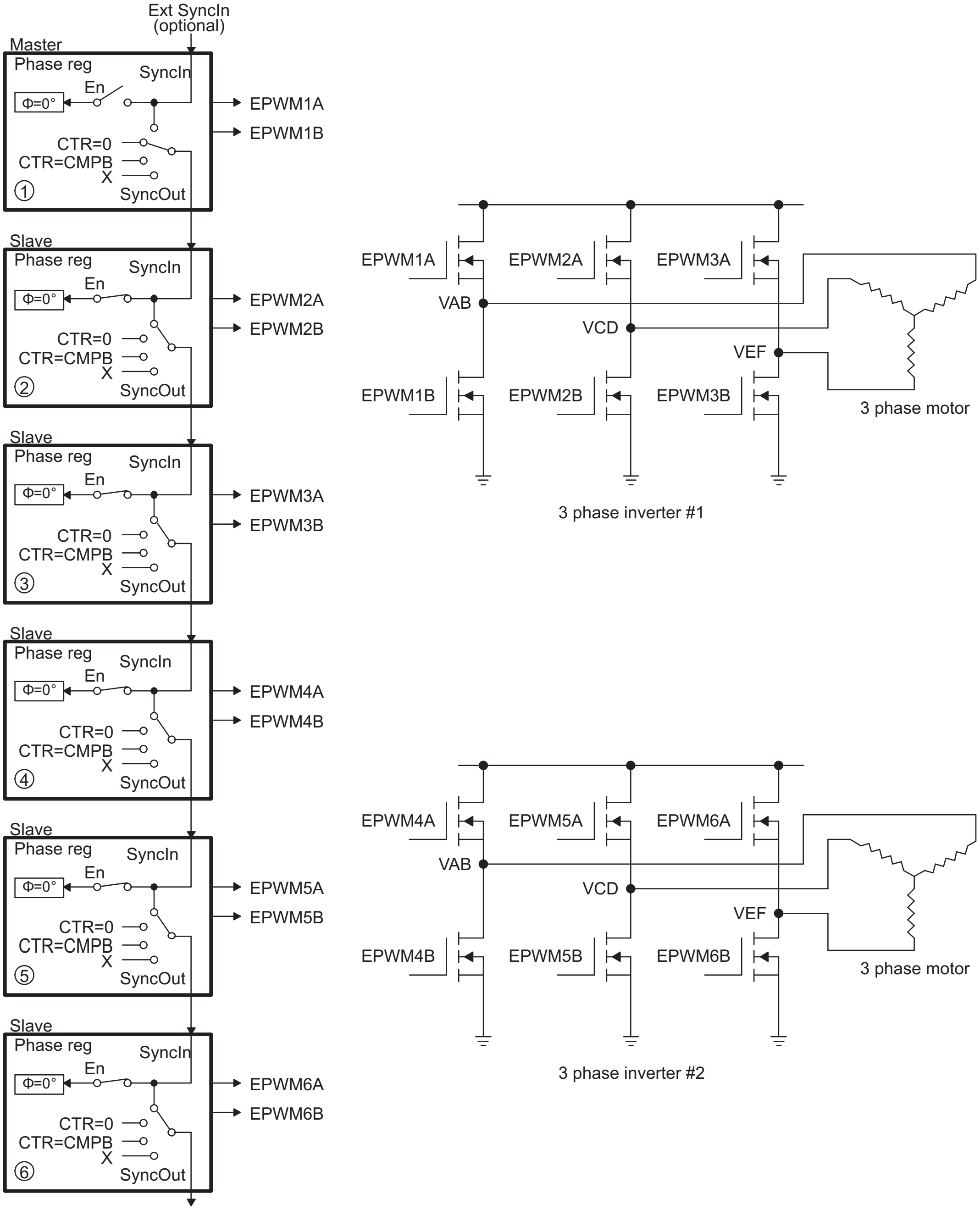


Figure 20-64. 3-Phase Inverter Waveforms for Figure 20-63 (Only One Inverter Shown)

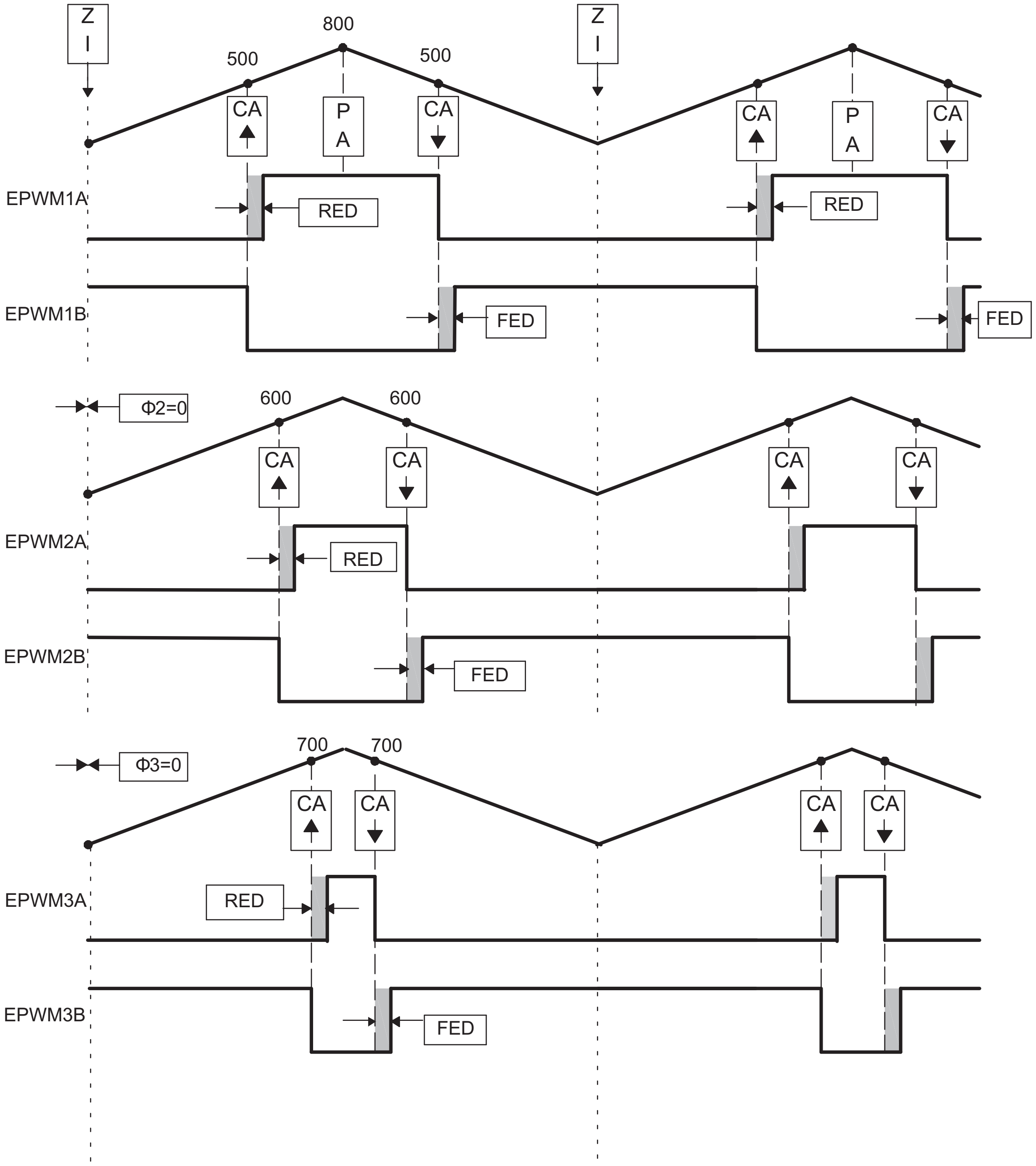


Table 20-50. EPWM1 Initialization for Figure 20-63

Register	Bit	Value	Comments
TBPRD	TBPRD	800 (320h)	Period = 1600 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_CTR_ZERO	Sync down-stream module
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	Set actions for EPWM1A
	CAD	AQ_CLEAR	
DBCTL	MODE	DB_FULL_ENABLE	Enable Dead-band module
	POLSEL	DB_ACTV_HIC	Active Hi complementary
DBFED	DBFED	50	FED = 50 TBCLKs
	DBRED	50	RED = 50 TBCLKs

Table 20-51. EPWM2 Initialization for Figure 20-63

Register	Bit	Value	Comments
TBPRD	TBPRD	800 (320h)	Period = 1600 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_ENABLE	Slave module
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_IN	Sync flow-through
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	Set actions for EPWM2A
	CAD	AQ_CLEAR	
DBCTL	MODE	DB_FULL_ENABLE	Enable Dead-band module
	POLSEL	DB_ACTV_HIC	Active Hi complementary
DBFED	DBFED	50	FED = 50 TBCLKs
	DBRED	50	RED = 50 TBCLKs

Table 20-52. EPWM3 Initialization for Figure 20-63

Register	Bit	Value	Comments
TBPRD	TBPRD	800 (320h)	Period = 1600 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_ENABLE	Slave module
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_IN	Sync flow-through
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	Set actions for EPWM3A
	CAD	AQ_CLEAR	
DBCTL	MODE	DB_FULL_ENABLE	Enable Dead-band module
	POLSEL	DB_ACTV_HIC	Active Hi complementary
DBFED	DBFED	50	FED = 50 TBCLKs
	DBRED	50	RED = 50 TBCLKs

Example 20-6. Code Snippet for Configuration in Figure 20-63

```
// Run Time (Note: Example execution of one run-time instance)
//=========================================================
EPwm1Regs.CMPA.half.CMPA = 500; // adjust duty for output EPWM1A
EPwm2Regs.CMPA.half.CMPA = 600; // adjust duty for output EPWM2A
EPwm3Regs.CMPA.half.CMPA = 700; // adjust duty for output EPWM3A
```

20.2.3.7 Practical Applications Using Phase Control Between PWM Modules

So far, none of the examples have made use of the phase register (TBPHS). It has either been set to zero or its value has been a don't care. However, by programming appropriate values into TBPHS, multiple PWM modules can address another class of power topologies that rely on phase relationship between legs (or stages) for correct operation. As described in the TB module section, a PWM module can be configured to allow a SyncIn pulse to cause the TBPHS register to be loaded into the TBCNT register. To illustrate this concept, Figure 20-65 shows a master and slave module with a phase relationship of 120°, that is, the slave leads the master.

Figure 20-65. Configuring Two PWM Modules for Phase Control

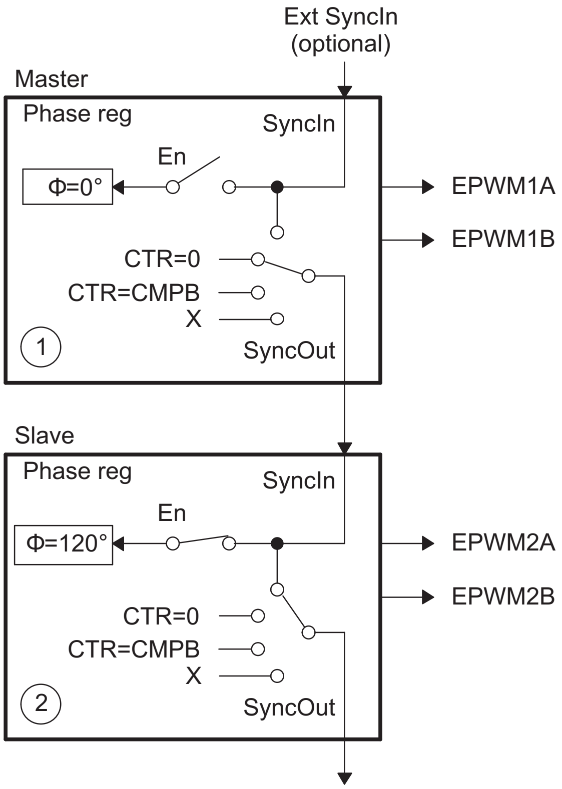
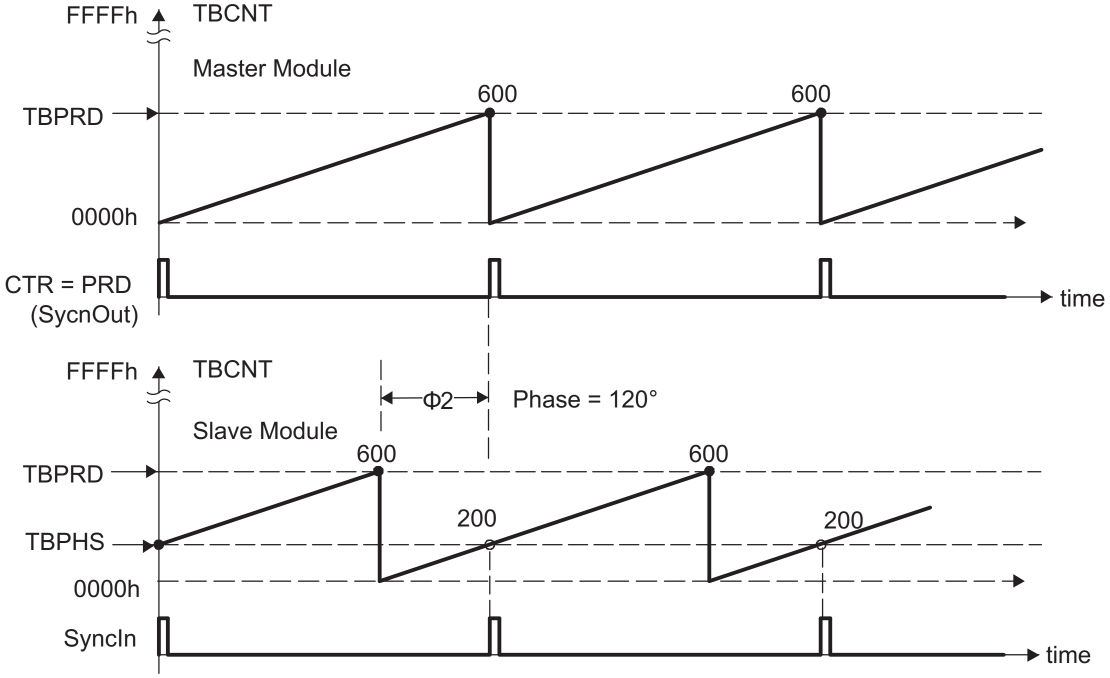


Figure 20-66 shows the associated timing waveforms for this configuration. Here, TBPRD = 600 for both master and slave. For the slave, TBPHS = 200 (200/600 × 360° = 120°). Whenever the master generates a SyncIn pulse (CTR = PRD), the value of TBPHS = 200 is loaded into the slave TBCNT register so the slave time-base is always leading the master's time-base by 120°.

Figure 20-66. Timing Waveforms Associated With Phase Control Between 2 Modules



20.2.3.8 Controlling a 3-Phase Interleaved DC/DC Converter

A popular power topology that makes use of phase-offset between modules is shown in Figure 20-67. This system uses three PWM modules, with module 1 configured as the master. To work, the phase relationship between adjacent modules must be F = 120°. This is achieved by setting the slave TBPHS registers 2 and 3 with values of 1/3 and 2/3 of the period value, respectively. For example, if the period register is loaded with a value of 600 counts, then TBPHS (slave 2) = 200 and TBPHS (slave 3) = 400. Both slave modules are synchronized to the master 1 module.

This concept can be extended to four or more phases, by setting the TBPHS values appropriately. The following formula gives the TBPHS values for N phases:

TBPHS(N,M) = (TBPRD/N) × (M - 1)

Where:

N = number of phases

M = PWM module number

For example, for the 3-phase case (N = 3), TBPRD = 600,

TBPHS(3,2) = (600/3) × (2 - 1) = 200 × 1 = 200 (Phase value for Slave module 2)

TBPHS(3,3) = (600/3) × (3 - 1) = 200 × 2 = 400 (Phase value for Slave module 3)

Figure 20-68 shows the waveforms for the configuration in Figure 20-67.

Figure 20-67. Control of a 3-Phase Interleaved DC/DC Converter

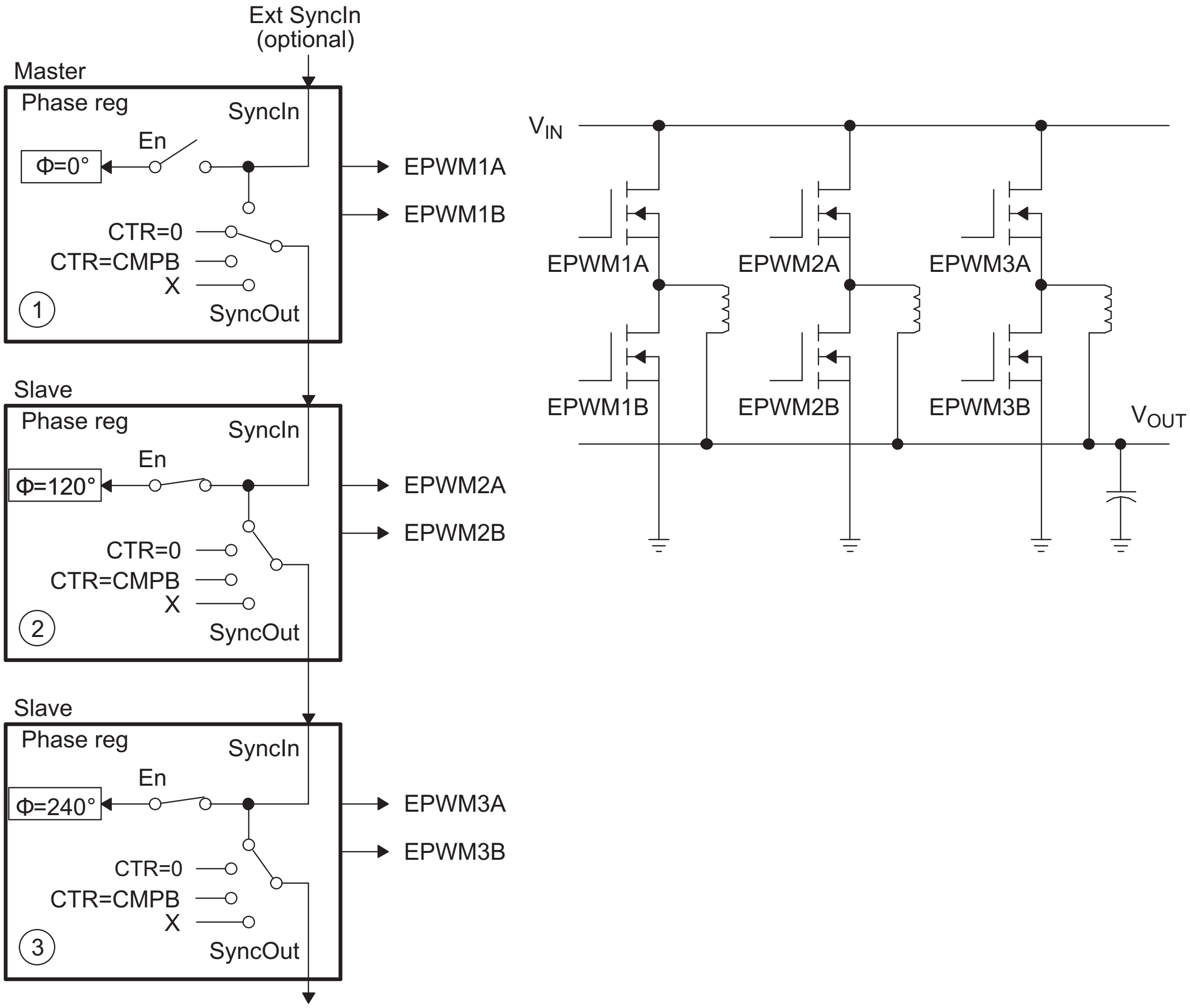


Figure 20-68. 3-Phase Interleaved DC/DC Converter Waveforms for Figure 20-67

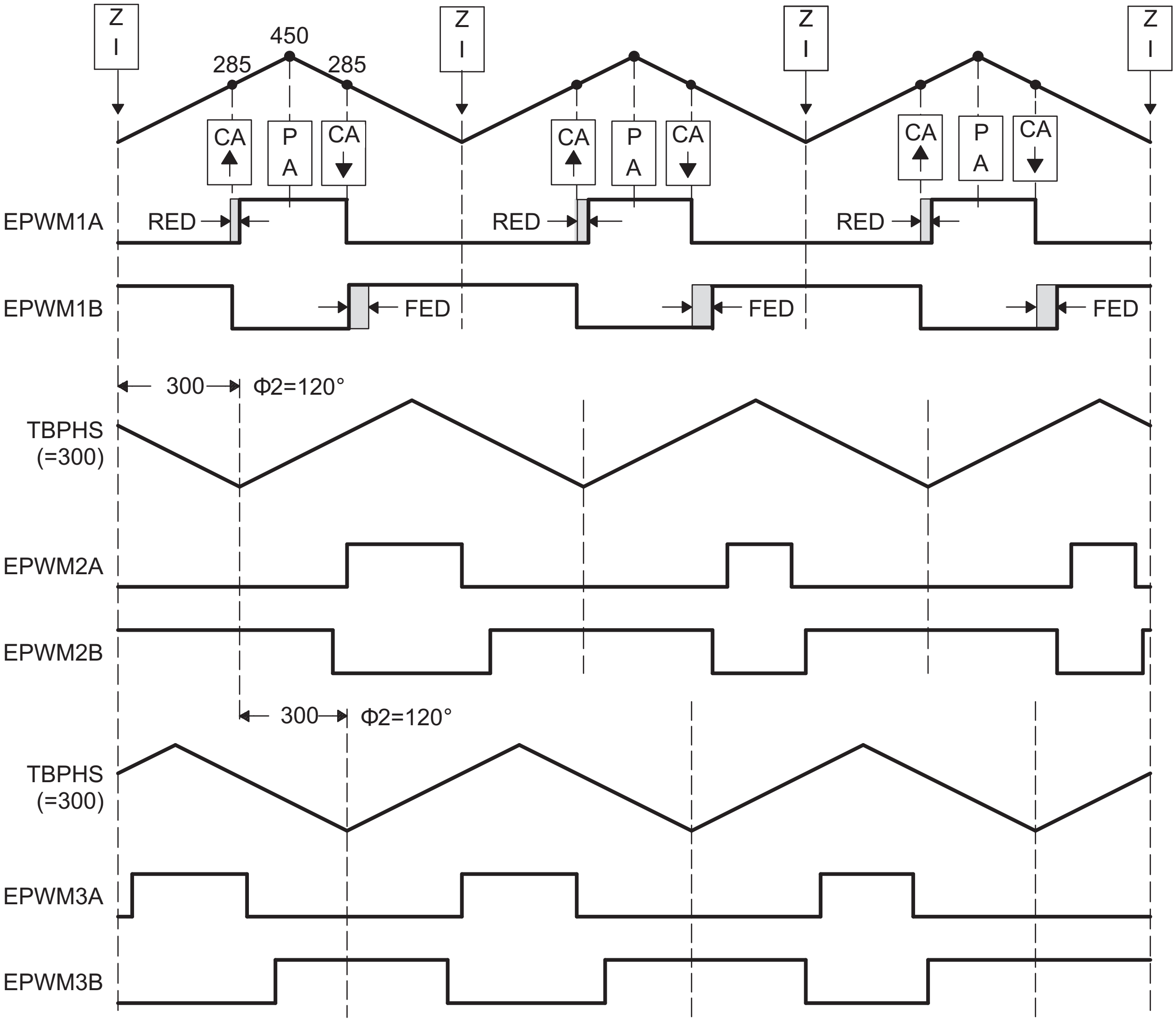


Table 20-53. EPWM1 Initialization for Figure 20-67

Register	Bit	Value	Comments
TBPRD	TBPRD	450 (1C2h)	Period = 900 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_CTR_ZERO	Sync down-stream module
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	Set actions for EPWM1A
	CAD	AQ_CLEAR	
DBCTL	MODE	DB_FULL_ENABLE	Enable Dead-band module
	POLSEL	DB_ACTV_HIC	Active Hi complementary
DBFED	DBFED	20	FED = 20 TBCLKs
	DBRED	20	RED = 20 TBCLKs

Table 20-54. EPWM2 Initialization for Figure 20-67

Register	Bit	Value	Comments
TBPRD	TBPRD	450 (1C2h)	Period = 900 TBCLK counts
TBPHS	TBPHS	300	Phase = (300/900) × 360 = 120°
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_ENABLE	Slave module
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_IN	Sync flow-through
	PHSDIR	TB_DOWN	Count DOWN on sync
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	Set actions for EPWM2A
	CAD	AQ_CLEAR	
DBCTL	MODE	DB_FULL_ENABLE	Enable Dead-band module
	POLSEL	DB_ACTV_HIC	Active Hi complementary
DBFED	DBFED	20	FED = 20 TBCLKs
	DBRED	20	RED = 20 TBCLKs

Table 20-55. EPWM3 Initialization for Figure 20-67

Register	Bit	Value	Comments
TBPRD	TBPRD	450 (1C2h)	Period = 900 TBCLK counts
TBPHS	TBPHS	300	Phase = (300/900) × 360 = 120°
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_ENABLE	Slave module
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_IN	Sync flow-through
	PHSDIR	TB_UP	Count UP on sync
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	Set actions for EPWM3A
	CAD	AQ_CLEAR	
DBCTL	MODE	DB_FULL_ENABLE	Enable Dead-band module
	POLSEL	DB_ACTV_HIC	Active Hi complementary
DBFED	DBFED	20	FED = 20 TBCLKs
	DBRED	20	RED = 20 TBCLKs

Example 20-7. Code Snippet for Configuration in Figure 20-67

```
// Run Time (Note: Example execution of one run-time instance)
//=========================================================
EPwm1Regs.CMPA.half.CMPA = 285;         // adjust duty for output EPWM1A
EPwm2Regs.CMPA.half.CMPA = 285;         // adjust duty for output EPWM2A
EPwm3Regs.CMPA.half.CMPA = 285;         // adjust duty for output EPWM3A
```

20.2.3.9 Controlling Zero Voltage Switched Full Bridge (ZVSFB) Converter

The example given in Figure 20-69 assumes a static or constant phase relationship between legs (modules). In such a case, control is achieved by modulating the duty cycle. It is also possible to dynamically change the phase value on a cycle-by-cycle basis. This feature lends itself to controlling a class of power topologies known as *phase-shifted full bridge*, or *zero voltage switched full bridge.* Here the controlled parameter is not duty cycle (this is kept constant at approximately 50 percent); instead it is the phase relationship between legs. Such a system can be implemented by allocating the resources of two PWM modules to control a single power stage, which in turn requires control of four switching elements. Figure 20-70 shows a master/slave module combination synchronized together to control a full H-bridge. In this case, both master and slave modules are required to switch at the same PWM frequency. The phase is controlled by using the slave's phase register (TBPHS). The master's phase register is not used and therefore can be initialized to zero.

Figure 20-69. Controlling a Full-H Bridge Stage ($F_{PWM2} = F_{PWM1}$)

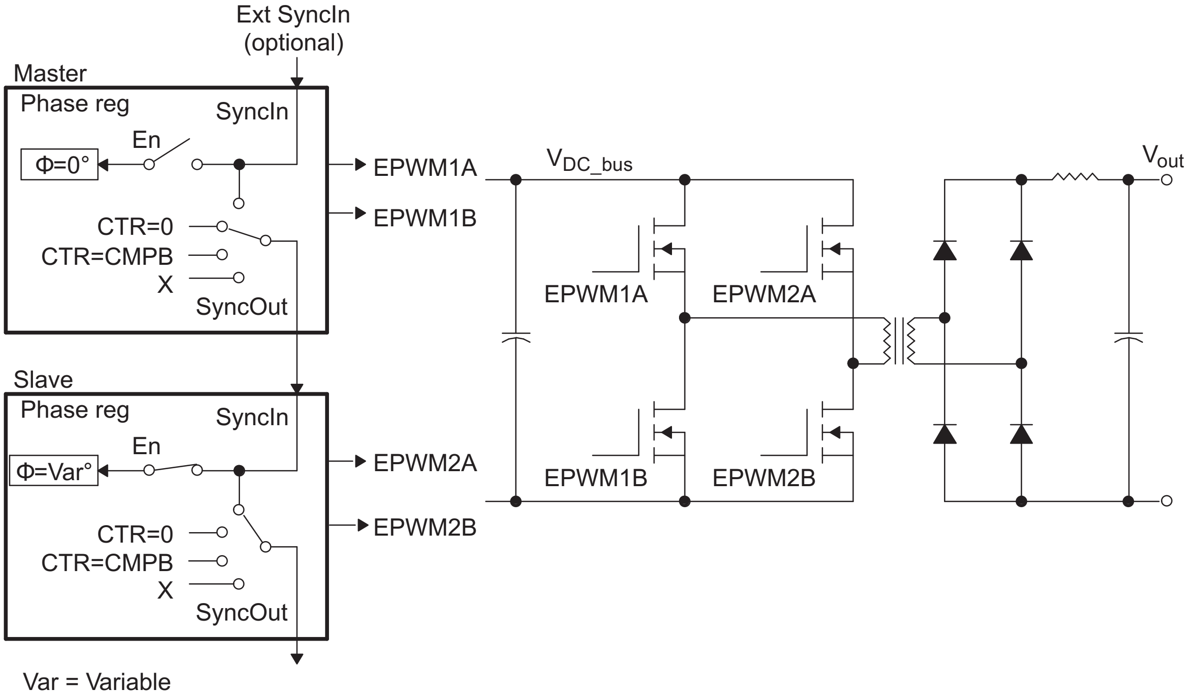


Figure 20-70. ZVS Full-H Bridge Waveforms

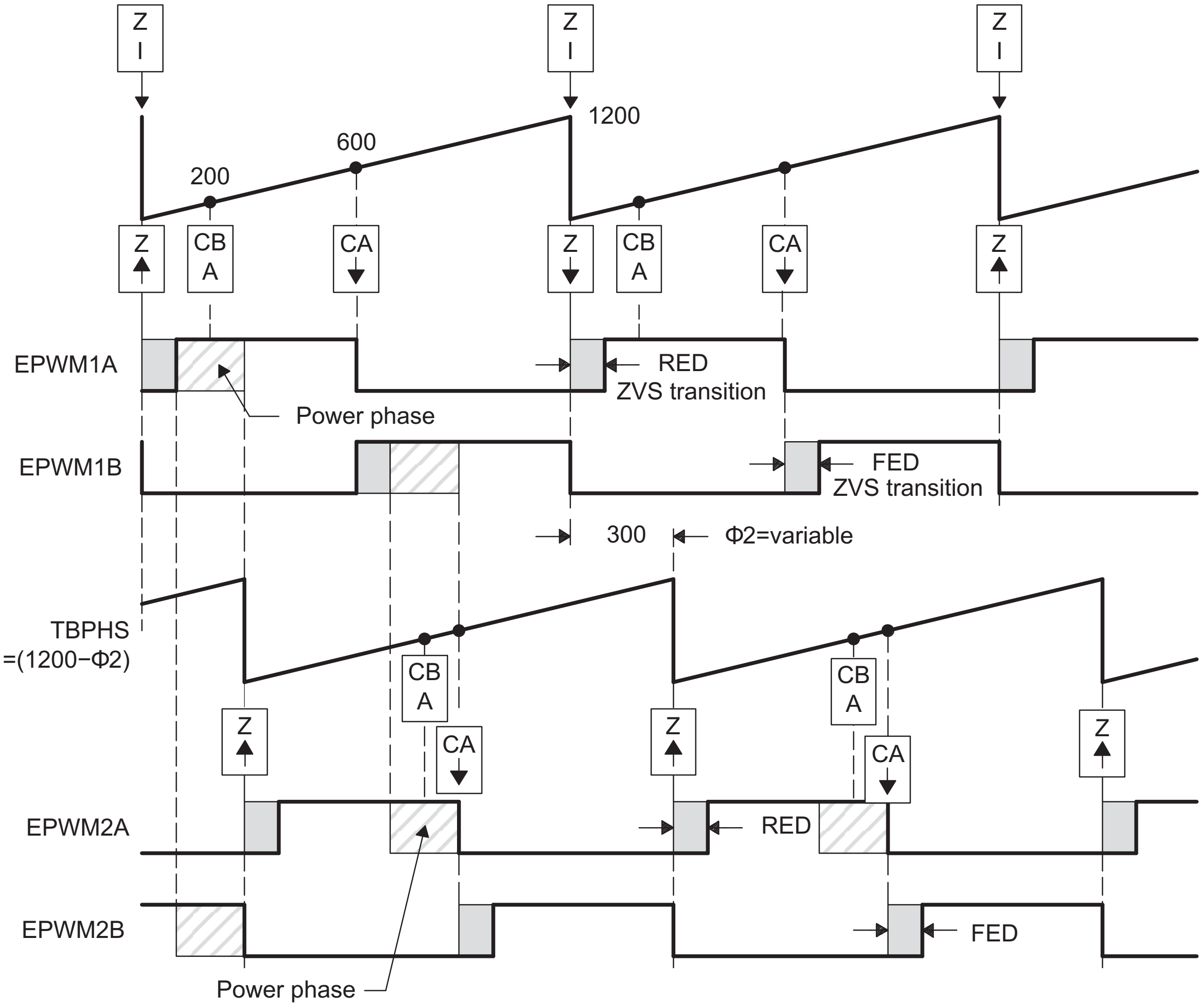


Table 20-56. EPWM1 Initialization for Figure 20-69

Register	Bit	Value	Comments
TBPRD	TBPRD	1200 (4B0h)	Period = 1201 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_CTR_ZERO	Sync down-stream module
CMPA	CMPA	600 (258h)	Set 50% duty for EPWM1A
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	ZRO	AQ_SET	Set actions for EPWM1A
	CAU	AQ_CLEAR	
DBCTL	MODE	DB_FULL_ENABLE	Enable Dead-band module
	POLSEL	DB_ACTV_HIC	Active Hi complementary
DBFED	DBFED	50	FED = 50 TBCLKs
	DBRED	70	RED = 70 TBCLKs

Table 20-57. EPWM2 Initialization for Figure 20-69

Register	Bit	Value	Comments
TBPRD	TBPRD	1200 (4B0h)	Period = 1201 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_ENABLE	Slave module
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_IN	Sync flow-through
CMPA	CMPA	600 (258h)	Set 50% duty for EPWM2A
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	ZRO	AQ_SET	Set actions for EPWM2A
	CAU	AQ_CLEAR	
DBCTL	MODE	DB_FULL_ENABLE	Enable Dead-band module
	POLSEL	DB_ACTV_HIC	Active Hi complementary
DBFED	DBFED	30	FED = 30 TBCLKs
	DBRED	40	RED = 40 TBCLKs

Example 20-8. Code Snippet for Configuration in Figure 20-69

```
// Run Time (Note: Example execution of one run-time instance)
//============================================================
EPwm2Regs.TBPHS = 1200-300;                        // Set Phase reg to 300/1200 * 360 = 90 deg
EPwm1Regs.DBFED = FED1_NewValue;                   // Update ZVS transition interval
EPwm1Regs.DBRED = RED1_NewValue;                   // Update ZVS transition interval
EPwm2Regs.DBFED = FED2_NewValue;                   // Update ZVS transition interval
```

Example 20-8. Code Snippet for Configuration in Figure 20-69 (continued)

```
EPwm2Regs.DBRED = RED2_NewValue;                    // Update ZVS transition interval
```

20.2.4 PWMSS_EPWM Registers

Table 20-58 lists the memory-mapped registers for the PWMSS_EPWM. All register offset addresses not listed in Table 20-58 should be considered as reserved locations and the register contents should not be modified.

Table 20-58. PWMSS_EPWM Registers

Offset	Acronym	Register Name	Section
0h	TBCTL	Time-Base Control Register	Section 20.2.4.1
2h	TBSTS	Time-Base Status Register	Section 20.2.4.2
4h	TBPHSHR	Extension for HRPWM Phase Register	Section 20.2.4.3
6h	TBPHS	Time-Base Phase Register	Section 20.2.4.4
8h	TBCNT	Time-Base Counter Register	Section 20.2.4.5
Ah	TBPRD	Time-Base Period Register	Section 20.2.4.6
Eh	CMPCTL	Counter-Compare Control Register	Section 20.2.4.7
10h	CMPAHR	Extension for HRPWM Counter-Compare A Register	Section 20.2.4.8
12h	CMPA	Counter-Compare A Register	Section 20.2.4.9
14h	CMPB	Counter-Compare B Register	Section 20.2.4.10
16h	AQCTLA	Action-Qualifier Control Register for Output A (EPWMxA)	Section 20.2.4.11
18h	AQCTLB	Action-Qualifier Control Register for Output B (EPWMxB)	Section 20.2.4.12
1Ah	AQSFRC	Action-Qualifier Software Force Register	Section 20.2.4.13
1Ch	AQCSFRC	Action-Qualifier Continuous S/W Force Register Set	Section 20.2.4.14
1Eh	DBCTL	Dead-Band Generator Control Register	Section 20.2.4.15
20h	DBRED	Dead-Band Generator Rising Edge Delay Count Register	Section 20.2.4.16
22h	DBFED	Dead-Band Generator Falling Edge Delay Count Register	Section 20.2.4.17
24h	TZSEL	Trip-Zone Select Register	Section 20.2.4.18
28h	TZCTL	Trip-Zone Control Register	Section 20.2.4.19
2Ah	TZEINT	Trip-Zone Enable Interrupt Register	Section 20.2.4.20
2Ch	TZFLG	Trip-Zone Flag Register	Section 20.2.4.21
2Eh	TZCLR	Trip-Zone Clear Register	Section 20.2.4.22
30h	TZFRC	Trip-Zone Force Register	Section 20.2.4.23
32h	ETSEL	Event-Trigger Selection Register	Section 20.2.4.24
34h	ETPS	Event-Trigger Pre-Scale Register	Section 20.2.4.25
36h	ETFLG	Event-Trigger Flag Register	Section 20.2.4.26
38h	ETCLR	Event-Trigger Clear Register	Section 20.2.4.27
3Ah	ETFRC	Event-Trigger Force Register	Section 20.2.4.28
3Ch	PCCTL	PWM-Chopper Control Register	Section 20.2.4.29
C0h	HRCTL	HRPWM Control Register	Section 20.2.4.30

20.2.4.1 TBCTL Register (Offset = 0h) [reset = 0h]

TBCTL is shown in Figure 20-71 and described in Table 20-59.

Return to Summary Table.

Figure 20-71. TBCTL Register

15	14	13	12	11	10	9	8
FREE_SOFT		PHSDIR	CLKDIV			HSPCLKDIV	
R/W-0h		R/W-0h	R/W-0h			R/W-0h	

7	6	5	4	3	2	1	0
HSPCLKDIV	SWFSYNC	SYNCOSEL		PRDLD	PHSEN	CTRMODE	
R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	

Table 20-59. TBCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	FREE_SOFT	R/W	0h	Emulation Mode Bits. These bits select the behavior of the ePWM time-base counter during emulation suspend events. Emulation debug events can be set up in the Debug Subsystem. 0h (R/W) = Stop after the next time-base counter increment or decrement 1h (R/W) = Stop when counter completes a whole cycle. (a) Up-count mode: stop when the time-base counter = period (TBCNT = TBPRD). (b) Down-count mode: stop when the time-base counter = 0000 (TBCNT = 0000h). (c) Up-down-count mode: stop when the time-base counter = 0000 (TBCNT = 0000h). 2h (R/W) = Free run 3h (R/W) = Free run
13	PHSDIR	R/W	0h	Phase Direction Bit. This bit is only used when the time-base counter is configured in the up-down-count mode. The PHSDIR bit indicates the direction the time-base counter (TBCNT) will count after a synchronization event occurs and a new phase value is loaded from the phase (TBPHS) register. This is irrespective of the direction of the counter before the synchronization event.. In the up-count and down-count modes this bit is ignored. 0h (R/W) = Count down after the synchronization event. 1h (R/W) = Count up after the synchronization event.
12-10	CLKDIV	R/W	0h	Time-base Clock Prescale Bits. These bits determine part of the time-base clock prescale value. TBCLK = SYSCLKOUT/(HSPCLKDIV * CLKDIV) 0h (R/W) = /1 (default on reset) 1h (R/W) = /2 2h (R/W) = /4 3h (R/W) = /8 4h (R/W) = /16 5h (R/W) = /32 6h (R/W) = /64 7h (R/W) = /128

Table 20-59. TBCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9-7	HSPCLKDIV	R/W	0h	High-Speed Time-base Clock Prescale Bits. These bits determine part of the time-base clock prescale value. TBCLK = SYSCLKOUT/(HSPCLKDIV * CLKDIV). This divisor emulates the HSPCLK in the TMS320x281x system as used on the Event Manager (EV) peripheral. 0h (R/W) = /1 1h (R/W) = /2 (default on reset) 2h (R/W) = /4 3h (R/W) = /6 4h (R/W) = /8 5h (R/W) = /10 6h (R/W) = /12 7h (R/W) = /14
6	SWFSYNC	R/W	0h	Software Forced Synchronization Pulse. 0h (R/W) = Writing a 0 has no effect and reads always return a 0. 1h (R/W) = Writing a 1 forces a one-time synchronization pulse to be generated. This event is ORed with the EPWMxSYNCI input of the ePWM module. SWFSYNC is valid (operates) only when EPWMxSYNCI is selected by SYNCOSEL = 00.
5-4	SYNCOSEL	R/W	0h	Synchronization Output Select. These bits select the source of the EPWMxSYNCO signal. 0h (R/W) = EPWMxSYNC: 1h (R/W) = CTR = 0: Time-base counter equal to zero (TBCNT = 0000h) 2h (R/W) = CTR = CMPB : Time-base counter equal to counter-compare B (TBCNT = CMPB) 3h (R/W) = Disable EPWMxSYNCO signal
3	PRDLD	R/W	0h	Active Period Register Load From Shadow Register Select 0h (R/W) = The period register (TBPRD) is loaded from its shadow register when the time-base counter, TBCNT, is equal to zero. A write or read to the TBPRD register accesses the shadow register. 1h (R/W) = Load the TBPRD register immediately without using a shadow register. A write or read to the TBPRD register directly accesses the active register.
2	PHSEN	R/W	0h	Counter Register Load From Phase Register Enable 0h (R/W) = Do not load the time-base counter (TBCNT) from the time-base phase register (TBPHS) 1h (R/W) = Load the time-base counter with the phase register when an EPWMxSYNCI input signal occurs or when a software synchronization is forced by the SWFSYNC bit.
1-0	CTRMODE	R/W	0h	Counter Mode. The time-base counter mode is normally configured once and not changed during normal operation. If you change the mode of the counter, the change will take effect at the next TBCLK edge and the current counter value shall increment or decrement from the value before the mode change. These bits set the time-base counter mode of operation as follows: 0h (R/W) = Up-count mode 1h (R/W) = Down-count mode 2h (R/W) = Up-down-count mode 3h (R/W) = Stop-freeze counter operation (default on reset)

20.2.4.2 TBSTS Register (Offset = 2h) [reset = 0h]

TBSTS is shown in Figure 20-72 and described in Table 20-60.

Return to Summary Table.

Figure 20-72. TBSTS Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					CTRMAX	SYNCI	CTRDIR
R-0h					0h	W1C-0h	R-0h

Table 20-60. TBSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-3	RESERVED	R	0h	
2	CTRMAX		0h	Time-Base Counter Max Latched Status Bit. 0h (R/W) = Reading a 0 indicates the time-base counter never reached its maximum value. Writing a 0 will have no effect. 1h (R/W) = Reading a 1 on this bit indicates that the time-base counter reached the max value 0xFFFF. Writing a 1 to this bit will clear the latched event.
1	SYNCI	W1C	0h	Input Synchronization Latched Status Bit. 0h (R/W) = Writing a 0 will have no effect. Reading a 0 indicates no external synchronization event has occurred. 1h (R/W) = Reading a 1 on this bit indicates that an external synchronization event has occurred (EPWMxSYNCI). Writing a 1 to this bit will clear the latched event.
0	CTRDIR	R	0h	Time-Base Counter Direction Status Bit. At reset, the counter is frozen, therefore, this bit has no meaning. To make this bit meaningful, you must first set the appropriate mode via TBCTL[CTRMODE]. 0h (R/W) = Time-Base Counter is currently counting down. 1h (R/W) = Time-Base Counter is currently counting up.

20.2.4.3 TBPHSHR Register (Offset = 4h) [reset = 0h]

TBPHSHR is shown in Figure 20-73 and described in Table 20-61.

Return to Summary Table.

Figure 20-73. TBPHSHR Register

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">TBPHSH</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
</table>

Table 20-61. TBPHSHR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	TBPHSH	R/W	0h	Time-base phase high-resolution bits
7-0	RESERVED	R	0h	

20.2.4.4 TBPHS Register (Offset = 6h) [reset = 0h]

TBPHS is shown in Figure 20-74 and described in Table 20-62.

Return to Summary Table.

This register is only available on ePWM instances that include the high-resolution PWM (HRPWM) extension, otherwise, this location is reserved.

Figure 20-74. TBPHS Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBPHS															
R/W-0h															

Table 20-62. TBPHS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	TBPHS	R/W	0h	These bits set time-base counter phase of the selected ePWM relative to the time-base that is supplying the synchronization input signal. (a) If TBCTL[PHSEN] = 0, then the synchronization event is ignored and the time-base counter is not loaded with the phase. (b) If TBCTL[PHSEN] = 1, then the time-base counter (TBCNT) will be loaded with the phase (TBPHS) when a synchronization event occurs. The synchronization event can be initiated by the input synchronization signal (EPWMxSYNCI) or by a software forced synchronization.

20.2.4.5 TBCNT Register (Offset = 8h) [reset = 0h]

TBCNT is shown in Figure 20-75 and described in Table 20-63.

Return to Summary Table.

Figure 20-75. TBCNT Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBCNT															
R/W-0h															

Table 20-63. TBCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	TBCNT	R/W	0h	Reading these bits gives the current time-base counter value. Writing to these bits sets the current time-base counter value. The update happens as soon as the write occurs. The write is NOT synchronized to the time-base clock (TBCLK) and the register is not shadowed.

20.2.4.6 TBPRD Register (Offset = Ah) [reset = 0h]

TBPRD is shown in Figure 20-76 and described in Table 20-64.

Return to Summary Table.

Figure 20-76. TBPRD Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBPRD															
R/W-0h															

Table 20-64. TBPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	TBPRD	R/W	0h	These bits determine the period of the time-base counter. This sets the PWM frequency. Shadowing of this register is enabled and disabled by the TBCTL[PRDLD] bit. By default this register is shadowed. (a) If TBCTL[PRDLD] = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the active register will be loaded from the shadow register when the time-base counter equals zero. (b) If TBCTL[PRDLD] = 1, then the shadow is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware. (c) The active and shadow registers share the same memory map address.

20.2.4.7 CMPCTL Register (Offset = Eh) [reset = 0h]

CMPCTL is shown in Figure 20-77 and described in Table 20-65.

Return to Summary Table.

Figure 20-77. CMPCTL Register

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="6">RESERVED</td><td>SHDWBFULL</td><td>SHDWAFULL</td></tr>
<tr><td colspan="6">R-0h</td><td>R-0h</td><td>R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>RESERVED</td><td>SHDWBMODE</td><td>RESERVED</td><td>SHDWAMODE</td><td colspan="2">LOADBMODE</td><td colspan="2">LOADAMODE</td></tr>
<tr><td>R-0h</td><td>R/W-0h</td><td>R-0h</td><td>R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 20-65. CMPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	
9	SHDWBFULL	R	0h	Counter-compare B (CMPB) Shadow Register Full Status Flag. This bit self clears once a load-strobe occurs. 0h (R/W) = CMPB shadow FIFO not full yet 1h (R/W) = Indicates the CMPB shadow FIFO is full. A CPU write will overwrite current shadow value.
8	SHDWAFULL	R	0h	Counter-compare A (CMPA) Shadow Register Full Status Flag. The flag bit is set when a 32 bit write to CMPA:CMPAHR register or a 16 bit write to CMPA register is made. A 16 bit write to CMPAHR register will not affect the flag. This bit self clears once a load-strobe occurs. 0h (R/W) = CMPA shadow FIFO not full yet 1h (R/W) = Indicates the CMPA shadow FIFO is full, a CPU write will overwrite the current shadow value.
7	RESERVED	R	0h	
6	SHDWBMODE	R/W	0h	Counter-compare B (CMPB) Register Operating Mode. 0h (R/W) = Shadow mode. Operates as a double buffer. All writes via the CPU access the shadow register. 1h (R/W) = Immediate mode. Only the active compare B register is used. All writes and reads directly access the active register for immediate compare action.
5	RESERVED	R	0h	
4	SHDWAMODE	R/W	0h	Counter-compare A (CMPA) Register Operating Mode. 0h (R/W) = Shadow mode. Operates as a double buffer. All writes via the CPU access the shadow register. 1h (R/W) = Immediate mode. Only the active compare register is used. All writes and reads directly access the active register for immediate compare action
3-2	LOADBMODE	R/W	0h	Active Counter-Compare B (CMPB) Load From Shadow Select Mode. This bit has no effect in immediate mode (CMPCTL[SHDWBMODE] = 1). 0h (R/W) = Load on CTR = 0: Time-base counter equal to zero (TBCNT = 0000h) 1h (R/W) = Load on CTR = PRD: Time-base counter equal to period (TBCNT = TBPRD) 2h (R/W) = Load on either CTR = 0 or CTR = PRD 3h (R/W) = Freeze (no loads possible)

Table 20-65. CMPCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	LOADAMODE	R/W	0h	Active Counter-Compare A (CMPA) Load From Shadow Select Mode. This bit has no effect in immediate mode (CMPCTL[SHDWAMODE] = 1). 0h (R/W) = Load on CTR = 0: Time-base counter equal to zero (TBCNT = 0000h) 1h (R/W) = Load on CTR = PRD: Time-base counter equal to period (TBCNT = TBPRD) 2h (R/W) = Load on either CTR = 0 or CTR = PRD 3h (R/W) = Freeze (no loads possible)

20.2.4.8 CMPAHR Register (Offset = 10h) [reset = 100h]

CMPAHR is shown in Figure 20-78 and described in Table 20-66.

Return to Summary Table.

This register is only available on ePWM instances that include the high-resolution PWM (HRPWM) extension; otherwise, this location is reserved.

Figure 20-78. CMPAHR Register

15	14	13	12	11	10	9	8
CMPAHR							
R/W-1h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							
R-0h							

Table 20-66. CMPAHR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	CMPAHR	R/W	1h	Compare A High-Resolution register bits for MEP step control. A minimum value of 1h is needed to enable HRPWM capabilities. Valid MEP range of operation 1-255h.
7-0	RESERVED	R	0h	

20.2.4.9 CMPA Register (Offset = 12h) [reset = 0h]

CMPA is shown in Figure 20-79 and described in Table 20-67.

Return to Summary Table.

Figure 20-79. CMPA Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPA															
R/W-0h															

Table 20-67. CMPA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	CMPA	R/W	0h	The value in the active CMPA register is continuously compared to the time-base counter (TBCNT). When the values are equal, the counter-compare module generates a "time-base counter equal to counter compare A" event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the EPWMxA or the EPWMxB output depending on the configuration of the AQCTLA and AQCTLB registers. The actions that can be defined in the AQCTLA and AQCTLB registers include the following. (a) Do nothing the event is ignored. (b) Clear: Pull the EPWMxA and/or EPWMxB signal low. (c) Set: Pull the EPWMxA and/or EPWMxB signal high. (d) Toggle the EPWMxA and/or EPWMxB signal. Shadowing of this register is enabled and disabled by the CMPCTL[SHDWAMODE] bit. By default this register is shadowed. (a) If CMPCTL[SHDWAMODE] = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the CMPCTL[LOADAMODE] bit field determines which event will load the active register from the shadow register. (b) Before a write, the CMPCTL[SHDWAFULL] bit can be read to determine if the shadow register is currently full. (c) If CMPCTL[SHDWAMODE] = 1, then the shadow register is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware. (d) In either mode, the active and shadow registers share the same memory map address.

20.2.4.10 CMPB Register (Offset = 14h) [reset = 0h]

CMPB is shown in Figure 20-80 and described in Table 20-68.

Return to Summary Table.

Figure 20-80. CMPB Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPB															
R/W-0h															

Table 20-68. CMPB Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	CMPB	R/W	0h	The value in the active CMPB register is continuously compared to the time-base counter (TBCNT). When the values are equal, the counter-compare module generates a "time-base counter equal to counter compare B" event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the EPWMxA or the EPWMxB output depending on the configuration of the AQCTLA and AQCTLB registers. The actions that can be defined in the AQCTLA and AQCTLB registers include the following. (a) Do nothing, the event is ignored. (b) Clear: Pull the EPWMxA and/or EPWMxB signal low. (c) Set: Pull the EPWMxA and/or EPWMxB signal high. (d) Toggle the EPWMxA and/or EPWMxB signal. Shadowing of this register is enabled and disabled by the CMPCTL[SHDWBMODE] bit. By default this register is shadowed. (a) If CMPCTL[SHDWBMODE] = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the CMPCTL[LOADBMODE] bit field determines which event will load the active register from the shadow register: (b) Before a write, the CMPCTL[SHDWBFULL] bit can be read to determine if the shadow register is currently full. (c) If CMPCTL[SHDWBMODE] = 1, then the shadow register is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware. (d) In either mode, the active and shadow registers share the same memory map address.

20.2.4.11 AQCTLA Register (Offset = 16h) [reset = 0h]

AQCTLA is shown in Figure 20-81 and described in Table 20-69.

Return to Summary Table.

Figure 20-81. AQCTLA Register

15	14	13	12	11	10	9	8
RESERVED				CBD		CBU	
R-0h				R/W-0h		R/W-0h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CAD		CAU		PRD		ZRO	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 20-69. AQCTLA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	
11-10	CBD	R/W	0h	Action when the time-base counter equals the active CMPB register and the counter is decrementing. 0h (R/W) = Do nothing (action disabled) 1h (R/W) = Clear - force EPWMxA output low. 2h (R/W) = Set - force EPWMxA output high. 3h (R/W) = Toggle EPWMxA output - low output signal will be forced high, and a high signal will be forced low.
9-8	CBU	R/W	0h	Action when the counter equals the active CMPB register and the counter is incrementing. 0h (R/W) = Do nothing (action disabled) 1h (R/W) = Clear - force EPWMxA output low. 2h (R/W) = Set - force EPWMxA output high. 3h (R/W) = Toggle EPWMxA output - low output signal will be forced high, and a high signal will be forced low.
7-6	CAD	R/W	0h	Action when the counter equals the active CMPA register and the counter is decrementing. 0h (R/W) = Do nothing (action disabled) 1h (R/W) = Clear - force EPWMxA output low. 2h (R/W) = Set - force EPWMxA output high. 3h (R/W) = Toggle EPWMxA output - low output signal will be forced high, and a high signal will be forced low.
5-4	CAU	R/W	0h	Action when the counter equals the active CMPA register and the counter is incrementing. 0h (R/W) = Do nothing (action disabled) 1h (R/W) = Clear - force EPWMxA output low. 2h (R/W) = Set - force EPWMxA output high. 3h (R/W) = Toggle EPWMxA output - low output signal will be forced high, and a high signal will be forced low.
3-2	PRD	R/W	0h	Action when the counter equals the period. Note: By definition, in count up-down mode when the counter equals period the direction is defined as 0 or counting down. 0h (R/W) = Do nothing (action disabled) 1h (R/W) = Clear - force EPWMxA output low. 2h (R/W) = Set - force EPWMxA output high. 3h (R/W) = Toggle EPWMxA output - low output signal will be forced high, and a high signal will be forced low.

Table 20-69. AQCTLA Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	ZRO	R/W	0h	Action when counter equals zero. Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 0h (R/W) = Do nothing (action disabled) 1h (R/W) = Clear - force EPWMxA output low. 2h (R/W) = Set - force EPWMxA output high. 3h (R/W) = Toggle EPWMxA output - low output signal will be forced high, and a high signal will be forced low.

20.2.4.12 AQCTLB Register (Offset = 18h) [reset = 0h]

AQCTLB is shown in Figure 20-82 and described in Table 20-70.

Return to Summary Table.

Figure 20-82. AQCTLB Register

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="2">CBD</td><td colspan="2">CBU</td></tr>
<tr><td colspan="4">R-0h</td><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CAD</td><td colspan="2">CAU</td><td colspan="2">PRD</td><td colspan="2">ZRO</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 20-70. AQCTLB Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	
11-10	CBD	R/W	0h	Action when the counter equals the active CMPB register and the counter is decrementing. 0h (R/W) = Do nothing (action disabled) 1h (R/W)(Read) = Clearforce EPWMxB output low. 2h (R/W) = Set: force EPWMxB output high. 3h (R/W) = Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low.
9-8	CBU	R/W	0h	Action when the counter equals the active CMPB register and the counter is incrementing. 0h (R/W) = Do nothing (action disabled) 1h (R/W)(Read) = Clearforce EPWMxB output low. 2h (R/W) = Set: force EPWMxB output high. 3h (R/W) = Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low.
7-6	CAD	R/W	0h	Action when the counter equals the active CMPA register and the counter is decrementing. 0h (R/W) = Do nothing (action disabled) 1h (R/W)(Read) = Clearforce EPWMxB output low. 2h (R/W) = Set: force EPWMxB output high. 3h (R/W) = Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low.
5-4	CAU	R/W	0h	Action when the counter equals the active CMPA register and the counter is incrementing. 0h (R/W) = Do nothing (action disabled) 1h (R/W)(Read) = Clearforce EPWMxB output low. 2h (R/W) = Set: force EPWMxB output high. 3h (R/W) = Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low.
3-2	PRD	R/W	0h	Action when the counter equals the period. Note: By definition, in count up-down mode when the counter equals period the direction is defined as 0 or counting down. 0h (R/W) = Do nothing (action disabled) 1h (R/W)(Read) = Clearforce EPWMxB output low. 2h (R/W) = Set: force EPWMxB output high. 3h (R/W) = Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low.

Table 20-70. AQCTLB Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	ZRO	R/W	0h	Action when counter equals zero. Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 0h (R/W) = Do nothing (action disabled) 1h (R/W)(Read) = Cleaforce EPWMxB output low. 2h (R/W) = Set: force EPWMxB output high. 3h (R/W) = Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low.

20.2.4.13 AQSFRC Register (Offset = 1Ah) [reset = 0h]

AQSFRC is shown in Figure 20-83 and described in Table 20-71.

Return to Summary Table.

Figure 20-83. AQSFRC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RLDCSF		OTSFB	ACTSFB		OTSFA	ACTSFA	
R/W-0h		R/W-0h	R/W-0h		R/W-0h	R/W-0h	

Table 20-71. AQSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-6	RLDCSF	R/W	0h	AQCSFRC Active Register Reload From Shadow Options. 0h (R/W) = Load on event counter equals zero 1h (R/W) = Load on event counter equals period 2h (R/W) = Load on event counter equals zero or counter equals period 3h (R/W) = Load immediately (the active register is directly accessed by the CPU and is not loaded from the shadow register).
5	OTSFB	R/W	0h	One-Time Software Forced Event on Output B. 0h (R/W) = Writing a 0 (zero) has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete, that is, a forced event is initiated. This is a one-shot forced event. It can be overridden by another subsequent event on output B. 1h (R/W) = Initiates a single s/w forced event
4-3	ACTSFB	R/W	0h	Action when One-Time Software Force B Is invoked 0h (R/W) = Does nothing (action disabled) 1h (R/W) = Clear (low) 2h (R/W) = Set (high) 3h (R/W) = Toggle (Low -> High, High -> Low). Note: This action is not qualified by counter direction (CNT_dir)
2	OTSFA	R/W	0h	One-Time Software Forced Event on Output A. 0h (R/W) = Writing a 0 (zero) has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (that is, a forced event is initiated). 1h (R/W) = Initiates a single software forced event.
1-0	ACTSFA	R/W	0h	Action When One-Time Software Force A Is Invoked. 0h (R/W) = Does nothing (action disabled). 1h (R/W) = Clear (low). 2h (R/W) = Set (high). 3h (R/W) = Toggle (Low -> High, High -> Low). Note: This action is not qualified by counter direction (CNT_dir)

20.2.4.14 AQCSFRC Register (Offset = 1Ch) [reset = 0h]

AQCSFRC is shown in Figure 20-84 and described in Table 20-72.

Return to Summary Table.

Figure 20-84. AQCSFRC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CSFB		CSFA	
R-0h				R/W-0h		R/W-0h	

Table 20-72. AQCSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	
3-2	CSFB	R/W	0h	Continuous Software Force on Output B. In immediate mode, a continuous force takes effect on the next TBCLK edge. In shadow mode, a continuous force takes effect on the next TBCLK edge after a shadow load into the active register. To configure shadow mode, use AQSFRC[RLDCSF]. 0h (R/W) = Forcing disabled, that is, has no effect 1h (R/W) = Forces a continuous low on output B 2h (R/W) = Forces a continuous high on output B 3h (R/W) = Software forcing is disabled and has no effect
1-0	CSFA	R/W	0h	Continuous Software Force on Output A In immediate mode, a continuous force takes effect on the next TBCLK edge. In shadow mode, a continuous force takes effect on the next TBCLK edge after a shadow load into the active register. 0h (R/W) = Forcing disabled, that is, has no effect 1h (R/W) = Forces a continuous low on output A 2h (R/W) = Forces a continuous high on output A 3h (R/W) = Software forcing is disabled and has no effect

20.2.4.15 DBCTL Register (Offset = 1Eh) [reset = 0h]

DBCTL is shown in Figure 20-85 and described in Table 20-73.

Return to Summary Table.

Figure 20-85. DBCTL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED		IN_MODE		POLSEL		OUT_MODE	
R-0h		R/W-0h		R/W-0h		R/W-0h	

Table 20-73. DBCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-6	RESERVED	R	0h	
5-4	IN_MODE	R/W	0h	Dead Band Input Mode Control. Bit 5 controls the S5 switch and bit 4 controls the S4 switch. This allows you to select the input source to the falling-edge and rising-edge delay. To produce classical dead-band waveforms, the default is EPWMxA In is the source for both falling and rising-edge delays. 0h (R/W) = EPWMxA In (from the action-qualifier) is the source for both falling-edge and rising-edge delay. 1h (R/W) = EPWMxB In (from the action-qualifier) is the source for rising-edge delayed signal. EPWMxA In (from the action-qualifier) is the source for falling-edge delayed signal. 2h (R/W) = EPWMxA In (from the action-qualifier) is the source for rising-edge delayed signal. EPWMxB In (from the action-qualifier) is the source for falling-edge delayed signal. 3h (R/W) = EPWMxB In (from the action-qualifier) is the source for both rising-edge delay and falling-edge delayed signal.
3-2	POLSEL	R/W	0h	Polarity Select Control. Bit 3 controls the S3 switch and bit 2 controls the S2 switch. This allows you to selectively invert one of the delayed signals before it is sent out of the dead-band submodule. The following descriptions correspond to classical upper/lower switch control as found in one leg of a digital motor control inverter. These assume that DBCTL[OUT_MODE] = 1,1 and DBCTL[IN_MODE] = 0,0. Other enhanced modes are also possible, but not regarded as typical usage modes. 0h (R/W) = Active high (AH) mode. Neither EPWMxA nor EPWMxB is inverted (default). 1h (R/W) = Active low complementary (ALC) mode. EPWMxA is inverted. 2h (R/W) = Active high complementary (AHC). EPWMxB is inverted. 3h (R/W) = Active low (AL) mode. Both EPWMxA and EPWMxB are inverted.

Table 20-73. DBCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	OUT_MODE	R/W	0h	Dead-band Output Mode Control. Bit 1 controls the S1 switch and bit 0 controls the S0 switch. This allows you to selectively enable or bypass the dead-band generation for the falling-edge and rising-edge delay. 0h (R/W) = Dead-band generation is bypassed for both output signals. In this mode, both the EPWMxA and EPWMxB output signals from the action-qualifier are passed directly to the PWM-chopper submodule. In this mode, the POLSEL and IN_MODE bits have no effect. 1h (R/W) = Disable rising-edge delay. The EPWMxA signal from the action-qualifier is passed straight through to the EPWMxA input of the PWM-chopper submodule. The falling-edge delayed signal is seen on output EPWMxB. The input signal for the delay is determined by DBCTL[IN_MODE]. 2h (R/W) = Disable falling-edge delay. The EPWMxB signal from the action-qualifier is passed straight through to the EPWMxB input of the PWM-chopper submodule. The rising-edge delayed signal is seen on output EPWMxA. The input signal for the delay is determined by DBCTL[IN_MODE]. 3h (R/W) = Dead-band is fully enabled for both rising-edge delay on output EPWMxA and falling-edge delay on output EPWMxB. The input signal for the delay is determined by DBCTL[IN_MODE].

20.2.4.16 DBRED Register (Offset = 20h) [reset = 0h]

DBRED is shown in Figure 20-86 and described in Table 20-74.

Return to Summary Table.

Figure 20-86. DBRED Register

15	14	13	12	11	10	9	8
RESERVED						DEL	
R-0h						R/W-0h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DEL							
R/W-0h							

Table 20-74. DBRED Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	
9-0	DEL	R/W	0h	Rising Edge Delay Count. 10 bit counter.

20.2.4.17 DBFED Register (Offset = 22h) [reset = 0h]

DBFED is shown in Figure 20-87 and described in Table 20-75.

Return to Summary Table.

Figure 20-87. DBFED Register

15	14	13	12	11	10	9	8
RESERVED						DEL	
R-0h						R/W-0h	

7	6	5	4	3	2	1	0
DEL							
R/W-0h							

Table 20-75. DBFED Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	
9-0	DEL	R/W	0h	Falling Edge Delay Count. 10 bit counter

20.2.4.18 TZSEL Register (Offset = 24h) [reset = 0h]

TZSEL is shown in Figure 20-88 and described in Table 20-76.

Return to Summary Table.

Figure 20-88. TZSEL Register

15	14	13	12	11	10	9	8
OSHTn							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CBCn							
R/W-0h							

Table 20-76. TZSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	OSHTn	R/W	0h	Trip-zone n (TZn) select. One-Shot (OSHT) trip-zone enable/disable. When any of the enabled pins go low, a one-shot trip event occurs for this ePWM module. When the event occurs, the action defined in the TZCTL register is taken on the EPWMxA and EPWMxB outputs. The one-shot trip condition remains latched until you clear the condition via the TZCLR register. 0h (R/W) = Disable TZn as a one-shot trip source for this ePWM module. 1h (R/W) = Enable TZn as a one-shot trip source for this ePWM module.
7-0	CBCn	R/W	0h	Trip-zone n (TZn) select. Cycle-by-Cycle (CBC) trip-zone enable/disable. When any of the enabled pins go low, a cycle-by-cycle trip event occurs for this ePWM module. When the event occurs, the action defined in the TZCTL register is taken on the EPWMxA and EPWMxB outputs. A cycle-by-cycle trip condition is automatically cleared when the time-base counter reaches zero. 0h (R/W) = Disable TZn as a CBC trip source for this ePWM module. 1h (R/W) = Enable TZn as a CBC trip source for this ePWM module.

20.2.4.19 TZCTL Register (Offset = 28h) [reset = 0h]

TZCTL is shown in Figure 20-89 and described in Table 20-77.

Return to Summary Table.

Figure 20-89. TZCTL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				TZB		TZA	
R-0h				R/W-0h		R/W-0h	

Table 20-77. TZCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	
3-2	TZB	R/W	0h	When a trip event occurs the following action is taken on output EPWMxB. Which trip-zone pins can cause an event is defined in the TZSEL register. 0h (R/W) = High impedance (EPWMxB = High-impedance state) 1h (R/W) = Force EPWMxB to a high state 2h (R/W) = Force EPWMxB to a low state 3h (R/W) = Do nothing, no action is taken on EPWMxB.
1-0	TZA	R/W	0h	When a trip event occurs the following action is taken on output EPWMxA. Which trip-zone pins can cause an event is defined in the TZSEL register. 0h (R/W) = High impedance (EPWMxA = High-impedance state) 1h (R/W) = Force EPWMxA to a high state 2h (R/W) = Force EPWMxA to a low state 3h (R/W) = Do nothing, no action is taken on EPWMxA.

20.2.4.20 TZEINT Register (Offset = 2Ah) [reset = 0h]

TZEINT is shown in Figure 20-90 and described in Table 20-78.

Return to Summary Table.

Figure 20-90. TZEINT Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED					OST	CBC	RESERVED
R-0h					R/W-0h	R/W-0h	R-0h

Table 20-78. TZEINT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-3	RESERVED	R	0h	
2	OST	R/W	0h	Trip-zone One-Shot Interrupt Enable 0h (R/W) = Disable one-shot interrupt generation 1h (R/W) = Enable Interrupt generation; a one-shot trip event will cause a EPWMxTZINT interrupt.
1	CBC	R/W	0h	Trip-zone Cycle-by-Cycle Interrupt Enable 0h (R/W) = Disable cycle-by-cycle interrupt generation. 1h (R/W) = Enable interrupt generation; a cycle-by-cycle trip event will cause an EPWMxTZINT interrupt.
0	RESERVED	R	0h	

20.2.4.21 TZFLG Register (Offset = 2Ch) [reset = 0h]

TZFLG is shown in Figure 20-91 and described in Table 20-79.

Return to Summary Table.

Figure 20-91. TZFLG Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED					OST	CBC	INT
R-0h					R-0h	R-0h	R-0h

Table 20-79. TZFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-3	RESERVED	R	0h	
2	OST	R	0h	Latched Status Flag for A One-Shot Trip Event. 0h (R/W) = No one-shot trip event has occurred. 1h (R/W) = Indicates a trip event has occurred on a pin selected as a one-shot trip source. This bit is cleared by writing the appropriate value to the TZCLR register.
1	CBC	R	0h	Latched Status Flag for Cycle-By-Cycle Trip Event 0h (R/W) = No cycle-by-cycle trip event has occurred. 1h (R/W) = Indicates a trip event has occurred on a pin selected as a cycle-by-cycle trip source. The TZFLG[CBC] bit will remain set until it is manually cleared by the user. If the cycle-by-cycle trip event is still present when the CBC bit is cleared, then CBC will be immediately set again. The specified condition on the pins is automatically cleared when the ePWM time-base counter reaches zero (TBCNT = 0000h) if the trip condition is no longer present. The condition on the pins is only cleared when the TBCNT = 0000h no matter where in the cycle the CBC flag is cleared. This bit is cleared by writing the appropriate value to the TZCLR register.
0	INT	R	0h	Latched Trip Interrupt Status Flag 0h (R/W) = Indicates no interrupt has been generated. 1h (R/W) = Indicates an EPWMxTZINT interrupt was generated because of a trip condition. No further EPWMxTZINT interrupts will be generated until this flag is cleared. If the interrupt flag is cleared when either CBC or OST is set, then another interrupt pulse will be generated. Clearing all flag bits will prevent further interrupts. This bit is cleared by writing the appropriate value to the TZCLR register.

20.2.4.22 TZCLR Register (Offset = 2Eh) [reset = 0h]

TZCLR is shown in Figure 20-92 and described in Table 20-80.

Return to Summary Table.

Figure 20-92. TZCLR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					OST	CBC	INT
R-0h					R/W-0h	R/W-0h	R/W-0h

Table 20-80. TZCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-3	RESERVED	R	0h	
2	OST	R/W	0h	Clear Flag for One-Shot Trip (OST) Latch 0h (R/W) = Has no effect. Always reads back a 0. 1h (R/W) = Clears this Trip (set) condition.
1	CBC	R/W	0h	Clear Flag for Cycle-By-Cycle (CBC) Trip Latch 0h (R/W) = Has no effect. Always reads back a 0. 1h (R/W) = Clears this Trip (set) condition.
0	INT	R/W	0h	Global Interrupt Clear Flag 0h (R/W) = Has no effect. Always reads back a 0. 1h (R/W) = Clears the trip-interrupt flag for this ePWM module (TZFLG[INT]). Note: No further EPWMxTZINT interrupts will be generated until the flag is cleared. If the TZFLG[INT] bit is cleared and any of the other flag bits are set, then another interrupt pulse will be generated. Clearing all flag bits will prevent further interrupts.

20.2.4.23 TZFRC Register (Offset = 30h) [reset = 0h]

TZFRC is shown in Figure 20-93 and described in Table 20-81.

Return to Summary Table.

Figure 20-93. TZFRC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					OST	CBC	RESERVED
R-0h					R/W-0h	R/W-0h	R-0h

Table 20-81. TZFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-3	RESERVED	R	0h	
2	OST	R/W	0h	Force a One-Shot Trip Event via Software 0h (R/W) = Writing of 0 is ignored. Always reads back a 0. 1h (R/W) = Forces a one-shot trip event and sets the TZFLG[OST] bit.
1	CBC	R/W	0h	Force a Cycle-by-Cycle Trip Event via Software 0h (R/W) = Writing of 0 is ignored. Always reads back a 0. 1h (R/W) = Forces a cycle-by-cycle trip event and sets the TZFLG[CBC] bit.
0	RESERVED	R	0h	

20.2.4.24 ETSEL Register (Offset = 32h) [reset = 0h]

ETSEL is shown in Figure 20-94 and described in Table 20-82.

Return to Summary Table.

Figure 20-94. ETSEL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				INTEN	INTSEL		
R-0h				R/W-0h	R/W-0h		

Table 20-82. ETSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	
3	INTEN	R/W	0h	Enable ePWM Interrupt (EPWMx_INT) Generation 0h (R/W) = Disable EPWMx_INT generation 1h (R/W) = Enable EPWMx_INT generation
2-0	INTSEL	R/W	0h	ePWM Interrupt (EPWMx_INT) Selection Options 0h (R/W) = Reserved 1h (R/W) = Enable event time-base counter equal to zero. (TBCNT = 0000h) 2h (R/W) = Enable event time-base counter equal to period (TBCNT = TBPRD) 3h (R/W) = Reserved 4h (R/W) = Enable event time-base counter equal to CMPA when the timer is incrementing. 5h (R/W) = Enable event time-base counter equal to CMPA when the timer is decrementing. 6h (R/W) = Enable event: time-base counter equal to CMPB when the timer is incrementing. 7h (R/W) = Enable event: time-base counter equal to CMPB when the timer is decrementing.

20.2.4.25 ETPS Register (Offset = 34h) [reset = 0h]

ETPS is shown in Figure 20-95 and described in Table 20-83.

Return to Summary Table.

Figure 20-95. ETPS Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				INTCNT		INTPRD	
R-0h				R-0h		R/W-0h	

Table 20-83. ETPS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	
3-2	INTCNT	R	0h	ePWM Interrupt Event (EPWMx_INT) Counter Register. These bits indicate how many selected ETSEL[INTSEL] events have occurred. These bits are automatically cleared when an interrupt pulse is generated. If interrupts are disabled, ETSEL[INT] = 0 or the interrupt flag is set, ETFLG[INT] = 1, the counter will stop counting events when it reaches the period value ETPS[INTCNT] = ETPS[INTPRD]. 0h (R/W) = No events have occurred. 1h (R/W) = 1 event has occurred. 2h (R/W) = 2 events have occurred. 3h (R/W) = 3 events have occurred.
1-0	INTPRD	R/W	0h	ePWM Interrupt (EPWMx_INT) Period Select. These bits determine how many selected ETSEL[INTSEL] events need to occur before an interrupt is generated. To be generated, the interrupt must be enabled (ETSEL[INT] = 1). If the interrupt status flag is set from a previous interrupt (ETFLG[INT] = 1) then no interrupt will be generated until the flag is cleared via the ETCLR[INT] bit. This allows for one interrupt to be pending while another is still being serviced. Once the interrupt is generated, the ETPS[INTCNT] bits will automatically be cleared. Writing a INTPRD value that is the same as the current counter value will trigger an interrupt if it is enabled and the status flag is clear. Writing a INTPRD value that is less than the current counter value will result in an undefined state. If a counter event occurs at the same instant as a new zero or non-zero INTPRD value is written, the counter is incremented. 0h (R/W) = Disable the interrupt event counter. No interrupt will be generated and ETFRC[INT] is ignored. 1h (R/W) = Generate an interrupt on the first event INTCNT = 01 (first event) 2h (R/W) = Generate interrupt on ETPS[INTCNT] = 1,0 (second event) 3h (R/W) = Generate interrupt on ETPS[INTCNT] = 1,1 (third event)

20.2.4.26 ETFLG Register (Offset = 36h) [reset = 0h]

ETFLG is shown in Figure 20-96 and described in Table 20-84.

Return to Summary Table.

Figure 20-96. ETFLG Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							INT
R-0h							R-0h

Table 20-84. ETFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R	0h	
0	INT	R	0h	Latched ePWM Interrupt (EPWMx_INT) Status Flag 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates that an ePWMx interrupt (EWPMx_INT) was generated. No further interrupts will be generated until the flag bit is cleared. Up to one interrupt can be pending while the ETFLG[INT] bit is still set. If an interrupt is pending, it will not be generated until after the ETFLG[INT] bit is cleared.

20.2.4.27 ETCLR Register (Offset = 38h) [reset = 0h]

ETCLR is shown in Figure 20-97 and described in Table 20-85.

Return to Summary Table.

Figure 20-97. ETCLR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							INT
R-0h							R-0h

Table 20-85. ETCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R	0h	
0	INT	R	0h	ePWM Interrupt (EPWMx_INT) Flag Clear Bit 0h (R/W) = Writing a 0 has no effect. Always reads back a 0. 1h (R/W) = Clears the ETFLG[INT] flag bit and enable further interrupts pulses to be generated. NOTE: Interrupts can also used as DMA events, and this will also enable further DMA events to be generated

20.2.4.28 ETFRC Register (Offset = 3Ah) [reset = 0h]

ETFRC is shown in Figure 20-98 and described in Table 20-86.

Return to Summary Table.

Figure 20-98. ETFRC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							INT
R-0h							R-0h

Table 20-86. ETFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R	0h	
0	INT	R	0h	INT Force Bit. The interrupt will only be generated if the event is enabled in the ETSEL register. The INT flag bit will be set regardless. 0h (R/W) = Writing 0 to this bit will be ignored. Always reads back a 0. 1h (R/W) = Generates an interrupt on EPWMxINT and set the INT flag bit. This bit is used for test purposes.

20.2.4.29 PCCTL Register (Offset = 3Ch) [reset = 0h]

PCCTL is shown in Figure 20-99 and described in Table 20-87.

Return to Summary Table.

Figure 20-99. PCCTL Register

15	14	13	12	11	10	9	8
RESERVED					CHPDUTY		
R-0h					R/W-0h		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CHPFREQ			OSHTWTH				CHPEN
R/W-0h			R/W-0h				R/W-0h

Table 20-87. PCCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	RESERVED	R	0h	
10-8	CHPDUTY	R/W	0h	Chopping Clock Duty Cycle 0h (R/W) = Duty = 1/8 (12.5%) 1h (R/W) = Duty = 2/8 (25.0%) 2h (R/W) = Duty = 3/8 (37.5%) 3h (R/W) = Duty = 4/8 (50.0%) 4h (R/W) = Duty = 5/8 (62.5%) 5h (R/W) = Duty = 6/8 (75.0%) 6h (R/W) = Duty = 7/8 (87.5%) 7h (R/W) = Reserved.
7-5	CHPFREQ	R/W	0h	Chopping Clock Frequency 0h (R/W) = Divide by 1 (no prescale). 1h (R/W) = Divide by 2. 2h (R/W) = Divide by 3. 3h (R/W) = Divide by 4. 4h (R/W) = Divide by 5. 5h (R/W) = Divide by 6. 6h (R/W) = Divide by 7. 7h (R/W) = Divide by 8.
4-1	OSHTWTH	R/W	0h	One-Shot Pulse Width 0h (R/W) = 1 - SYSCLKOUT/8 wide 1h (R/W) = 2 - SYSCLKOUT/8 wide 2h (R/W) = 3 - SYSCLKOUT/8 wide 3h (R/W) = 4 - SYSCLKOUT/8 wide Fh (R/W) = 16 - SYSCLKOUT/8 wide
0	CHPEN	R/W	0h	PWM-chopping Enable 0h (R/W) = Disable (bypass) PWM chopping function 1h (R/W) = Enable chopping function

20.2.4.30 HRCTL Register (Offset = C0h) [reset = 0h]

HRCTL is shown in Figure 20-100 and described in Table 20-88.

Return to Summary Table.

This register is only available on ePWM instances that include the high-resolution PWM (HRPWM) extension; otherwise, this location is reserved.

Figure 20-100. HRCTL Register

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td>PULSESEL</td><td>DELBUSSEL</td><td colspan="2">DELMODE</td></tr>
<tr><td colspan="4">R-0h</td><td>R/W-0h</td><td>R/W-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 20-88. HRCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	
3	PULSESEL	R/W	0h	Pulse select bits. Selects which pulse to use for timing events in the HRPWM module. Note: The user needs to select the pulse to match the selection in the EPWM module. If TBPHSHR bus is selected, then CNT_zero pulse should be used. If COMPAHR bus is selected, then it should match the bit setting of the CMPCTL[LOADMODE] bits in the EPWM module as follows. 0: CNT_zero pulse. 1h: PRD_eq pulse. 2h: CNT_zero or PRD_eq (should not use with HRPWM). 3h: No loads (should not use with HRPWM). 0h (R/W) = Select CNT_zero pulse 1h (R/W) = Select PRD_eq pulse
2	DELBUSSEL	R/W	0h	Delay Bus Select Bit: Selects which bus is used to select the delay for the PWM pulse. 0h (R/W) = Select CMPAHR(8) bus from compare module of EPWM (default on reset). 1h (R/W) = Select TBPHSHR(8) bus from time base module.
1-0	DELMODE	R/W	0h	Delay Mode Bits: Selects which edge of the PWM pulse the delay is inserted. Note: When DELMODE = 0,0, the HRCALM[CALMODE] bits are ignored and the delay line is in by-pass mode. Additionally, DLYIN is connected to CALIN and a continuous low value is fed to the delay line to minimize activity in the module. 0h (R/W) = No delay inserted (default on reset) 1h (R/W) = Delay inserted rising edge 2h (R/W) = Delay inserted falling edge 3h (R/W) = Delay inserted on both edges

20.3 Enhanced Capture (eCAP) Module

20.3.1 Introduction

20.3.1.1 Purpose of the Peripheral

Uses for eCAP include:

- Sample rate measurements of audio inputs
- Speed measurements of rotating machinery (for example, toothed sprockets sensed via Hall sensors)
- Elapsed time measurements between position sensor pulses
- Period and duty cycle measurements of pulse train signals
- Decoding current or voltage amplitude derived from duty cycle encoded current/voltage sensors

20.3.1.2 Features

The eCAP module includes the following features:

- 32-bit time base counter
- 4-event time-stamp registers (each 32 bits)
- Edge polarity selection for up to four sequenced time-stamp capture events
- Interrupt on either of the four events
- Single shot capture of up to four event time-stamps
- Continuous mode capture of time-stamps in a four-deep circular buffer
- Absolute time-stamp capture
- Difference (Delta) mode time-stamp capture
- All above resources dedicated to a single input pin
- When not used in capture mode, the ECAP module can be configured as a single channel PWM output

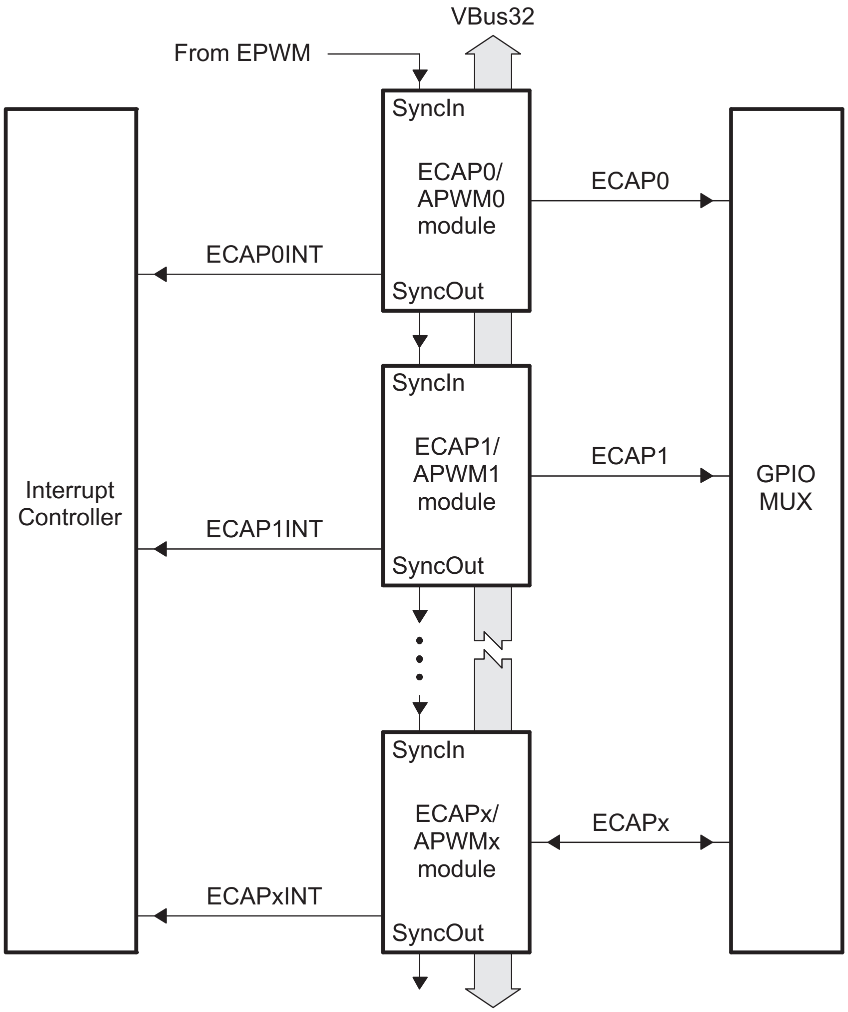
20.3.2 Functional Description

The eCAP module represents one complete capture channel that can be instantiated multiple times depending on the target device. In the context of this guide, one eCAP channel has the following independent key resources:

- Dedicated input capture pin
- 32-bit time base counter
- 4 × 32-bit time-stamp capture registers (CAP1-CAP4)
- 4-stage sequencer (Modulo4 counter) that is synchronized to external events, ECAP pin rising/falling edges.
- Independent edge polarity (rising/falling edge) selection for all 4 events
- Input capture signal prescaling (from 2-62)
- One-shot compare register (2 bits) to freeze captures after 1 to 4 time-stamp events
- Control for continuous time-stamp captures using a 4-deep circular buffer (CAP1-CAP4) scheme
- Interrupt capabilities on any of the 4 capture events

Multiple identical eCAP modules can be contained in a system as shown in Figure 20-101. The number of modules is device-dependent and is based on target application needs. In this chapter, the letter x within a signal or module name is used to indicate a generic eCAP instance on a device.

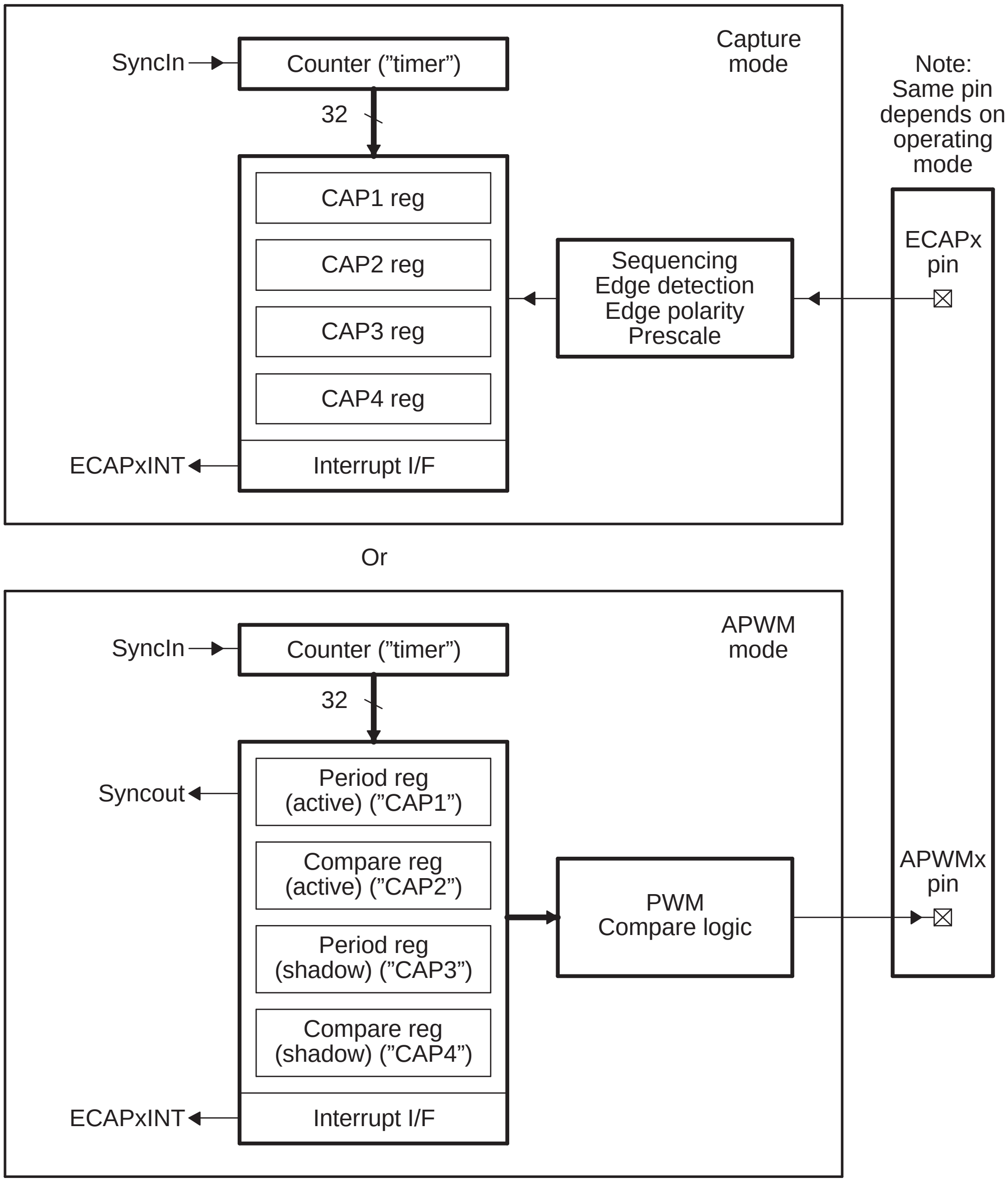
Figure 20-101. Multiple eCAP Modules



20.3.2.1 Capture and APWM Operating Mode

You can use the eCAP module resources to implement a single-channel PWM generator (with 32 bit capabilities) when it is not being used for input captures. The counter operates in count-up mode, providing a time-base for asymmetrical pulse width modulation (PWM) waveforms. The CAP1 and CAP2 registers become the active period and compare registers, respectively, while CAP3 and CAP4 registers become the period and capture shadow registers, respectively. Figure 20-102 is a high-level view of both the capture and auxiliary pulse-width modulator (APWM) modes of operation.

Figure 20-102. Capture and APWM Modes of Operation



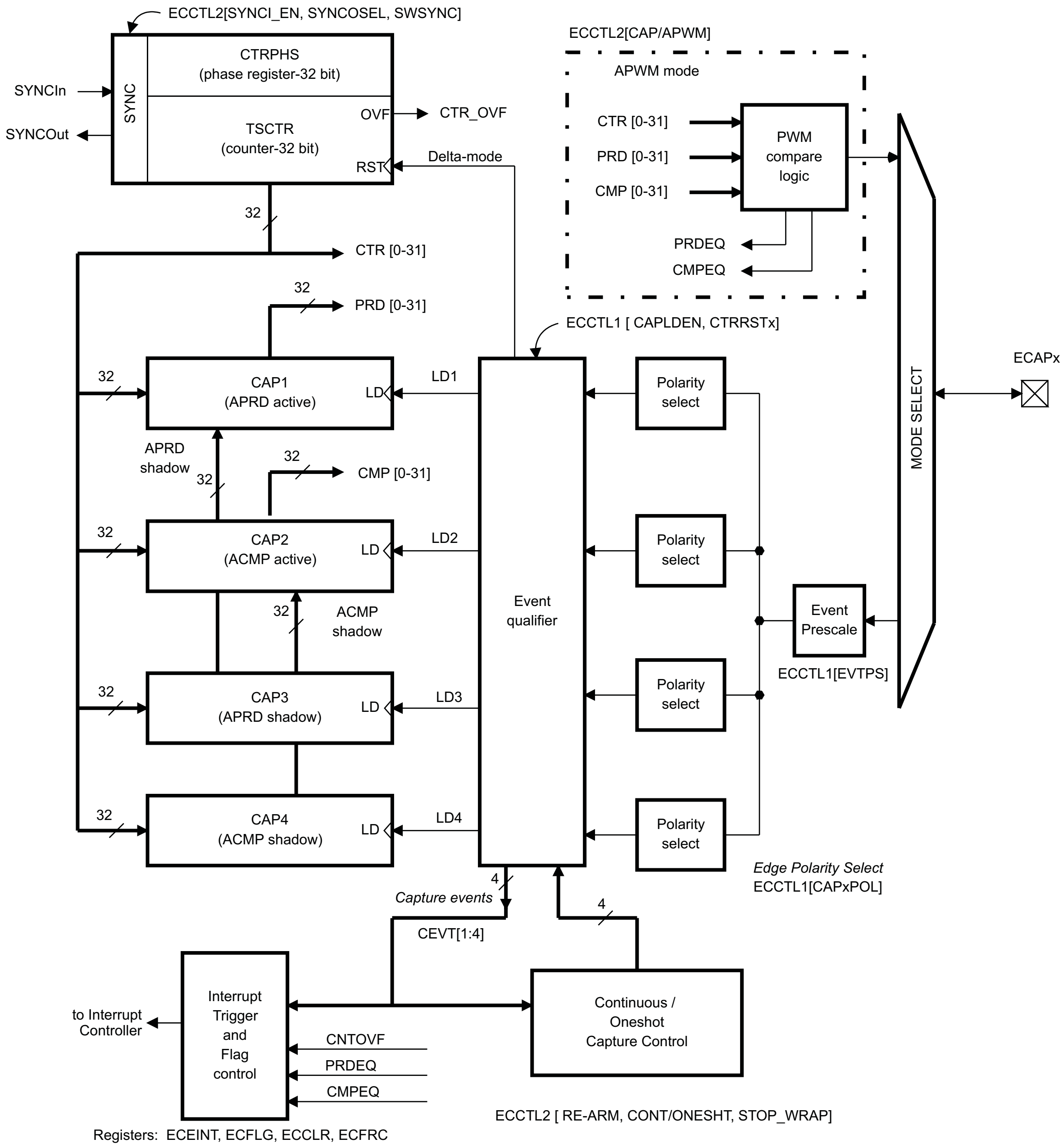
(1) A single pin is shared between CAP and APWM functions. In capture mode, it is an input; in APWM mode, it is an output.

(2) In APWM mode, writing any value to CAP1/CAP2 active registers also writes the same value to the corresponding shadow registers CAP3/CAP4. This emulates immediate mode. Writing to the shadow registers CAP3/CAP4 invokes the shadow mode.

20.3.2.2 Capture Mode Description

Figure 20-103 shows the various components that implement the capture function.

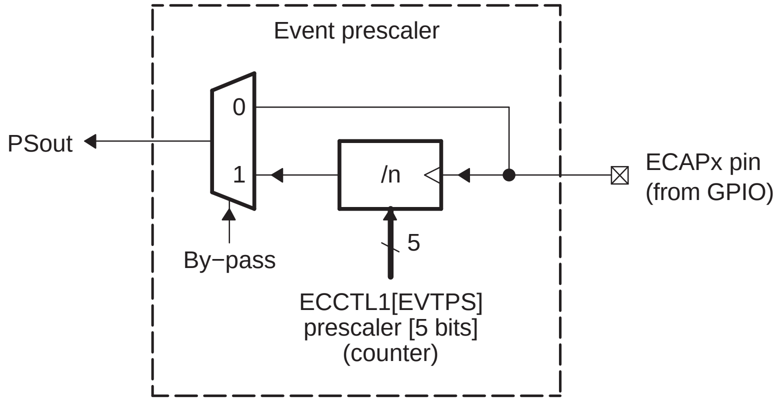
Figure 20-103. Capture Function Diagram



20.3.2.2.1 Event Prescaler

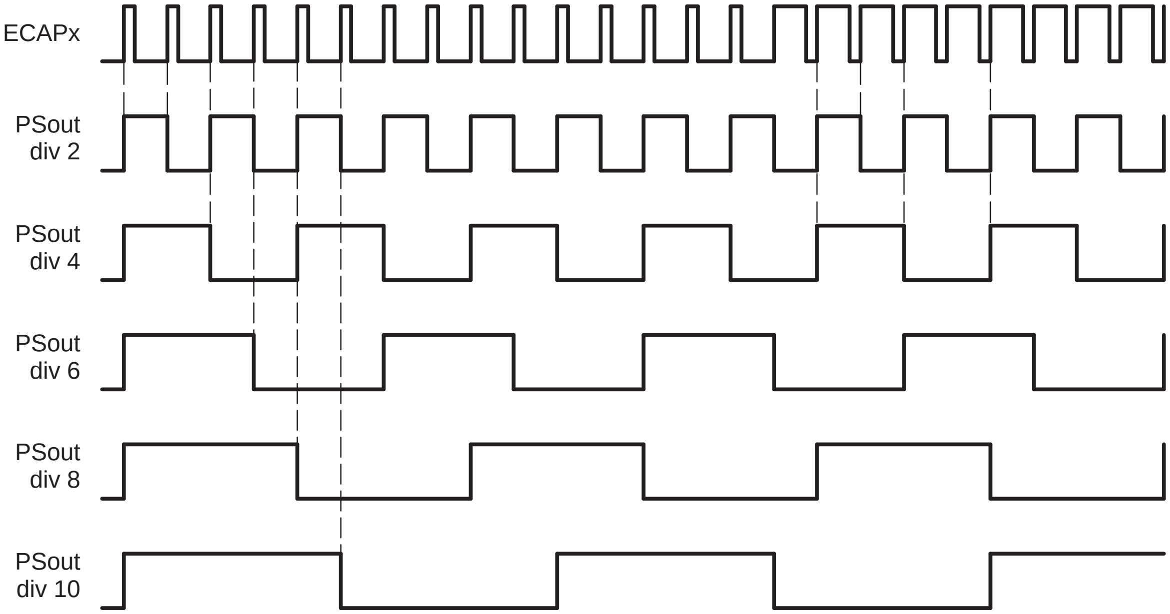
An input capture signal (pulse train) can be prescaled by N = 2-62 (in multiples of 2) or can bypass the prescaler. This is useful when very high frequency signals are used as inputs. Figure 20-104 shows a functional diagram and Figure 20-105 shows the operation of the prescale function.

Figure 20-104. Event Prescale Control



(1) When a prescale value of 1 is chosen (ECCTL1[13:9] = 0000) the input capture signal by-passes the prescale logic completely.

Figure 20-105. Prescale Function Waveforms



20.3.2.2.2 Edge Polarity Select and Qualifier

- Four independent edge polarity (rising edge/falling edge) selection multiplexers are used, one for each capture event.
- Each edge (up to 4) is event qualified by the Modulo4 sequencer.
- The edge event is gated to its respective CAP*n* register by the Mod4 counter. The CAP*n* register is loaded on the falling edge.

20.3.2.2.3 Continuous/One-Shot Control

- The Mod4 (2 bit) counter is incremented via edge qualified events (CEVT1-CEVT4).
- The Mod4 counter continues counting (0->1->2->3->0) and wraps around unless stopped.
- A 2-bit stop register is used to compare the Mod4 counter output, and when equal stops the Mod4 counter and inhibits further loads of the CAP1-CAP4 registers. This occurs during one-shot operation.

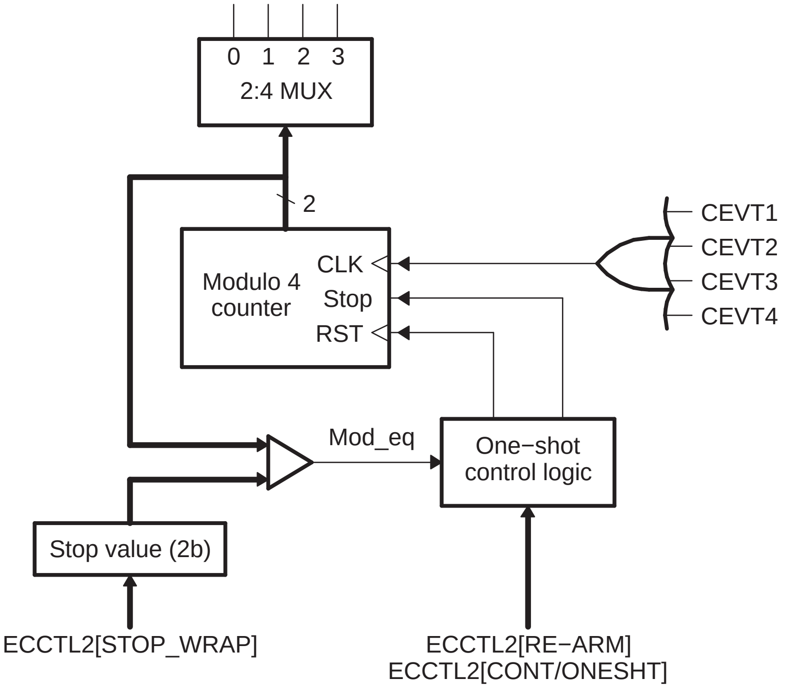
The continuous/one-shot block (Figure 20-106) controls the start/stop and reset (zero) functions of the Mod4 counter via a mono-shot type of action that can be triggered by the stop-value comparator and re-armed via software control.

Once armed, the eCAP module waits for 1-4 (defined by stop-value) capture events before freezing both the Mod4 counter and contents of CAP1-4 registers (time-stamps).

Re-arming prepares the eCAP module for another capture sequence. Also re-arming clears (to zero) the Mod4 counter and permits loading of CAP1-4 registers again, providing the CAPLDEN bit is set.

In continuous mode, the Mod4 counter continues to run (0->1->2->3->0, the one-shot action is ignored, and capture values continue to be written to CAP1-4 in a circular buffer sequence.

Figure 20-106. Continuous/One-shot Block Diagram



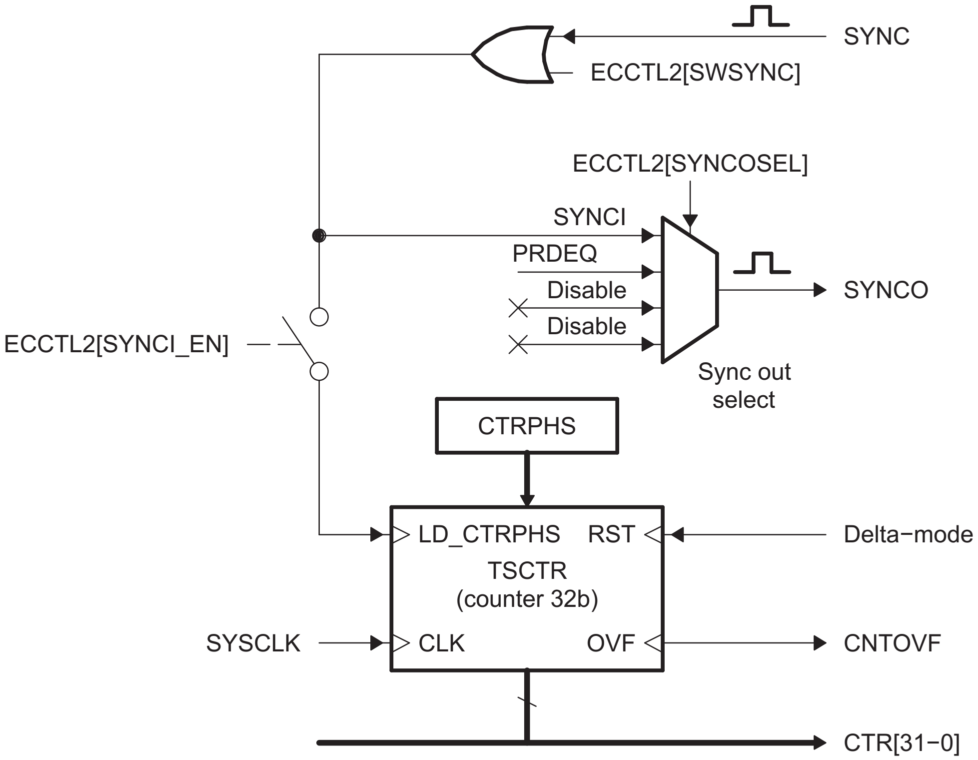
20.3.2.2.4 32-Bit Counter and Phase Control

This counter (Figure 20-107) provides the time-base for event captures, and is clocked via the system clock.

A phase register is provided to achieve synchronization with other counters, via a hardware and software forced sync. This is useful in APWM mode when a phase offset between modules is needed.

On any of the four event loads, an option to reset the 32-bit counter is given. This is useful for time difference capture. The 32-bit counter value is captured first, then it is reset to 0 by any of the LD1-LD4 signals.

Figure 20-107. Counter and Synchronization Block Diagram



20.3.2.2.5 CAP1-CAP4 Registers

These 32-bit registers are fed by the 32-bit counter timer bus, CTR[0-31] and are loaded (capture a time-stamp) when their respective LD inputs are strobed.

Loading of the capture registers can be inhibited via control bit CAPLDEN. During one-shot operation, this bit is cleared (loading is inhibited) automatically when a stop condition occurs, StopValue = Mod4.

CAP1 and CAP2 registers become the active period and compare registers, respectively, in APWM mode.

CAP3 and CAP4 registers become the respective shadow registers (APRD and ACMP) for CAP1 and CAP2 during APWM operation.

20.3.2.2.6 Interrupt Control

An Interrupt can be generated on capture events (CEVT1-CEVT4, CNTOVF) or APWM events (PRDEQ, CMPEQ). See Figure 20-108.

A counter overflow event (FFFF FFFFh->0000 0000h) is also provided as an interrupt source (CNTOVF).

The capture events are edge and sequencer qualified (that is, ordered in time) by the polarity select and Mod4 gating, respectively.

One of these events can be selected as the interrupt source (from the eCAP*n* module) going to the interrupt controller.

Seven interrupt events (CEVT1, CEVT2, CEVT3, CEVT4, CNTOVF, PRDEQ, CMPEQ) can be generated. The interrupt enable register (ECEINT) is used to enable/disable individual interrupt event sources. The interrupt flag register (ECFLG) indicates if any interrupt event has been latched and contains the global interrupt flag bit (INT). An interrupt pulse is generated to the interrupt controller only if any of the interrupt events are enabled, the flag bit is 1, and the INT flag bit is 0. The interrupt service routine must clear the global interrupt flag bit and the serviced event via the interrupt clear register (ECCLR) before any other interrupt pulses are generated. You can force an interrupt event via the interrupt force register (ECFRC). This is useful for test purposes.

Note that the interrupts coming from the eCAP module are also used as DMA events. The interrupt registers should be used to enable and clear the current DMA event in order for the eCAP module to generate subsequent DMA events.

20.3.2.2.7 Shadow Load and Lockout Control

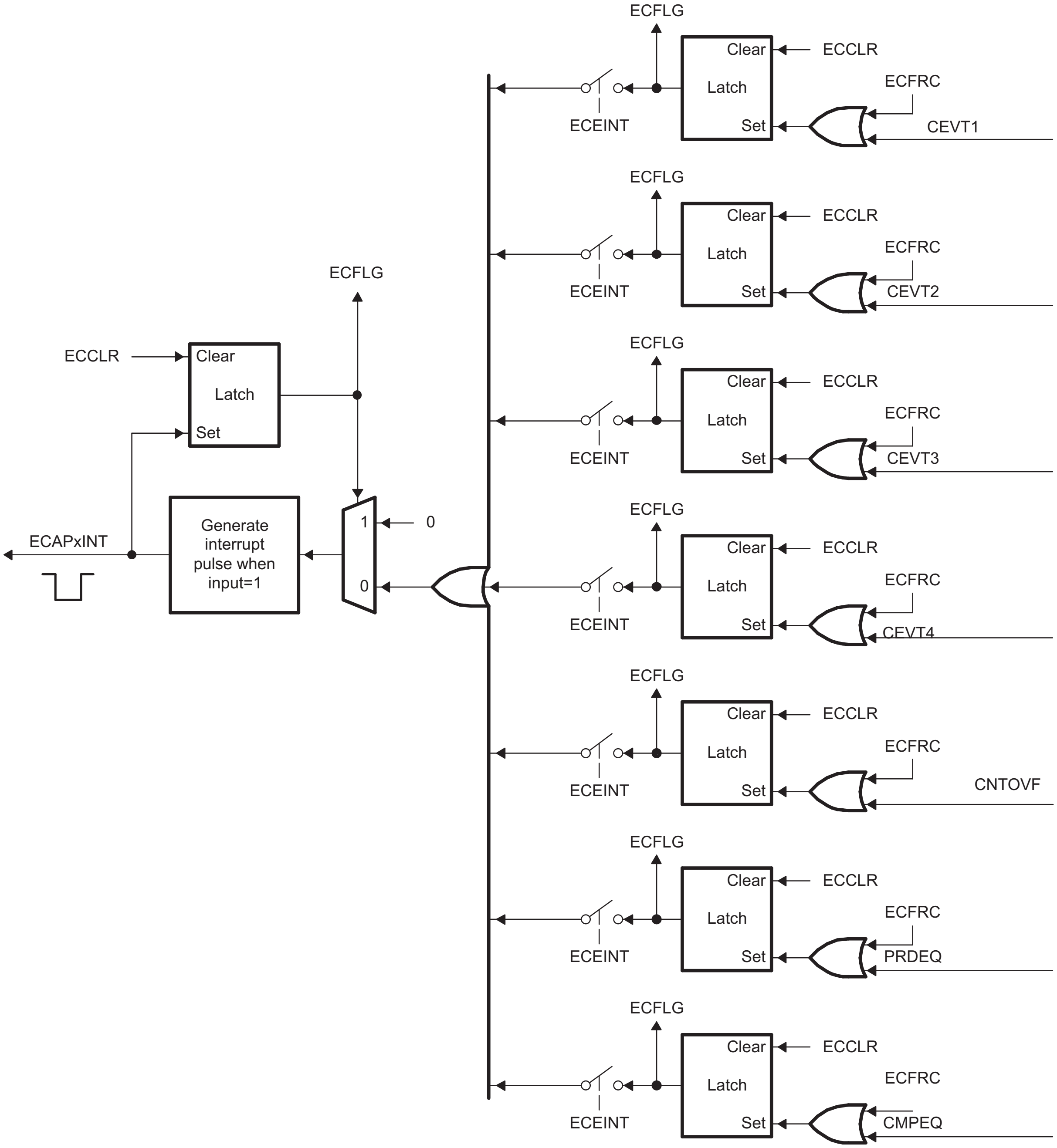
In capture mode, this logic inhibits (locks out) any shadow loading of CAP1 or CAP2 from APRD and ACMP registers, respectively.

In APWM mode, shadow loading is active and two choices are permitted:

- Immediate - APRD or ACMP are transferred to CAP1 or CAP2 immediately upon writing a new value.
- On period equal, CTR[31:0] = PRD[31:0]

NOTE: The CEVT1, CEVT2, CEVT3, CEVT4 flags are only active in capture mode (ECCTL2[CAP/APWM == 0]). The PRDEQ, CMPEQ flags are only valid in APWM mode (ECCTL2[CAP/APWM == 1]). CNTOVF flag is valid in both modes.

Figure 20-108. Interrupts in eCAP Module

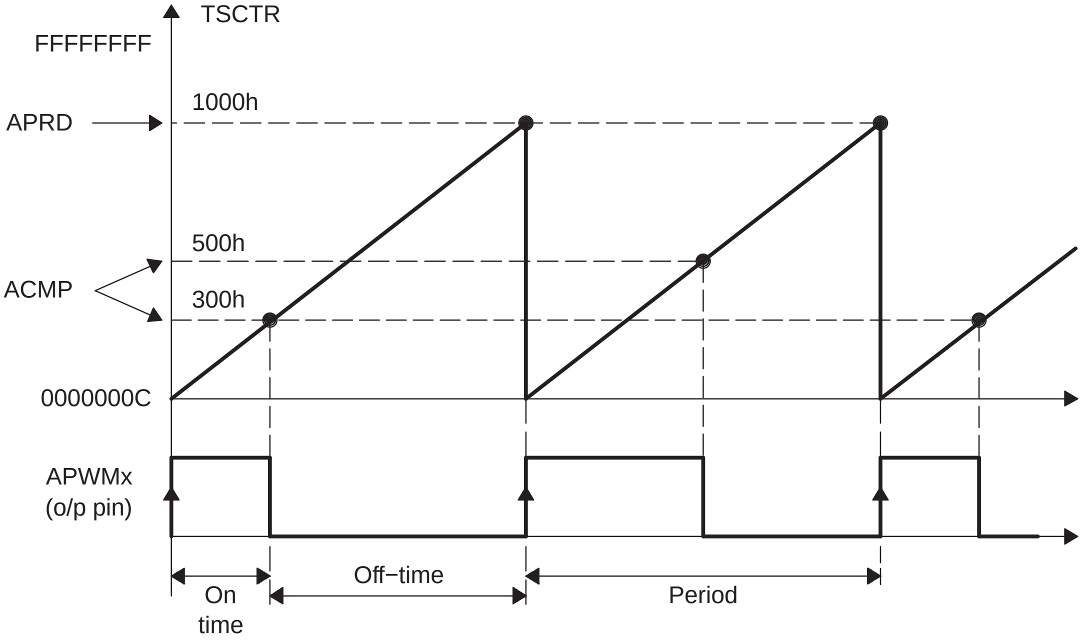


20.3.2.2.8 APWM Mode Operation

Main operating highlights of the APWM section:

- The time-stamp counter bus is made available for comparison via 2 digital (32-bit) comparators.
- When CAP1/2 registers are not used in capture mode, their contents can be used as Period and Compare values in APWM mode.
- Double buffering is achieved via shadow registers APRD and ACMP (CAP3/4). The shadow register contents are transferred over to CAP1/2 registers either immediately upon a write, or on a PRDEQ trigger.
- In APWM mode, writing to CAP1/CAP2 active registers will also write the same value to the corresponding shadow registers CAP3/CAP4. This emulates immediate mode. Writing to the shadow registers CAP3/CAP4 will invoke the shadow mode.
- During initialization, you must write to the active registers for both period and compare. This automatically copies the initial values into the shadow values. For subsequent compare updates, during run-time, you only need to use the shadow registers.

Figure 20-109. PWM Waveform Details Of APWM Mode Operation



The behavior of APWM active-high mode (APWMPOL == 0) is:

```
CMP = 0x00000000, output low for duration of period (0% duty)
CMP = 0x00000001, output high 1 cycle
CMP = 0x00000002, output high 2 cycles
CMP = PERIOD, output high except for 1 cycle (<100% duty)
CMP = PERIOD+1, output high for complete period (100% duty)
CMP > PERIOD+1, output high for complete period
```

The behavior of APWM active-low mode (APWMPOL == 1) is:

```
CMP = 0x00000000, output high for duration of period (0% duty)
CMP = 0x00000001, output low 1 cycle
CMP = 0x00000002, output low 2 cycles
CMP = PERIOD, output low except for 1 cycle (<100% duty)
CMP = PERIOD+1, output low for complete period (100% duty)
CMP > PERIOD+1, output low for complete period
```

20.3.3 Use Cases

The following sections will provide Applications examples and code snippets to show how to configure and operate the eCAP module. For clarity and ease of use, below are useful #defines which will help in the understanding of the examples.

```
// ECCTL1 ( ECAP Control Reg 1)
//==========================
// CAPxPOL bits
#define    EC_RISING              0x0
#define    EC_FALLING             0x1

// CTRRSTx bits
#define    EC_ABS_MODE            0x0
#define    EC_DELTA_MODE          0x1

// PRESCALE bits
#define    EC_BYPASS              0x0
#define    EC_DIV1                0x0
#define    EC_DIV2                0x1
#define    EC_DIV4                0x2
#define    EC_DIV6                0x3
#define    EC_DIV8                0x4
#define    EC_DIV10               0x5

// ECCTL2 ( ECAP Control Reg 2)
//==========================
// CONT/ONESHOT bit
#define    EC_CONTINUOUS          0x0
#define    EC_ONESHOT             0x1

// STOPVALUE bit
#define    EC_EVENT1              0x0
#define    EC_EVENT2              0x1
#define    EC_EVENT3              0x2
#define    EC_EVENT4              0x3

// RE-ARM bit
#define    EC_ARM                 0x1

// TSCTRSTOP bit
#define    EC_FREEZE              0x0
#define    EC_RUN                 0x1

// SYNCO_SEL bit
#define    EC_SYNCIN              0x0
#define    EC_CTR_PRD             0x1
#define    EC_SYNCO_DIS           0x2

// CAP/APWM mode bit
#define    EC_CAP_MODE            0x0
#define    EC_APWM_MODE           0x1

// APWMPOL bit
#define    EC_ACTV_HI             0x0
#define    EC_ACTV_LO             0x1

// Generic
#define    EC_DISABLE             0x0
#define    EC_ENABLE              0x1
#define    EC_FORCE               0x1
```

20.3.3.1 Absolute Time-Stamp Operation Rising Edge Trigger Example

Figure 20-110 shows an example of continuous capture operation (Mod4 counter wraps around). In this figure, TSCTR counts-up without resetting and capture events are qualified on the rising edge only, this gives period (and frequency) information.

On an event, the TSCTR contents (time-stamp) is first captured, then Mod4 counter is incremented to the next state. When the TSCTR reaches FFFF FFFFh (maximum value), it wraps around to 0000 0000h (not shown in Figure 20-110), if this occurs, the CNTOVF (counter overflow) flag is set, and an interrupt (if enabled) occurs, CNTOVF (counter overflow) Flag is set, and an Interrupt (if enabled) occurs. Captured time-stamps are valid at the point indicated by the diagram, after the 4th event, hence event CEVT4 can conveniently be used to trigger an interrupt and the CPU can read data from the CAP*n* registers.

Figure 20-110. Capture Sequence for Absolute Time-Stamp, Rising Edge Detect

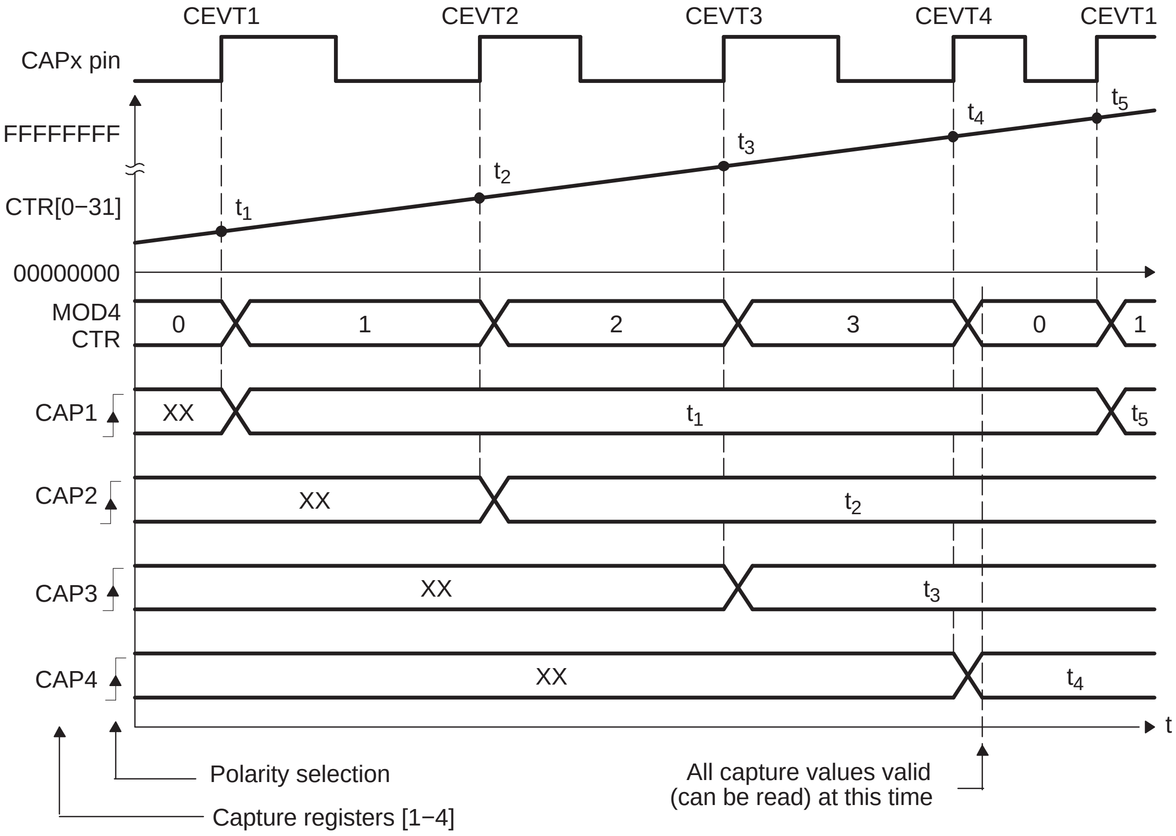


Table 20-89. ECAP Initialization for CAP Mode Absolute Time, Rising Edge Trigger

Register	Bit	Value
ECCTL1	CAP1POL	EC_RISING
ECCTL1	CAP2POL	EC_RISING
ECCTL1	CAP3POL	EC_RISING
ECCTL1	CAP4POL	EC_RISING
ECCTL1	CTRRST1	EC_ABS_MODE
ECCTL1	CTRRST2	EC_ABS_MODE
ECCTL1	CTRRST3	EC_ABS_MODE
ECCTL1	CTRRST4	EC_ABS_MODE
ECCTL1	CAPLDEN	EC_ENABLE
ECCTL1	PRESCALE	EC_DIV1
ECCTL2	CAP_APWM	EC_CAP_MODE
ECCTL2	CONT_ONESHT	EC_CONTINUOUS
ECCTL2	SYNCO_SEL	EC_SYNCO_DIS
ECCTL2	SYNCI_EN	EC_DISABLE
ECCTL2	TSCTRSTOP	EC_RUN

Example 20-9. Code Snippet for CAP Mode Absolute Time, Rising Edge Trigger

```
// Code snippet for CAP mode Absolute Time, Rising edge trigger

// Run Time ( e.g. CEVT4 triggered ISR call)
//==========================================
    TSt1 = ECAPxRegs.CAP1;        // Fetch Time-Stamp captured at t1
    TSt2 = ECAPxRegs.CAP2;        // Fetch Time-Stamp captured at t2
    TSt3 = ECAPxRegs.CAP3;        // Fetch Time-Stamp captured at t3
    TSt4 = ECAPxRegs.CAP4;        // Fetch Time-Stamp captured at t4

    Period1 = TSt2-TSt1;        // Calculate 1st period
    Period2 = TSt3-TSt2;        // Calculate 2nd period
    Period3 = TSt4-TSt3;        // Calculate 3rd period
```

20.3.3.2 Absolute Time-Stamp Operation Rising and Falling Edge Trigger Example

In Figure 20-111 the eCAP operating mode is almost the same as in the previous section except capture events are qualified as either rising or falling edge, this now gives both period and duty cycle information: Period1 = $t_3 - t_1$, Period2 = $t_5 - t_3$, …etc. Duty Cycle1 (on-time %) = ($t_2 - t_1$) / Period1 x 100%, etc. Duty Cycle1 (off-time %) = ($t_3 - t_2$) / Period1 x 100%, etc.

Figure 20-111. Capture Sequence for Absolute Time-Stamp, Rising and Falling Edge Detect

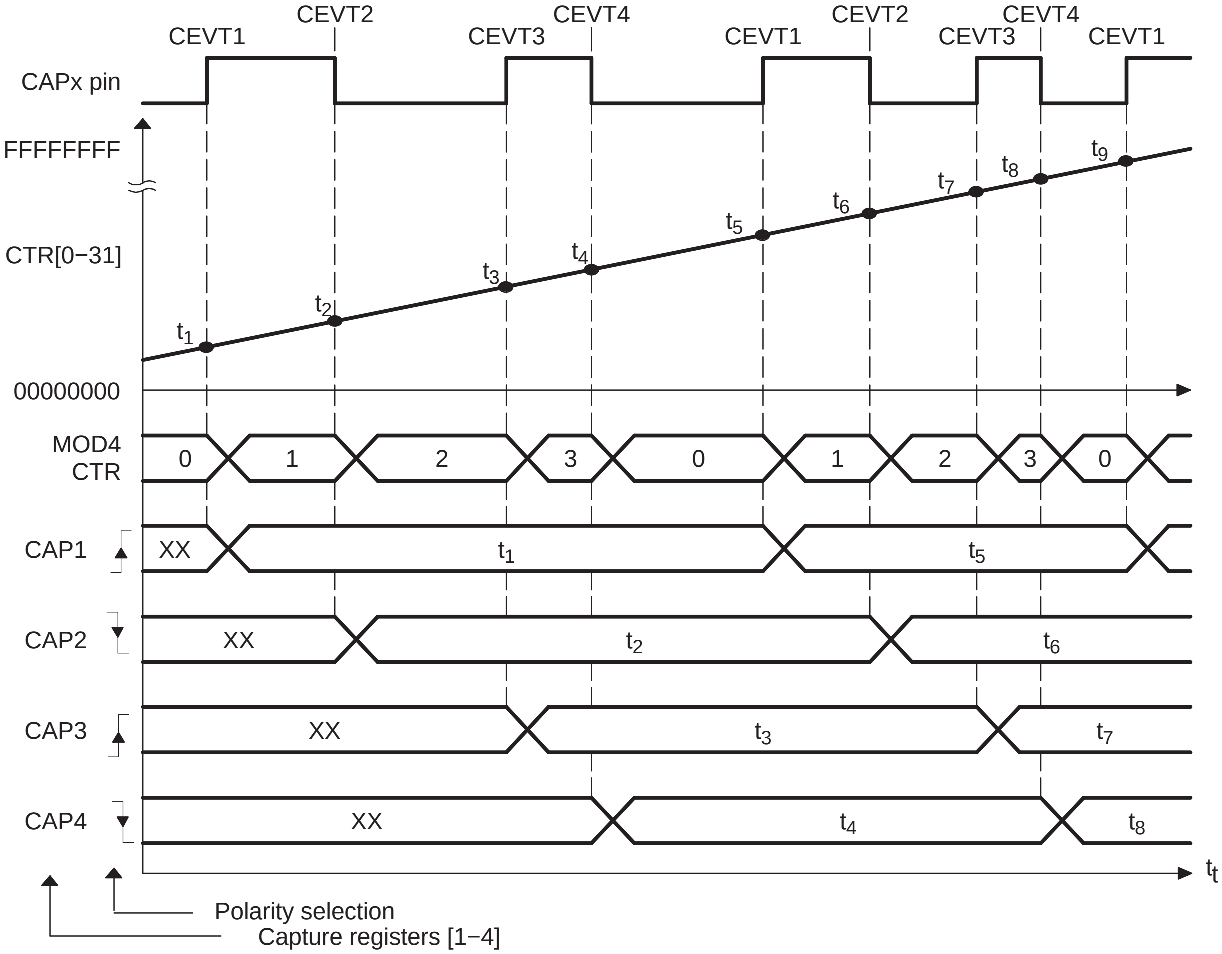


Table 20-90. ECAP Initialization for CAP Mode Absolute Time, Rising and Falling Edge Trigger

Register	Bit	Value
ECCTL1	CAP1POL	EC_RISING
ECCTL1	CAP2POL	EC_FALLING
ECCTL1	CAP3POL	EC_RISING
ECCTL1	CAP4POL	EC_FALLING
ECCTL1	CTRRST1	EC_ABS_MODE
ECCTL1	CTRRST2	EC_ABS_MODE
ECCTL1	CTRRST3	EC_ABS_MODE
ECCTL1	CTRRST4	EC_ABS_MODE
ECCTL1	CAPLDEN	EC_ENABLE
ECCTL1	PRESCALE	EC_DIV1
ECCTL2	CAP_APWM	EC_CAP_MODE
ECCTL2	CONT_ONESHT	EC_CONTINUOUS
ECCTL2	SYNCO_SEL	EC_SYNCO_DIS
ECCTL2	SYNCI_EN	EC_DISABLE
ECCTL2	TSCTRSTOP	EC_RUN

Example 20-10. Code Snippet for CAP Mode Absolute Time, Rising and Falling Edge Trigger

```
// Code snippet for CAP mode Absolute Time, Rising & Falling edge triggers

// Run Time ( e.g. CEVT4 triggered ISR call)
//==========================================
    TSt1 = ECAPxRegs.CAP1;        // Fetch Time-Stamp captured at t1
    TSt2 = ECAPxRegs.CAP2;        // Fetch Time-Stamp captured at t2
    TSt3 = ECAPxRegs.CAP3;        // Fetch Time-Stamp captured at t3
    TSt4 = ECAPxRegs.CAP4;        // Fetch Time-Stamp captured at t4

    Period1 = TSt3-TSt1;        // Calculate 1st period
    DutyOnTime1 = TSt2-TSt1;    // Calculate On time
    DutyOffTime1 = TSt3-TSt2;    // Calculate Off time
```

20.3.3.3 Time Difference (Delta) Operation Rising Edge Trigger Example

Figure 20-112 shows how the eCAP module can be used to collect Delta timing data from pulse train waveforms. Here Continuous Capture mode (TSCTR counts-up without resetting, and Mod4 counter wraps around) is used. In Delta-time mode, TSCTR is Reset back to Zero on every valid event. Here Capture events are qualified as Rising edge only. On an event, TSCTR contents (time-stamp) is captured first, and then TSCTR is reset to Zero. The Mod4 counter then increments to the next state. If TSCTR reaches FFFF FFFFh (maximum value), before the next event, it wraps around to 0000 0000h and continues, a CNTOVF (counter overflow) Flag is set, and an Interrupt (if enabled) occurs. The advantage of Delta-time Mode is that the CAP*n* contents directly give timing data without the need for CPU calculations: Period1 = T_1, Period2 = T_2,…etc. As shown in Figure 20-112, the CEVT1 event is a good trigger point to read the timing data, T_1, T_2, T_3, T_4 are all valid here.

Figure 20-112. Capture Sequence for Delta Mode Time-Stamp, Rising Edge Detect

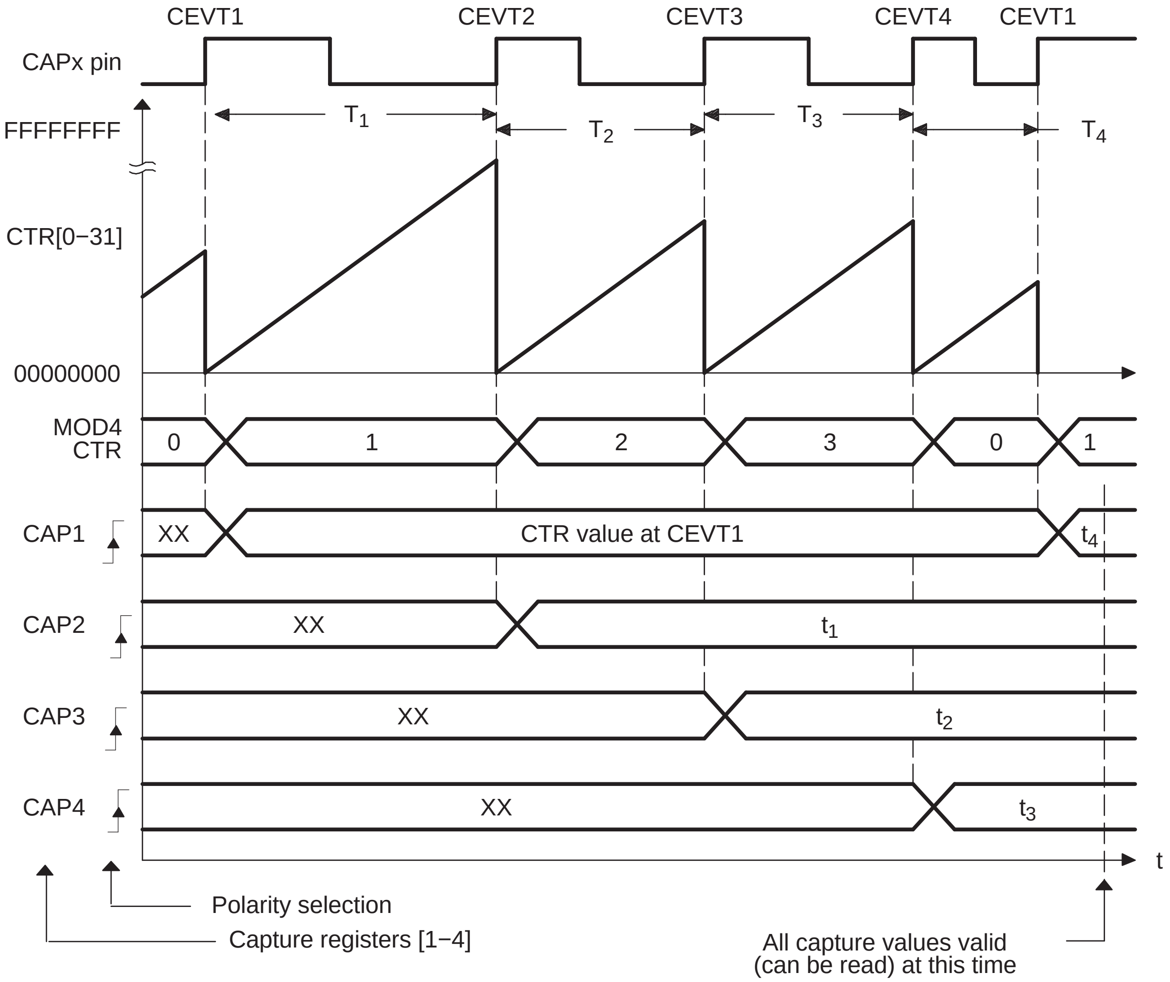


Table 20-91. ECAP Initialization for CAP Mode Delta Time, Rising Edge Trigger

Register	Bit	Value
ECCTL1	CAP1POL	EC_RISING
ECCTL1	CAP2POL	EC_RISING
ECCTL1	CAP3POL	EC_RISING
ECCTL1	CAP4POL	EC_RISING
ECCTL1	CTRRST1	EC_DELTA_MODE
ECCTL1	CTRRST2	EC_DELTA_MODE
ECCTL1	CTRRST3	EC_DELTA_MODE
ECCTL1	CTRRST4	EC_DELTA_MODE
ECCTL1	CAPLDEN	EC_ENABLE
ECCTL1	PRESCALE	EC_DIV1
ECCTL2	CAP_APWM	EC_CAP_MODE
ECCTL2	CONT_ONESHT	EC_CONTINUOUS
ECCTL2	SYNCO_SEL	EC_SYNCO_DIS
ECCTL2	SYNCI_EN	EC_DISABLE
ECCTL2	TSCTRSTOP	EC_RUN

Example 20-11. Code Snippet for CAP Mode Delta Time, Rising Edge Trigger

```
// Code snippet for CAP mode Delta Time, Rising edge trigger

// Run Time ( e.g. CEVT1 triggered ISR call)
//==========================================
// Note: here Time-stamp directly represents the Period value.
    Period4 = ECAPxRegs.CAP1;    // Fetch Time-Stamp captured at T1
    Period1 = ECAPxRegs.CAP2;    // Fetch Time-Stamp captured at T2
    Period2 = ECAPxRegs.CAP3;    // Fetch Time-Stamp captured at T3
    Period3 = ECAPxRegs.CAP4;    // Fetch Time-Stamp captured at T4
```

20.3.3.4 Time Difference (Delta) Operation Rising and Falling Edge Trigger Example

In Figure 20-113 the eCAP operating mode is almost the same as in previous section except Capture events are qualified as either Rising or Falling edge, this now gives both Period and Duty cycle information: Period1 = $T_1 + T_2$, Period2 = $T_3 + T_4$, …etc Duty Cycle1 (on-time %) = T_1 / Period1 × 100%, etc Duty Cycle1 (off-time %) = T_2 / Period1 × 100%, etc

During initialization, you must write to the active registers for both period and compare. This will then automatically copy the init values into the shadow values. For subsequent compare updates, that is, during run-time, only the shadow registers must be used.

Figure 20-113. Capture Sequence for Delta Mode Time-Stamp, Rising and Falling Edge Detect

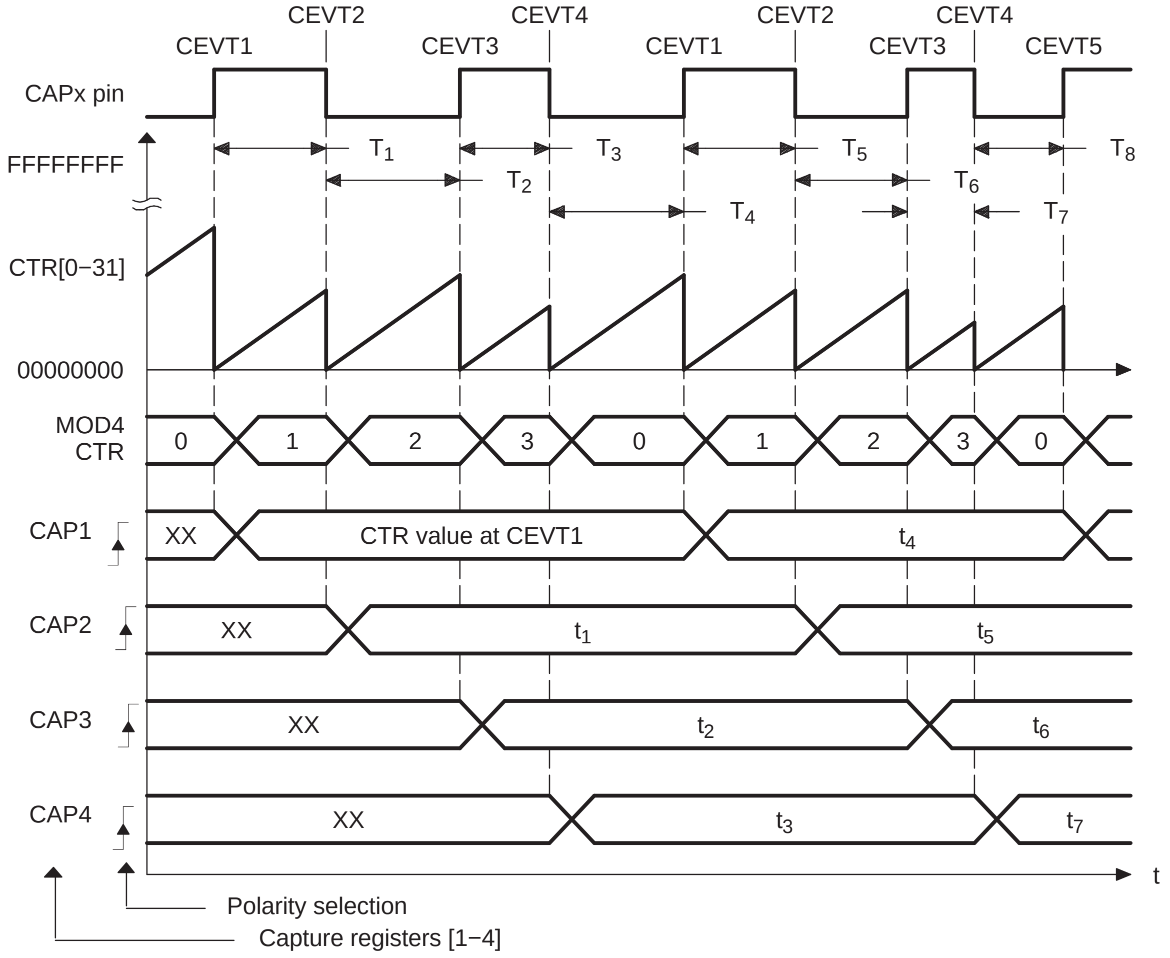


Table 20-92. ECAP Initialization for CAP Mode Delta Time, Rising and Falling Edge Triggers

Register	Bit	Value
ECCTL1	CAP1POL	EC_RISING
ECCTL1	CAP2POL	EC_FALLING
ECCTL1	CAP3POL	EC_RISING
ECCTL1	CAP4POL	EC_FALLING
ECCTL1	CTRRST1	EC_DELTA_MODE
ECCTL1	CTRRST2	EC_DELTA_MODE
ECCTL1	CTRRST3	EC_DELTA_MODE
ECCTL1	CTRRST4	EC_DELTA_MODE
ECCTL1	CAPLDEN	EC_ENABLE
ECCTL1	PRESCALE	EC_DIV1
ECCTL2	CAP_APWM	EC_CAP_MODE
ECCTL2	CONT_ONESHT	EC_CONTINUOUS
ECCTL2	SYNCO_SEL	EC_SYNCO_DIS
ECCTL2	SYNCI_EN	EC_DISABLE
ECCTL2	TSCTRSTOP	EC_RUN

Example 20-12. Code Snippet for CAP Mode Delta Time, Rising and Falling Edge Triggers

```
// Code snippet for CAP mode Delta Time, Rising and Falling edge triggers

// Run Time ( e.g. CEVT1 triggered ISR call)
//==========================================
// Note: here Time-stamp directly represents the Duty cycle values.
    DutyOnTime1 =  ECAPxRegs.CAP2;    // Fetch Time-Stamp captured at T2
    DutyOffTime1 = ECAPxRegs.CAP3;    // Fetch Time-Stamp captured at T3
    DutyOnTime2 =  ECAPxRegs.CAP4;    // Fetch Time-Stamp captured at T4
    DutyOffTime2 = ECAPxRegs.CAP1;    // Fetch Time-Stamp captured at T1

    Period1 = DutyOnTime1 + DutyOffTime1;
    Period2 = DutyOnTime2 + DutyOffTime2;
```

20.3.3.5 Application of the APWM Mode

20.3.3.5.1 Simple PWM Generation (Independent Channel/s) Example

In this example, the eCAP module is configured to operate as a PWM generator. Here a very simple single channel PWM waveform is generated from output pin APWM*n*. The PWM polarity is active high, which means that the compare value (CAP2 reg is now a compare register) represents the on-time (high level) of the period. Alternatively, if the APWMPOL bit is configured for active low, then the compare value represents the off-time.

Figure 20-114. PWM Waveform Details of APWM Mode Operation

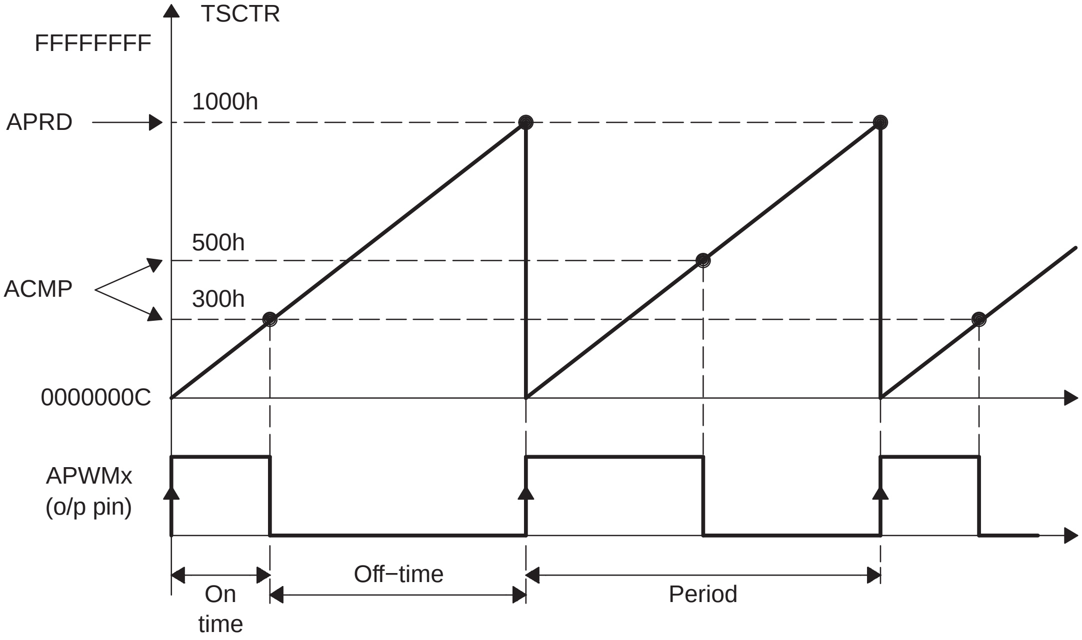


Table 20-93. ECAP Initialization for APWM Mode

Register	Bit	Value
CAP1	CAP1	0x1000
CTRPHS	CTRPHS	0x0
ECCTL2	CAP_APWM	EC_APWM_MODE
ECCTL2	APWMPOL	EC_ACTV_HI
ECCTL2	SYNCI_EN	EC_DISABLE
ECCTL2	SYNCO_SEL	EC_SYNCO_DIS
ECCTL2	TSCTRSTOP	EC_RUN

Example 20-13. Code Snippet for APWM Mode

```
// Code snippet for APWM mode Example 1

// Run Time (Instant 1, e.g. ISR call)
//======================
   ECAPxRegs.CAP2 = 0x300;      // Set Duty cycle i.e. compare value

// Run Time (Instant 2, e.g. another ISR call)
//======================
   ECAPxRegs.CAP2 = 0x500;      // Set Duty cycle i.e. compare value
```

20.3.3.5.2 Multichannel PWM Generation with Synchronization Example

Figure 20-115 takes advantage of the synchronization feature between eCAP modules. Here 4 independent PWM channels are required with different frequencies, but at integer multiples of each other to avoid "beat" frequencies. Hence one eCAP module is configured as the Master and the remaining 3 are Slaves all receiving their synch pulse (CTR = PRD) from the master. Note the Master is chosen to have the lower frequency (F1 = 1/20,000) requirement. Here Slave2 Freq = 2 × F1, Slave3 Freq = 4 × F1 and Slave4 Freq = 5 × F1. Note here values are in decimal notation. Also, only the APWM1 output waveform is shown.

Figure 20-115. Multichannel PWM Example Using 4 eCAP Modules

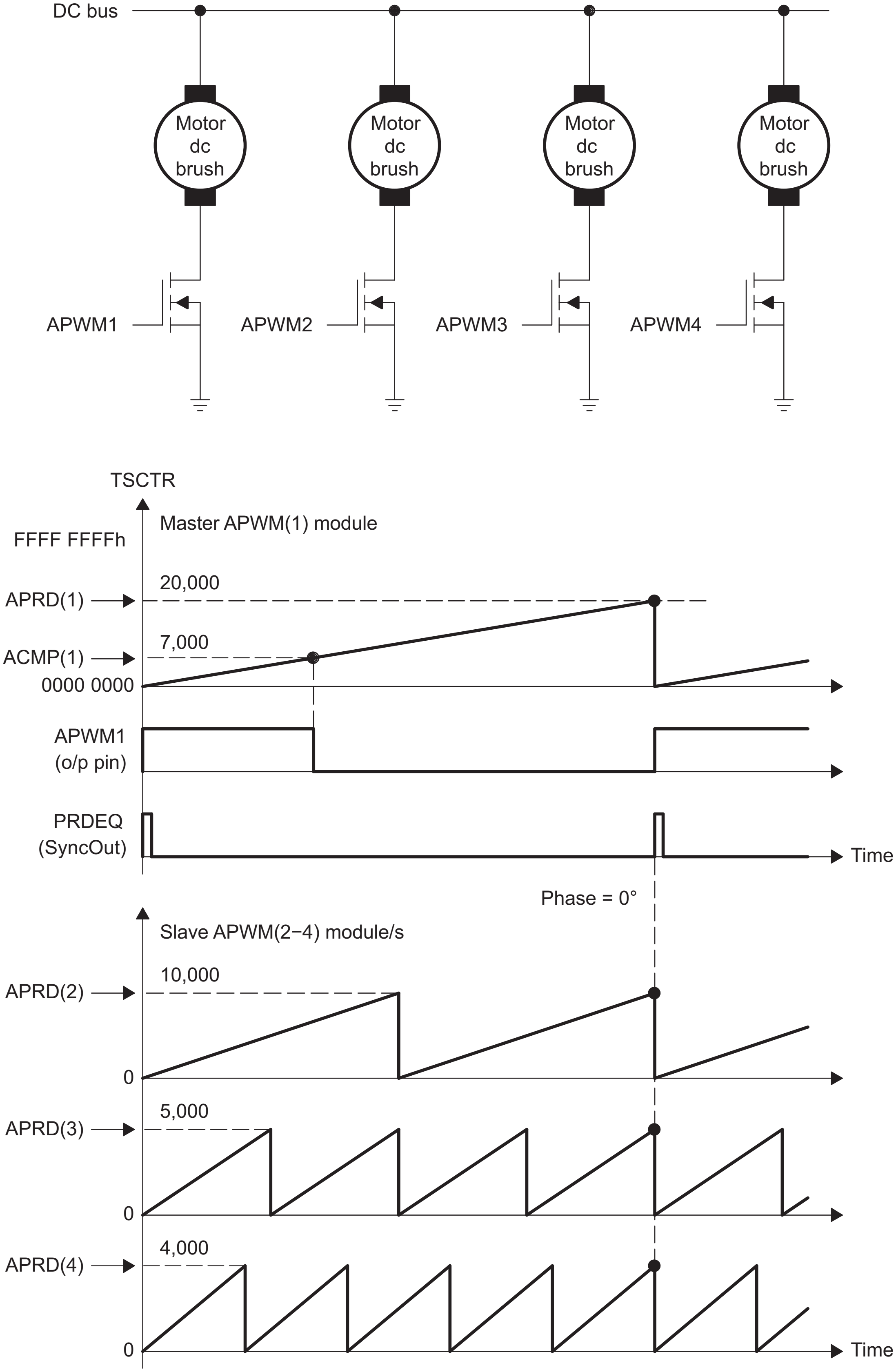


Table 20-94. ECAP1 Initialization for Multichannel PWM Generation with Synchronization

Register	Bit	Value
CAP1	CAP1	20000
CTRPHS	CTRPHS	0
ECCTL2	CAP_APWM	EC_APWM_MODE
ECCTL2	APWMPOL	EC_ACTV_HI
ECCTL2	SYNCI_EN	EC_DISABLE
ECCTL2	SYNCO_SEL	EC_CTR_PRD
ECCTL2	TSCTRSTOP	EC_RUN

Table 20-95. ECAP2 Initialization for Multichannel PWM Generation with Synchronization

Register	Bit	Value
CAP1	CAP1	10000
CTRPHS	CTRPHS	0
ECCTL2	CAP_APWM	EC_APWM_MODE
ECCTL2	APWMPOL	EC_ACTV_HI
ECCTL2	SYNCI_EN	EC_ENABLE
ECCTL2	SYNCO_SEL	EC_SYNCI
ECCTL2	TSCTRSTOP	EC_RUN

Table 20-96. ECAP3 Initialization for Multichannel PWM Generation with Synchronization

Register	Bit	Value
CAP1	CAP1	5000
CTRPHS	CTRPHS	0
ECCTL2	CAP_APWM	EC_APWM_MODE
ECCTL2	APWMPOL	EC_ACTV_HI
ECCTL2	SYNCI_EN	EC_ENABLE
ECCTL2	SYNCO_SEL	EC_SYNCI
ECCTL2	TSCTRSTOP	EC_RUN

Table 20-97. ECAP4 Initialization for Multichannel PWM Generation with Synchronization

Register	Bit	Value
CAP1	CAP1	4000
CTRPHS	CTRPHS	0
ECCTL2	CAP_APWM	EC_APWM_MODE
ECCTL2	APWMPOL	EC_ACTV_HI
ECCTL2	SYNCI_EN	EC_ENABLE
ECCTL2	SYNCO_SEL	EC_SYNCO_DIS
ECCTL2	TSCTRSTOP	EC_RUN

Example 20-14. Code Snippet for Multichannel PWM Generation with Synchronization

```
// Code snippet for APWM mode Example 2

// Run Time (Note: Example execution of one run-time instant)
//=========================================================
   ECAP1Regs.CAP2 = 7000;    // Set Duty cycle i.e., compare value = 7000
   ECAP2Regs.CAP2 = 2000;    // Set Duty cycle i.e., compare value = 2000
   ECAP3Regs.CAP2 = 550;     // Set Duty cycle i.e., compare value = 550
   ECAP4Regs.CAP2 = 6500;    // Set Duty cycle i.e., compare value = 6500
```

20.3.3.5.3 Multichannel PWM Generation with Phase Control Example

In Figure 20-116, the Phase control feature of the APWM mode is used to control a 3 phase Interleaved DC/DC converter topology. This topology requires each phase to be off-set by 120° from each other. Hence if “Leg” 1 (controlled by APWM1) is the reference Leg (or phase), that is, 0°, then Leg 2 need 120° off-set and Leg 3 needs 240° off-set. The waveforms in Figure 20-116 show the timing relationship between each of the phases (Legs). Note eCAP1 module is the Master and issues a sync out pulse to the slaves (modules 2, 3) whenever TSCTR = Period value.

Figure 20-116. Multiphase (channel) Interleaved PWM Example Using 3 eCAP Modules

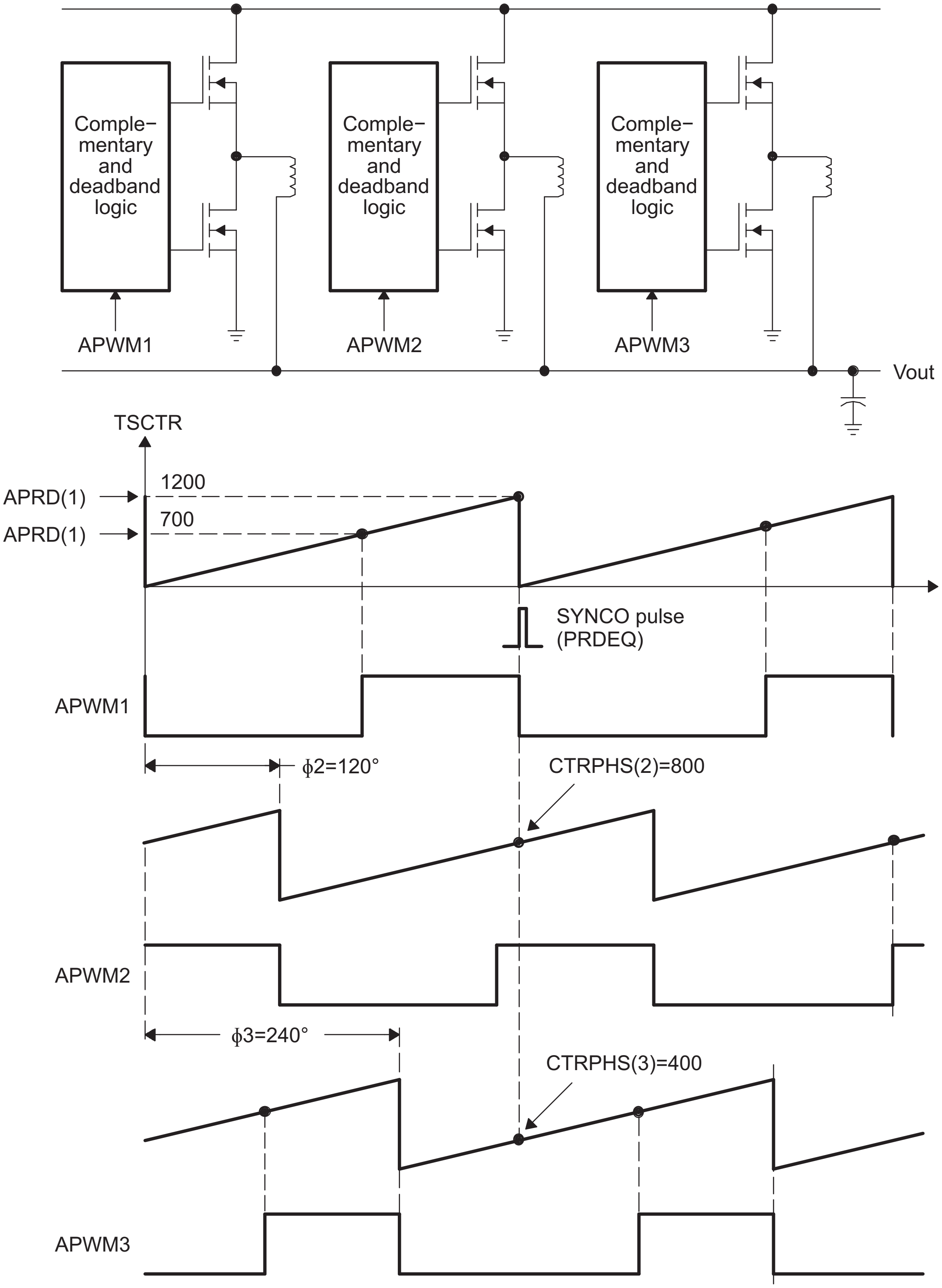


Table 20-98. ECAP1 Initialization for Multichannel PWM Generation with Phase Control

Register	Bit	Value
CAP1	CAP1	1200
CTRPHS	CTRPHS	0
ECCTL2	CAP_APWM	EC_APWM_MODE
ECCTL2	APWMPOL	EC_ACTV_HI
ECCTL2	SYNCI_EN	EC_DISABLE
ECCTL2	SYNCO_SEL	EC_CTR_PRD
ECCTL2	TSCTRSTOP	EC_RUN

Table 20-99. ECAP2 Initialization for Multichannel PWM Generation with Phase Control

Register	Bit	Value
CAP1	CAP1	1200
CTRPHS	CTRPHS	800
ECCTL2	CAP_APWM	EC_APWM_MODE
ECCTL2	APWMPOL	EC_ACTV_HI
ECCTL2	SYNCI_EN	EC_ENABLE
ECCTL2	SYNCO_SEL	EC_SYNCI
ECCTL2	TSCTRSTOP	EC_RUN

Table 20-100. ECAP3 Initialization for Multichannel PWM Generation with Phase Control

Register	Bit	Value
CAP1	CAP1	1200
CTRPHS	CTRPHS	400
ECCTL2	CAP_APWM	EC_APWM_MODE
ECCTL2	APWMPOL	EC_ACTV_HI
ECCTL2	SYNCI_EN	EC_ENABLE
ECCTL2	SYNCO_SEL	EC_SYNCO_DIS
ECCTL2	TSCTRSTOP	EC_RUN

Example 20-15. Code Snippet for Multichannel PWM Generation with Phase Control

```
// Code snippet for APWM mode Example 3


// Run Time (Note: Example execution of one run-time instant)
//=========================================================
// All phases are set to the same duty cycle
  ECAP1Regs.CAP2 = 700;     // Set Duty cycle i.e. compare value = 700
  ECAP2Regs.CAP2 = 700;     // Set Duty cycle i.e. compare value = 700
  ECAP3Regs.CAP2 = 700;     // Set Duty cycle i.e. compare value = 700
```

20.3.4 Registers

All 32-bit registers are aligned on even address boundaries and are organized in little-endian mode. The 16 least-significant bits of a 32-bit register are located on lowest address (even address).

NOTE: In APWM mode, writing to CAP1/CAP2 active registers also writes the same value to the corresponding shadow registers CAP3/CAP4. This emulates immediate mode. Writing to the shadow registers CAP3/CAP4 invokes the shadow mode.

20.3.5 PWMSS_ECAP Registers

Table 20-101 lists the memory-mapped registers for the PWMSS_ECAP. All register offset addresses not listed in Table 20-101 should be considered as reserved locations and the register contents should not be modified.

Table 20-101. PWMSS_ECAP Registers

Offset	Acronym	Register Name	Section
0h	TSCTR	Time-Stamp Counter Register	Section 20.3.5.1
4h	CTRPHS	Counter Phase Offset Value Register	Section 20.3.5.2
8h	CAP1	Capture 1 Register	Section 20.3.5.3
Ch	CAP2	Capture 2 Register	Section 20.3.5.4
10h	CAP3	Capture 3 Register	Section 20.3.5.5
14h	CAP4	Capture 4 Register	Section 20.3.5.6
28h	ECCTL1	Capture Control Register 1	Section 20.3.5.7
2Ah	ECCTL2	Capture Control Register 2	Section 20.3.5.8
2Ch	ECEINT	Capture Interrupt Enable Register	Section 20.3.5.9
2Eh	ECFLG	Capture Interrupt Flag Register	Section 20.3.5.10
30h	ECCLR	Capture Interrupt Clear Register	Section 20.3.5.11
32h	ECFRC	Capture Interrupt Force Register	Section 20.3.5.12
5Ch	REVID	Revision ID Register	Section 20.4.3.25

20.3.5.1 TSCTR Register (Offset = 0h) [reset = 0h]

TSCTR is shown in Figure 20-117 and described in Table 20-102.

Return to Summary Table.

Figure 20-117. TSCTR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">TSCTR</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 20-102. TSCTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TSCTR	R/W	0h	Active 32 bit counter register that is used as the capture time-base

20.3.5.2 CTRPHS Register (Offset = 4h) [reset = 0h]

CTRPHS is shown in Figure 20-118 and described in Table 20-103.

Return to Summary Table.

Figure 20-118. CTRPHS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTRPHS																															
R/W-0h																															

Table 20-103. CTRPHS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CTRPHS	R/W	0h	Counter phase value register that can be programmed for phase lag/lead. This register shadows TSCTR and is loaded into TSCTR upon either a SYNCI event or S/W force via a control bit. Used to achieve phase control synchronization with respect to other eCAP and EPWM time-bases.

20.3.5.3 CAP1 Register (Offset = 8h) [reset = 0h]

CAP1 is shown in Figure 20-119 and described in Table 20-104.

Return to Summary Table.

Figure 20-119. CAP1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP1																															
R/W-0h																															

Table 20-104. CAP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP1	R/W	0h	This register can be loaded (written) by the following. (a) Time-Stamp (that is, counter value) during a capture event. (b) Software may be useful for test purposes. (c) APRD active register when used in APWM mode.

20.3.5.4 CAP2 Register (Offset = Ch) [reset = 0h]

CAP2 is shown in Figure 20-120 and described in Table 20-105.

Return to Summary Table.

Figure 20-120. CAP2 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">CAP2</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 20-105. CAP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP2	R/W	0h	This register can be loaded (written) by the following. (a) Time-Stamp (that is, counter value) during a capture event. (b) Software may be useful for test purposes. (c) ACMP active register when used in APWM mode.

20.3.5.5 CAP3 Register (Offset = 10h) [reset = 0h]

CAP3 is shown in Figure 20-121 and described in Table 20-106.

Return to Summary Table.

Figure 20-121. CAP3 Register

31-0
CAP3
R/W-0h

Table 20-106. CAP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP3	R/W	0h	In CMP mode, this is a time-stamp capture register. In APWM mode, this is the period shadow (APRD) register. You update the PWM period value through this register. In this mode, CAP3 shadows CAP1.

20.3.5.6 CAP4 Register (Offset = 14h) [reset = 0h]

CAP4 is shown in Figure 20-122 and described in Table 20-107.

Return to Summary Table.

Figure 20-122. CAP4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP4																															
R/W-0h																															

Table 20-107. CAP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP4	R/W	0h	In CMP mode, this is a time-stamp capture register. In APWM mode, this is the compare shadow (ACMP) register. You update the PWM compare value through this register. In this mode, CAP4 shadows CAP2.

20.3.5.7 ECCTL1 Register (Offset = 28h) [reset = 0h]

ECCTL1 is shown in Figure 20-123 and described in Table 20-108.

Return to Summary Table.

Figure 20-123. ECCTL1 Register

15	14	13	12	11	10	9	8
FREE_SOFT		PRESCALE					CAPLDEN
R/W-0h		R/W-0h					R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CTRRST4	CAP4POL	CTRRST3	CAP3POL	CTRRST2	CAP2POL	CTRRST1	CAP1POL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-108. ECCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	FREE_SOFT	R/W	0h	Emulation Control 0h (R/W) = TSCTR counter stops immediately on emulation suspend. 1h (R/W) = TSCTR counter runs until = 0. 2h (R/W) = TSCTR counter is unaffected by emulation suspend (Run Free). 3h (R/W) = TSCTR counter is unaffected by emulation suspend (Run Free).
13-9	PRESCALE	R/W	0h	Event Filter prescale select ... 0h (R/W) = Divide by 1 (i.e,. no prescale, by-pass the prescaler) 1h (R/W) = Divide by 2 2h (R/W) = Divide by 4 3h (R/W) = Divide by 6 4h (R/W) = Divide by 8 5h (R/W) = Divide by 10 1Eh (R/W) = Divide by 60 1Fh (R/W) = Divide by 62
8	CAPLDEN	R/W	0h	Enable Loading of CAP1 to CAP4 registers on a capture event 0h (R/W) = Disable CAP1 through 4 register loads at capture event time. 1h (R/W) = Enable CAP1-4 register loads at capture event time.
7	CTRRST4	R/W	0h	Counter Reset on Capture Event 4 0h (R/W) = Do not reset counter on Capture Event 4 (absolute time stamp operation) 1h (R/W) = Reset counter after Capture Event 4 time-stamp has been captured (used in difference mode operation)
6	CAP4POL	R/W	0h	Capture Event 4 Polarity select 0h (R/W) = Capture Event 4 triggered on a rising edge (RE) 1h (R/W) = Capture Event 4 triggered on a falling edge (FE)
5	CTRRST3	R/W	0h	Counter Reset on Capture Event 3 0h (R/W) = Do not reset counter on Capture Event 3 (absolute time stamp) 1h (R/W) = Reset counter after Event 3 time-stamp has been captured (used in difference mode operation)
4	CAP3POL	R/W	0h	Capture Event 3 Polarity select 0h (R/W) = Capture Event 3 triggered on a rising edge (RE) 1h (R/W) = Capture Event 3 triggered on a falling edge (FE)

Table 20-108. ECCTL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	CTRRST2	R/W	0h	Counter Reset on Capture Event 2 0h (R/W) = Do not reset counter on Capture Event 2 (absolute time stamp) 1h (R/W) = Reset counter after Event 2 time-stamp has been captured (used in difference mode operation)
2	CAP2POL	R/W	0h	Capture Event 2 Polarity select 0h (R/W) = Capture Event 2 triggered on a rising edge (RE) 1h (R/W) = Capture Event 2 triggered on a falling edge (FE)
1	CTRRST1	R/W	0h	Counter Reset on Capture Event 1 0h (R/W) = Do not reset counter on Capture Event 1 (absolute time stamp) 1h (R/W) = Reset counter after Event 1 time-stamp has been captured (used in difference mode operation)
0	CAP1POL	R/W	0h	Capture Event 1 Polarity select 0h (R/W) = Capture Event 1 triggered on a rising edge (RE) 1h (R/W) = Capture Event 1 triggered on a falling edge (FE)

20.3.5.8 ECCTL2 Register (Offset = 2Ah) [reset = 6h]

ECCTL2 is shown in Figure 20-124 and described in Table 20-109.

Return to Summary Table.

Figure 20-124. ECCTL2 Register

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="5">RESERVED</td><td>APWMPOL</td><td>CAP_APWM</td><td>SWSYNC</td></tr>
<tr><td colspan="5">R-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">SYNCO_SEL</td><td>SYNCI_EN</td><td>TSCTRSTOP</td><td>REARM</td><td colspan="2">STOP_WRAP</td><td>CONT_ONESHT</td></tr>
<tr><td colspan="2">R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td colspan="2">R/W-3h</td><td>R/W-0h</td></tr>
</table>

Table 20-109. ECCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	RESERVED	R	0h	
10	APWMPOL	R/W	0h	APWM output polarity select. This is applicable only in APWM operating mode 0h (R/W) = Output is active high (Compare value defines high time) 1h (R/W) = Output is active low (Compare value defines low time)
9	CAP_APWM	R/W	0h	CAP/APWM operating mode select 0h (R/W) = ECAP module operates in capture mode. This mode forces the following configuration. (a) Inhibits TSCTR resets via PRDEQ event. (b) Inhibits shadow loads on CAP1 and 2 registers. (c) Permits user to enable CAP1-4 register load. (d) ECAPn/APWMn pin operates as a capture input. 1h (R/W) = ECAP module operates in APWM mode. This mode forces the following configuration. (a) Resets TSCTR on PRDEQ event (period boundary). (b) Permits shadow loading on CAP1 and 2 registers. (c) Disables loading of time-stamps into CAP1-4 registers. (d) ECAPn/APWMn pin operates as a APWM output.
8	SWSYNC	R/W	0h	Software-forced Counter (TSCTR) Synchronizing. This provides a convenient software method to synchronize some or all ECAP time bases. In APWM mode, the synchronizing can also be done via the PRDEQ event. Note: Selecting PRDEQ is meaningful only in APWM mode. However, you can choose it in CAP mode if you find doing so useful. 0h (R/W) = Writing a zero has no effect. Reading always returns a zero 1h (R/W) = Writing a one forces a TSCTR shadow load of current ECAP module and any ECAP modules down-stream providing the SYNCO_SEL bits are 0,0. After writing a 1, this bit returns to a zero.
7-6	SYNCO_SEL	R/W	0h	Sync-Out Select 0h (R/W) = Select sync-in event to be the sync-out signal (pass through) 1h (R/W) = Select PRDEQ event to be the sync-out signal 2h (R/W) = Disable sync out signal 3h (R/W) = Disable sync out signal
5	SYNCI_EN	R/W	0h	Counter (TSCTR) Sync-In select mode 0h (R/W) = Disable sync-in option 1h (R/W) = Enable counter (TSCTR) to be loaded from CTRPHS register upon either a SYNCI signal or a S/W force event.
4	TSCTRSTOP	R/W	0h	Time Stamp (TSCTR) Counter Stop (freeze) Control 0h (R/W) = TSCTR stopped 1h (R/W) = TSCTR free-running

Table 20-109. ECCTL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	REARM	R/W	0h	One-Shot Re-Arming Control, that is, wait for stop trigger. Note: The re-arm function is valid in one shot or continuous mode. 0h (R/W) = Has no effect (reading always returns a 0) 1h (R/W) = Arms the one-shot sequence as follows: 1) Resets the Mod4 counter to zero. 2) Unfreezes the Mod4 counter. 3) Enables capture register loads.
2-1	STOP_WRAP	R/W	3h	Stop value for one-shot mode. This is the number (between 1 and 4) of captures allowed to occur before the CAP (1 through 4) registers are frozen, that is, capture sequence is stopped. Wrap value for continuous mode. This is the number (between 1 and 4) of the capture register in which the circular buffer wraps around and starts again. Notes: STOP_WRAP is compared to Mod4 counter and, when equal, the following two actions occur. (1) Mod4 counter is stopped (frozen), and (2) Capture register loads are inhibited. In one-shot mode, further interrupt events are blocked until re-armed. 0h (R/W) = Stop after Capture Event 1 in one-shot mode. Wrap after Capture Event 1 in continuous mode. 1h (R/W) = Stop after Capture Event 2 in one-shot mode. Wrap after Capture Event 2 in continuous mode. 2h (R/W) = Stop after Capture Event 3 in one-shot mode. Wrap after Capture Event 3 in continuous mode. 3h (R/W) = Stop after Capture Event 4 in one-shot mode. Wrap after Capture Event 4 in continuous mode.
0	CONT_ONESHT	R/W	0h	Continuous or one-shot mode control (applicable only in capture mode) 0h (R/W) = Operate in continuous mode 1h (R/W) = Operate in one-shot mode

20.3.5.9 ECEINT Register (Offset = 2Ch) [reset = 0h]

ECEINT is shown in Figure 20-125 and described in Table 20-110.

Return to Summary Table.

The interrupt enable bits (CEVTn) block any of the selected events from generating an interrupt. Events will still be latched into the flag bit (ECFLG register) and can be forced or cleared via the ECFRC and ECCLR registers. The proper procedure for configuring peripheral modes and interrupts is: 1. Disable global interrupts. 2. Stop eCAP counter. 3. Disable eCAP interrupts. 4. Configure peripheral registers. 5. Clear spurious eCAP interrupt flags. 6. Enable eCAP interrupts. 7. Start eCAP counter. 8. Enable global interrupts.

Figure 20-125. ECEINT Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
CMPEQ	PRDEQ	CNTOVF	CEVT4	CEVT3	CEVT2	CEVT1	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 20-110. ECEINT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7	CMPEQ	R/W	0h	Counter Equal Compare Interrupt Enable. 0h (R/W) = Disable Compare Equal as an Interrupt source. 1h (R/W) = Enable Compare Equal as an Interrupt source.
6	PRDEQ	R/W	0h	Counter Equal Period Interrupt Enable. 0h (R/W) = Disable Period Equal as an Interrupt source. 1h (R/W) = Enable Period Equal as an Interrupt source.
5	CNTOVF	R/W	0h	Counter Overflow Interrupt Enable. 0h (R/W) = Disable counter Overflow as an Interrupt source. 1h (R/W) = Enable counter Overflow as an Interrupt source.
4	CEVT4	R/W	0h	Capture Event 4 Interrupt Enable. 0h (R/W) = Disable Capture Event 4 as an Interrupt source. 1h (R/W) = Enable Capture Event 4 as an Interrupt source.
3	CEVT3	R/W	0h	Capture Event 3 Interrupt Enable. 0h (R/W) = Disable Capture Event 3 as an Interrupt source. 1h (R/W) = Enable Capture Event 3 as an Interrupt source.
2	CEVT2	R/W	0h	Capture Event 2 Interrupt Enable. 0h (R/W) = Disable Capture Event 2 as an Interrupt source. 1h (R/W) = Enable Capture Event 2 as an Interrupt source.
1	CEVT1	R/W	0h	Capture Event 1 Interrupt Enable . 0h (R/W) = Disable Capture Event 1 as an Interrupt source. 1h (R/W) = Enable Capture Event 1 as an Interrupt source.
0	RESERVED	R	0h	

20.3.5.10 ECFLG Register (Offset = 2Eh) [reset = 0h]

ECFLG is shown in Figure 20-126 and described in Table 20-111.

Return to Summary Table.

Figure 20-126. ECFLG Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMPEQ	PRDEQ	CNTOVF	CEVT4	CEVT3	CEVT2	CEVT1	INT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-111. ECFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7	CMPEQ	R	0h	Compare Equal Compare Status Flag. This flag is only active in APWM mode. 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the counter (TSCTR) reached the compare register value (ACMP)
6	PRDEQ	R	0h	Counter Equal Period Status Flag. This flag is only active in APWM mode. 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the counter (TSCTR) reached the period register value (APRD) and was reset.
5	CNTOVF	R	0h	Counter Overflow Status Flag. This flag is active in CAP and APWM mode. 0h (R/W) = Indicates no event occurred. 1h (R/W) = Indicates the counter (TSCTR) has made the transition from 0xFFFFFFFF to 0x00000000
4	CEVT4	R	0h	Capture Event 4 Status Flag This flag is only active in CAP mode. 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the fourth event occurred at ECAPn pin
3	CEVT3	R	0h	Capture Event 3 Status Flag. This flag is active only in CAP mode. 0h (R/W) = Indicates no event occurred. 1h (R/W) = Indicates the third event occurred at ECAPn pin.
2	CEVT2	R	0h	Capture Event 2 Status Flag. This flag is only active in CAP mode. 0h (R/W) = Indicates no event occurred. 1h (R/W) = Indicates the second event occurred at ECAPn pin.
1	CEVT1	R	0h	Capture Event 1 Status Flag. This flag is only active in CAP mode. 0h (R/W) = Indicates no event occurred. 1h (R/W) = Indicates the first event occurred at ECAPn pin.
0	INT	R	0h	Global Interrupt Status Flag 0h (R/W) = Indicates no interrupt generated. 1h (R/W) = Indicates that an interrupt was generated.

20.3.5.11 ECCLR Register (Offset = 30h) [reset = 0h]

ECCLR is shown in Figure 20-127 and described in Table 20-112.

Return to Summary Table.

Figure 20-127. ECCLR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMPEQ	PRDEQ	CNTOVF	CEVT4	CEVT3	CEVT2	CEVT1	INT
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-112. ECCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7	CMPEQ	R/W	0h	Counter Equal Compare Status Flag 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CMPEQ flag condition
6	PRDEQ	R/W	0h	Counter Equal Period Status Flag 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the PRDEQ flag condition
5	CNTOVF	R/W	0h	Counter Overflow Status Flag 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CNTOVF flag condition
4	CEVT4	R/W	0h	Capture Event 4 Status Flag 0h (R/W) = Writing a 0 has no effect. Always reads back a 0. 1h (R/W) = Writing a 1 clears the CEVT3 flag condition.
3	CEVT3	R/W	0h	Capture Event 3 Status Flag 0h (R/W) = Writing a 0 has no effect. Always reads back a 0. 1h (R/W) = Writing a 1 clears the CEVT3 flag condition.
2	CEVT2	R/W	0h	Capture Event 2 Status Flag 0h (R/W) = Writing a 0 has no effect. Always reads back a 0. 1h (R/W) = Writing a 1 clears the CEVT2 flag condition.
1	CEVT1	R/W	0h	Capture Event 1 Status Flag 0h (R/W) = Writing a 0 has no effect. Always reads back a 0. 1h (R/W) = Writing a 1 clears the CEVT1 flag condition.
0	INT	R/W	0h	Global Interrupt Clear Flag 0h (R/W) = Writing a 0 has no effect. Always reads back a 0. 1h (R/W) = Writing a 1 clears the INT flag and enable further interrupts to be generated if any of the event flags are set to 1.

20.3.5.12 ECFRC Register (Offset = 32h) [reset = 0h]

ECFRC is shown in Figure 20-128 and described in Table 20-113.

Return to Summary Table.

Figure 20-128. ECFRC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
CMPEQ	PRDEQ	CNTOVF	CEVT4	CEVT3	CEVT2	CEVT1	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 20-113. ECFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7	CMPEQ	R/W	0h	Force Counter Equal Compare Interrupt 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the CMPEQ flag bit.
6	PRDEQ	R/W	0h	Force Counter Equal Period Interrupt 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the PRDEQ flag bit.
5	CNTOVF	R/W	0h	Force Counter Overflow 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 to this bit sets the CNTOVF flag bit.
4	CEVT4	R/W	0h	Force Capture Event 4 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the CEVT4 flag bit
3	CEVT3	R/W	0h	Force Capture Event 3 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the CEVT3 flag bit
2	CEVT2	R/W	0h	Force Capture Event 2 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the CEVT2 flag bit.
1	CEVT1	R/W	0h	Always reads back a 0. Force Capture Event 1 0h (R/W) = No effect. 1h (R/W) = Writing a 1 sets the CEVT1 flag bit.
0	RESERVED	R	0h	

20.3.5.13 REVID Register (Offset = 5Ch) [reset = 44D22100h]

REVID is shown in Figure 20-174 and described in Table 20-141.

Return to Summary Table.

Figure 20-129. REVID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REV																															
R-44D22100h																															

Table 20-114. REVID Register Field Descriptions

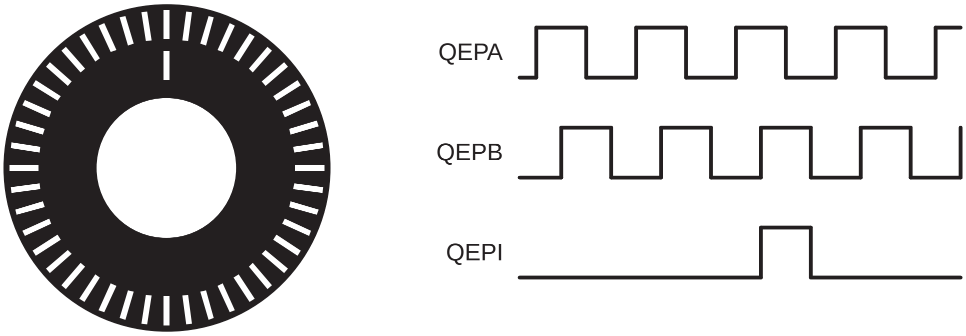
Bit	Field	Type	Reset	Description
31-0	REV	R	44D22100h	Revision ID.

20.4 Enhanced Quadrature Encoder Pulse (eQEP) Module

20.4.1 Introduction

A single track of slots patterns the periphery of an incremental encoder disk, as shown in Figure 20-130. These slots create an alternating pattern of dark and light lines. The disk count is defined as the number of dark/light line pairs that occur per revolution (lines per revolution). As a rule, a second track is added to generate a signal that occurs once per revolution (index signal: QEPI), which can be used to indicate an absolute position. Encoder manufacturers identify the index pulse using different terms such as index, marker, home position, and zero reference.

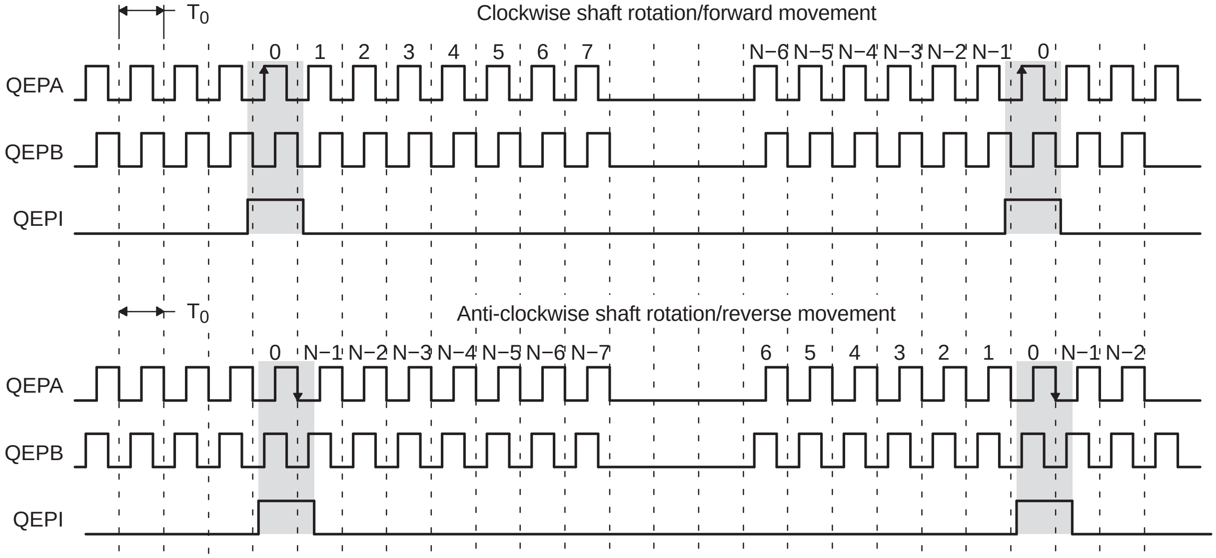
Figure 20-130. Optical Encoder Disk



To derive direction information, the lines on the disk are read out by two different photo-elements that "look" at the disk pattern with a mechanical shift of 1/4 the pitch of a line pair between them. This shift is realized with a reticle or mask that restricts the view of the photo-element to the desired part of the disk lines. As the disk rotates, the two photo-elements generate signals that are shifted 90 degrees out of phase from each other. These are commonly called the quadrature QEPA and QEPB signals. The clockwise direction for most encoders is defined as the QEPA channel going positive before the QEPB channel and vise versa as shown in Figure 20-131.

The encoder wheel typically makes one revolution for every revolution of the motor or the wheel may be at a geared rotation ratio with respect to the motor. Therefore, the frequency of the digital signal coming from the QEPA and QEPB outputs varies proportionally with the velocity of the motor. For example, a 2000-line encoder directly coupled to a motor running at 5000 revolutions per minute (rpm) results in a frequency of 166.6 KHz, so by measuring the frequency of either the QEPA or QEPB output, the processor can determine the velocity of the motor.

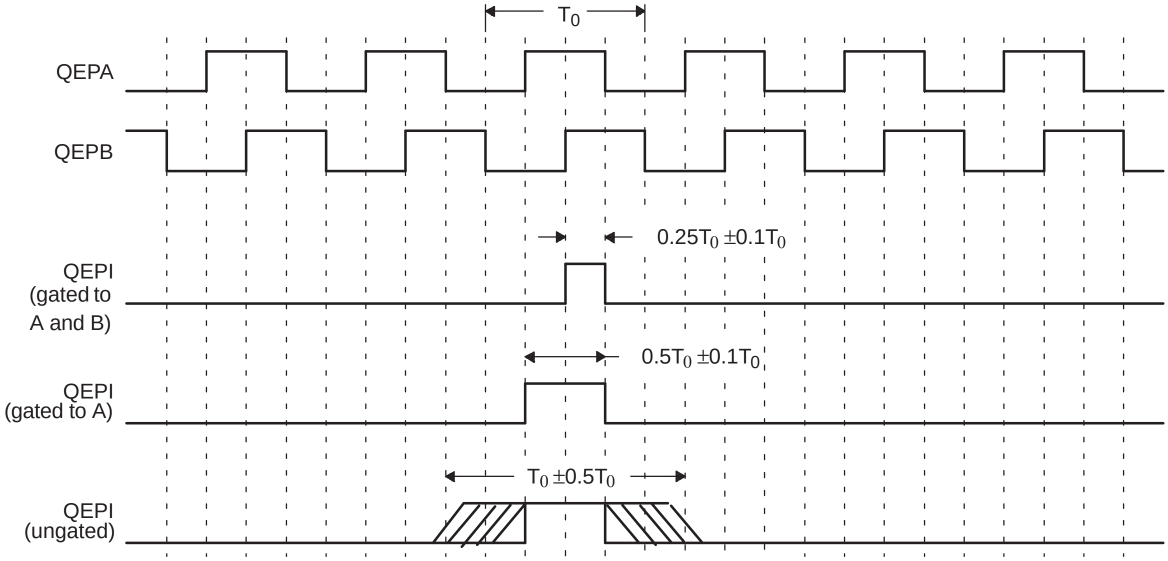
Figure 20-131. QEP Encoder Output Signal for Forward/Reverse Movement



Legend: N = lines per revolution

Quadrature encoders from different manufacturers come with two forms of index pulse (gated index pulse or ungated index pulse) as shown in Figure 20-132. A nonstandard form of index pulse is ungated. In the ungated configuration, the index edges are not necessarily coincident with A and B signals. The gated index pulse is aligned to any of the four quadrature edges and width of the index pulse and can be equal to a quarter, half, or full period of the quadrature signal.

Figure 20-132. Index Pulse Example



Some typical applications of shaft encoders include robotics and even computer input in the form of a mouse. Inside your mouse you can see where the mouse ball spins a pair of axles (a left/right, and an up/down axle). These axles are connected to optical shaft encoders that effectively tell the computer how fast and in what direction the mouse is moving.

General Issues: Estimating velocity from a digital position sensor is a cost-effective strategy in motor control. Two different first order approximations for velocity may be written as:

$$v(k) \approx \frac{x(k) - x(k-1)}{T} = \frac{\Delta X}{T} \qquad (22)$$

$$v(k) \approx \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T} \qquad (23)$$

where

v(k): Velocity at time instant k

x(k): Position at time instant k

x(k-1): Position at time instant k - 1

T: Fixed unit time or inverse of velocity calculation rate

ΔX: Incremental position movement in unit time

t(k): Time instant "k"

t(k-1): Time instant "k - 1"

X: Fixed unit position

ΔT: Incremental time elapsed for unit position movement.

Equation 22 is the conventional approach to velocity estimation and it requires a time base to provide unit time event for velocity calculation. Unit time is basically the inverse of the velocity calculation rate.

The encoder count (position) is read once during each unit time event. The quantity [x(k) - x(k-1)] is formed by subtracting the previous reading from the current reading. Then the velocity estimate is computed by multiplying by the known constant 1/T (where T is the constant time between unit time events and is known in advance).

Estimation based on Equation 22 has an inherent accuracy limit directly related to the resolution of the position sensor and the unit time period T. For example, consider a 500-line per revolution quadrature encoder with a velocity calculation rate of 400 Hz. When used for position the quadrature encoder gives a four-fold increase in resolution, in this case, 2000 counts per revolution. The minimum rotation that can be detected is therefore 0.0005 revolutions, which gives a velocity resolution of 12 rpm when sampled at 400 Hz. While this resolution may be satisfactory at moderate or high speeds, for example, 1% error at 1200 rpm, it would clearly prove inadequate at low speeds. In fact, at speeds below 12 rpm, the speed estimate would erroneously be zero much of the time.

At low speed, Equation 23 provides a more accurate approach. It requires a position sensor that outputs a fixed interval pulse train, such as the aforementioned quadrature encoder. The width of each pulse is defined by motor speed for a given sensor resolution. Equation 23 can be used to calculate motor speed by measuring the elapsed time between successive quadrature pulse edges. However, this method suffers from the opposite limitation, as does Equation 22. A combination of relatively large motor speeds and high sensor resolution makes the time interval ΔT small, and thus more greatly influenced by the timer resolution. This can introduce considerable error into high-speed estimates.

For systems with a large speed range (that is, speed estimation is needed at both low and high speeds), one approach is to use Equation 23 at low speed and have the software switch over to Equation 22 when the motor speed rises above some specified threshold.

20.4.2 Functional Description

This section provides the eQEP inputs and functional description.

NOTE: Multiple identical eQEP modules can be contained in a system. The number of modules is device-dependent and is based on target application needs. In this document, the letter x within a signal or module name is used to indicate a generic eQEP instance on a device.

20.4.2.1 EQEP Inputs

The eQEP inputs include two pins for quadrature-clock mode or direction-count mode, an index (or 0 marker), and a strobe input.

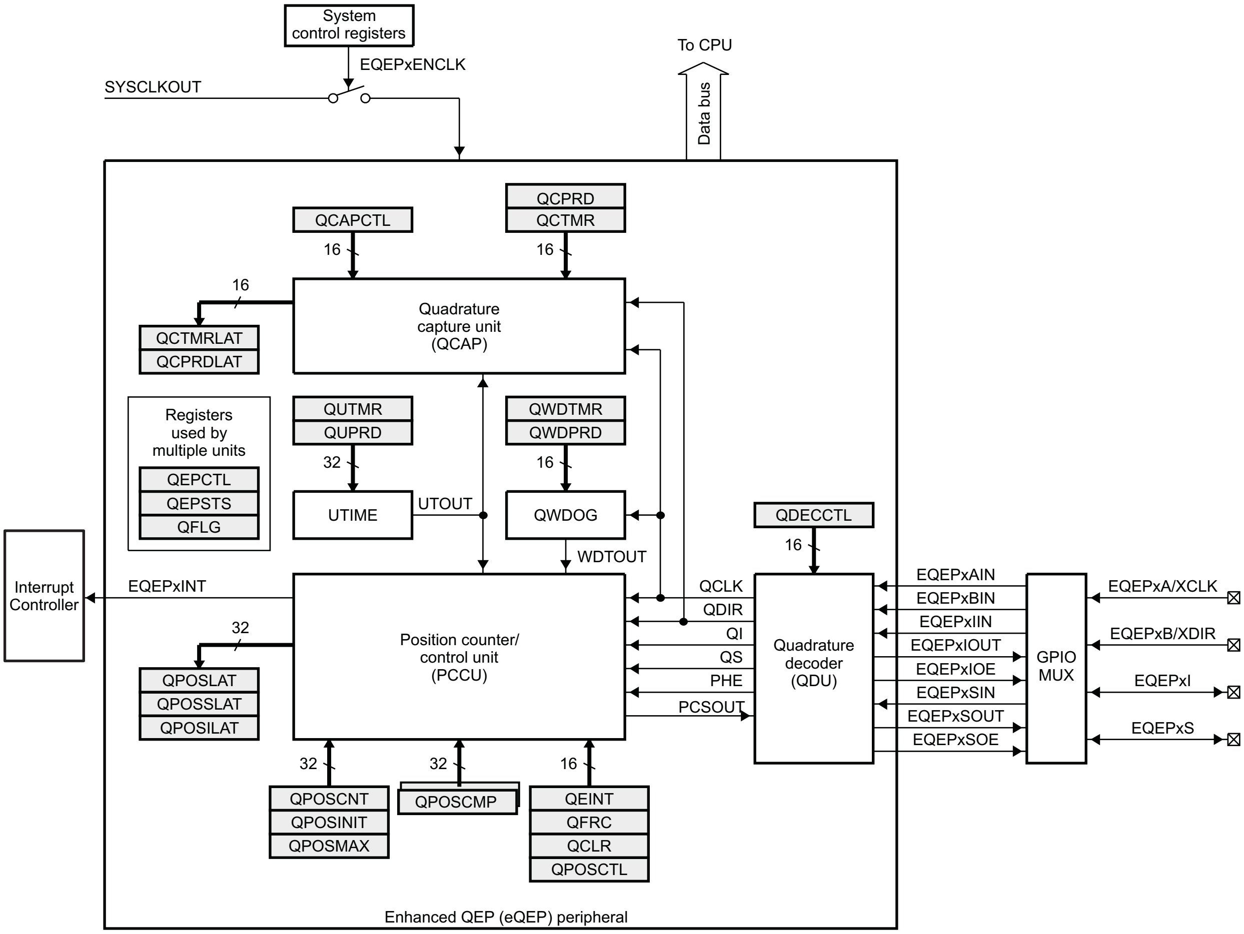
- QEPA/XCLK and QEPB/XDIR: These two pins can be used in quadrature-clock mode or direction-count mode.
  - Quadrature-clock Mode: The eQEP encoders provide two square wave signals (A and B) 90 electrical degrees out of phase whose phase relationship is used to determine the direction of rotation of the input shaft and number of eQEP pulses from the index position to derive the relative position information. For forward or clockwise rotation, QEPA signal leads QEPB signal and vice versa. The quadrature decoder uses these two inputs to generate quadrature-clock and direction signals.
  - Direction-count Mode: In direction-count mode, direction and clock signals are provided directly from the external source. Some position encoders have this type of output instead of quadrature output. The QEPA pin provides the clock input and the QEPB pin provides the direction input.
- QEPI: Index or Zero Marker: The eQEP encoder uses an index signal to assign an absolute start position from which position information is incrementally encoded using quadrature pulses. This pin is connected to the index output of the eQEP encoder to optionally reset the position counter for each revolution. This signal can be used to initialize or latch the position counter on the occurrence of a desired event on the index pin.
- QEPS: Strobe Input: This general-purpose strobe signal can initialize or latch the position counter on the occurrence of a desired event on the strobe pin. This signal is typically connected to a sensor or limit switch to notify that the motor has reached a defined position.

20.4.2.2 Functional Description

The eQEP peripheral contains the following major functional units (as shown in Figure 20-133):

- Programmable input qualification for each pin (part of the GPIO MUX)
- Quadrature decoder unit (QDU)
- Position counter and control unit for position measurement (PCCU)
- Quadrature edge-capture unit for low-speed measurement (QCAP)
- Unit time base for speed/frequency measurement (UTIME)
- Watchdog timer for detecting stalls (QWDOG)

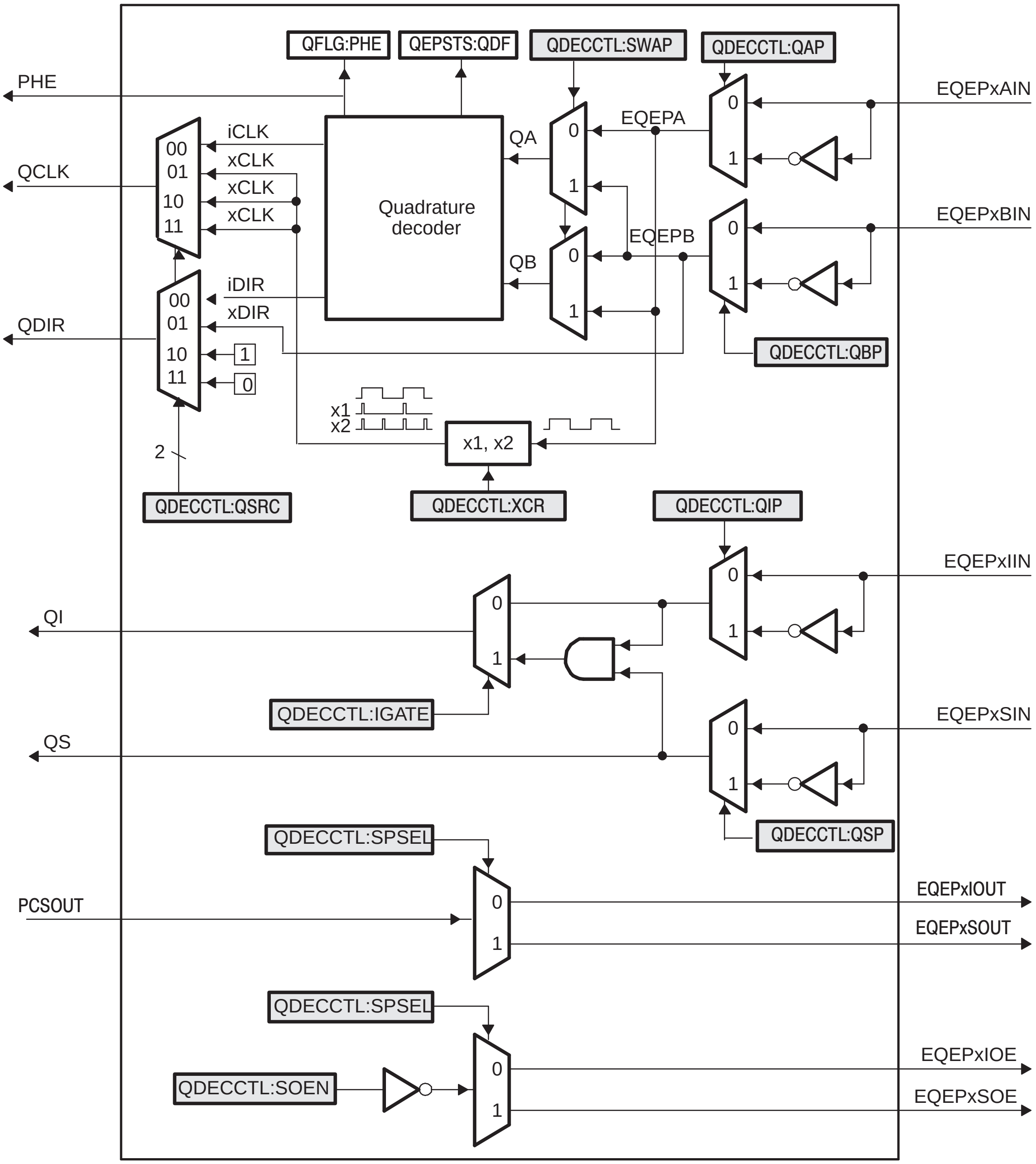
Figure 20-133. Functional Block Diagram of the eQEP Peripheral



20.4.2.3 Quadrature Decoder Unit (QDU)

Figure 20-134 shows a functional block diagram of the QDU.

Figure 20-134. Functional Block Diagram of Decoder Unit



20.4.2.3.1 Position Counter Input Modes

Clock and direction input to position counter is selected using the QSRC bit in the eQEP decoder control register (QDECCTL), based on interface input requirement as follows:

- Quadrature-count mode
- Direction-count mode
- UP-count mode
- DOWN-count mode

20.4.2.3.1.1 Quadrature Count Mode

The quadrature decoder generates the direction and clock to the position counter in quadrature count mode.

Direction Decoding— The direction decoding logic of the eQEP circuit determines which one of the sequences (QEPA, QEPB) is the leading sequence and accordingly updates the direction information in the QDF bit in the eQEP status register (QEPSTS). Table 20-115 and Figure 20-135 show the direction decoding logic in truth table and state machine form. Both edges of the QEPA and QEPB signals are sensed to generate count pulses for the position counter. Therefore, the frequency of the clock generated by the eQEP logic is four times that of each input sequence. Figure 20-136 shows the direction decoding and clock generation from the eQEP input signals.

Phase Error Flag— In normal operating conditions, quadrature inputs QEPA and QEPB will be 90 degrees out of phase. The phase error flag (PHE) is set in the QFLG register when edge transition is detected simultaneously on the QEPA and QEPB signals to optionally generate interrupts. State transitions marked by dashed lines in Figure 20-135 are invalid transitions that generate a phase error.

Count Multiplication— The eQEP position counter provides 4x times the resolution of an input clock by generating a quadrature-clock (QCLK) on the rising/falling edges of both eQEP input clocks (QEPA and QEPB) as shown in Figure 20-136.

Reverse Count— In normal quadrature count operation, QEPA input is fed to the QA input of the quadrature decoder and the QEPB input is fed to the QB input of the quadrature decoder. Reverse counting is enabled by setting the SWAP bit in the eQEP decoder control register (QDECCTL). This will swap the input to the quadrature decoder thereby reversing the counting direction.

Table 20-115. Quadrature Decoder Truth Table

Previous Edge	Present Edge	QDIR	QPOSCNT
QA↑	QB↑	UP	Increment
	QB↓	DOWN	Decrement
	QA↓	TOGGLE	Increment or Decrement
QA↓	QB↓	UP	Increment
	QB↑	DOWN	Decrement
	QA↑	TOGGLE	Increment or Decrement
QB↑	QA↑	DOWN	Increment
	QA↓	UP	Decrement
	QB↓	TOGGLE	Increment or Decrement
QB↓	QA↓	DOWN	Increment
	QA↑	UP	Decrement
	QB↑	TOGGLE	Increment or Decrement

Figure 20-135. Quadrature Decoder State Machine

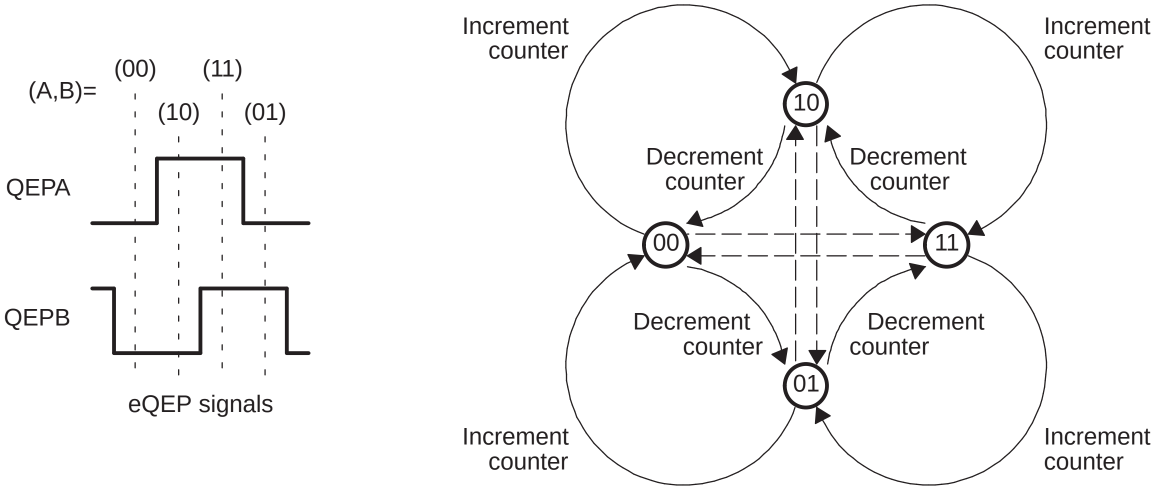
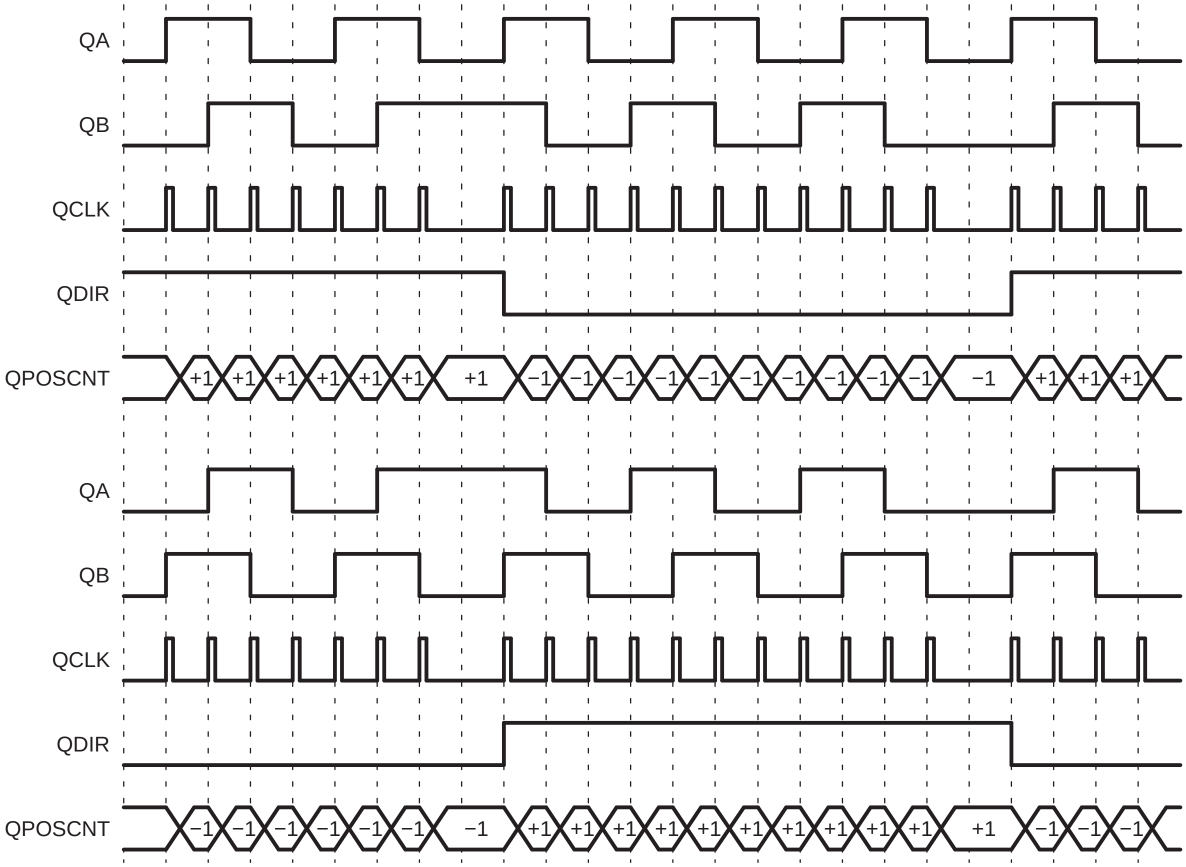


Figure 20-136. Quadrature-clock and Direction Decoding



20.4.2.3.1.2 Direction-count Mode

Some position encoders provide direction and clock outputs, instead of quadrature outputs. In such cases, direction-count mode can be used. QEPA input will provide the clock for position counter and the QEPB input will have the direction information. The position counter is incremented on every rising edge of a QEPA input when the direction input is high and decremented when the direction input is low.

20.4.2.3.1.3 Up-Count Mode

The counter direction signal is hard-wired for up count and the position counter is used to measure the frequency of the QEPA input. Setting of the XCR bit in the eQEP decoder control register (QDECCTL) enables clock generation to the position counter on both edges of the QEPA input, thereby increasing the measurement resolution by 2× factor.

20.4.2.3.1.4 Down-Count Mode

The counter direction signal is hardwired for a down count and the position counter is used to measure the frequency of the QEPA input. Setting of the XCR bit in the eQEP decoder control register (QDECCTL) enables clock generation to the position counter on both edges of a QEPA input, thereby increasing the measurement resolution by 2× factor.

20.4.2.3.2 eQEP Input Polarity Selection

Each eQEP input can be inverted using the in the eQEP decoder control register (QDECCTL[8:5]) control bits. As an example, setting of the QIP bit in QDECCTL inverts the index input.

20.4.2.3.3 Position-Compare Sync Output

The eQEP peripheral includes a position-compare unit that is used to generate the position-compare sync signal on compare match between the position counter register (QPOSCNT) and the position-compare register (QPOSCMP). This sync signal can be output using an index pin or strobe pin of the EQEP peripheral.

Setting the SOEN bit in the eQEP decoder control register (QDECCTL) enables the position-compare sync output and the SPSEL bit in QDECCTL selects either an eQEP index pin or an eQEP strobe pin.

20.4.2.4 Position Counter and Control Unit (PCCU)

The position counter and control unit provides two configuration registers (QEPCTL and QPOSCTL) for setting up position counter operational modes, position counter initialization/latch modes and position-compare logic for sync signal generation.

20.4.2.4.1 Position Counter Operating Modes

Position counter data may be captured in different manners. In some systems, the position counter is accumulated continuously for multiple revolutions and the position counter value provides the position information with respect to the known reference. An example of this is the quadrature encoder mounted on the motor controlling the print head in the printer. Here the position counter is reset by moving the print head to the home position and then position counter provides absolute position information with respect to home position.

In other systems, the position counter is reset on every revolution using index pulse and position counter provides rotor angle with respect to index pulse position.

Position counter can be configured to operate in following four modes

- Position Counter Reset on Index Event
- Position Counter Reset on Maximum Position
- Position Counter Reset on the first Index Event
- Position Counter Reset on Unit Time Out Event (Frequency Measurement)

In all the above operating modes, position counter is reset to 0 on overflow and to QPOSMAX register value on underflow. Overflow occurs when the position counter counts up after QPOSMAX value. Underflow occurs when position counter counts down after "0". Interrupt flag is set to indicate overflow/underflow in QFLG register.

20.4.2.4.1.1 Position Counter Reset on Index Event (QEPCTL[PCRM] = 00)

If the index event occurs during the forward movement, then position counter is reset to 0 on the next eQEP clock. If the index event occurs during the reverse movement, then the position counter is reset to the value in the QPOSMAX register on the next eQEP clock.

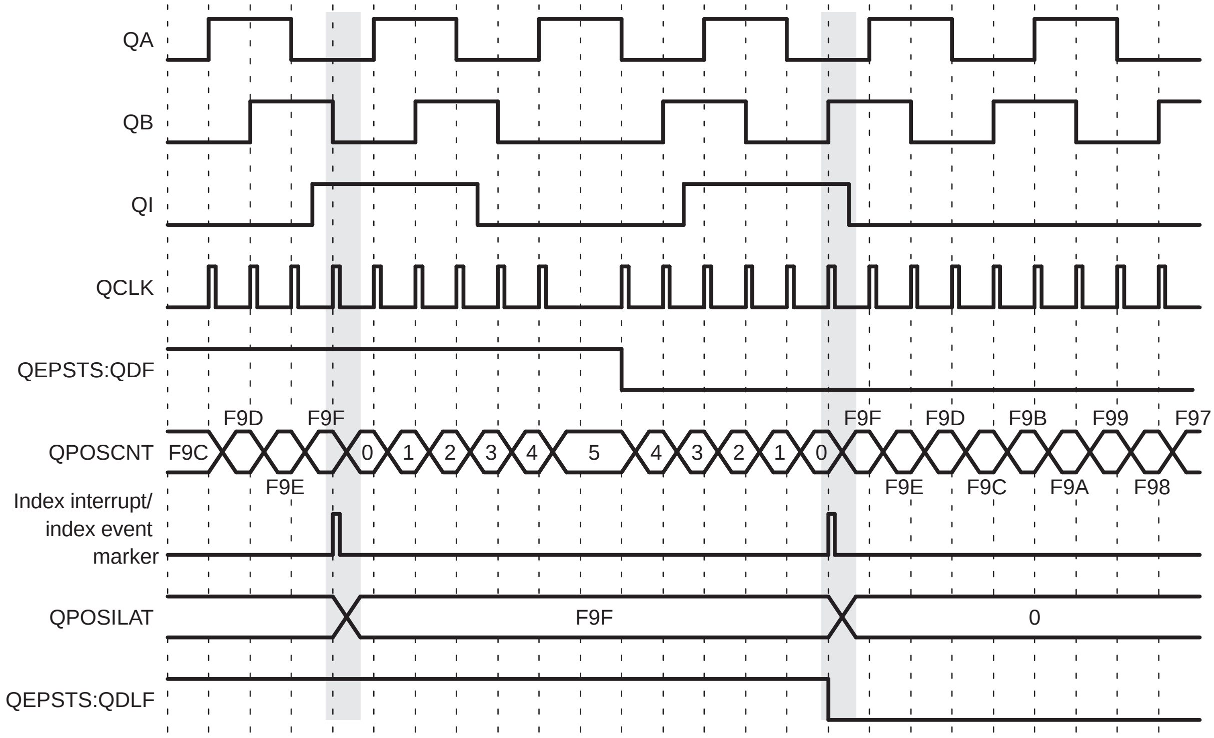
First index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (QEPSTS[FIMF]) and direction on the first index event marker (QEPSTS[FIDF]) in QEPSTS registers, it also remembers the quadrature edge on the first index marker so that same relative quadrature transition is used for index event reset operation.

For example, if the first reset operation occurs on the falling edge of QEPB during the forward direction, then all the subsequent reset must be aligned with the falling edge of QEPB for the forward rotation and on the rising edge of QEPB for the reverse rotation as shown in Figure 20-137.

The position-counter value is latched to the QPOSILAT register and direction information is recorded in the QEPSTS[QDLF] bit on every index event marker. The position-counter error flag (QEPSTS[PCEF]) and error interrupt flag (QFLG[PCE]) are set if the latched value is not equal to 0 or QPOSMAX. The position-counter error flag (QEPSTS[PCEF]) is updated on every index event marker and an interrupt flag (QFLG[PCE]) will be set on error that can be cleared only through software.

The index event latch configuration QEPCTL[IEL] bits are ignored in this mode and position counter error flag/interrupt flag are generated only in index event reset mode.

Figure 20-137. Position Counter Reset by Index Pulse for 1000 Line Encoder (QPOSMAX = 3999 or F9Fh)

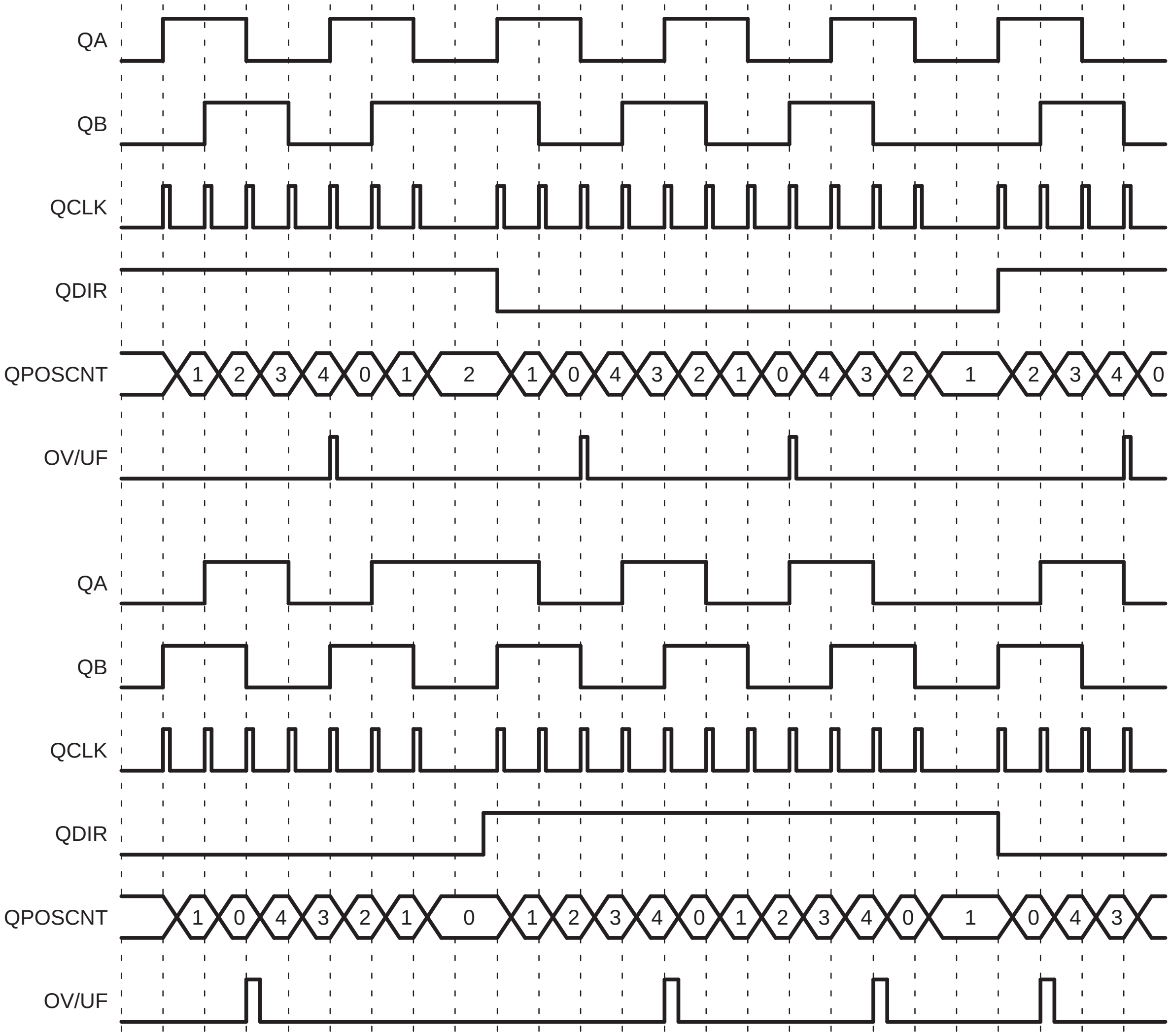


20.4.2.4.1.2 Position Counter Reset on Maximum Position (QEPCTL[PCRM]=01)

If the position counter is equal to QPOSMAX, then the position counter is reset to 0 on the next eQEP clock for forward movement and position counter overflow flag is set. If the position counter is equal to ZERO, then the position counter is reset to QPOSMAX on the next QEP clock for reverse movement and position counter underflow flag is set. Figure 20-138 shows the position counter reset operation in this mode.

First index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (QEPSTS[FIMF]) and direction on the first index event marker (QEPSTS[FIDF]) in the QEPSTS registers; it also remembers the quadrature edge on the first index marker so that the same relative quadrature transition is used for the software index marker (QEPCTL[IEL]=11).

Figure 20-138. Position Counter Underflow/Overflow (QPOSMAX = 4)



20.4.2.4.1.3 Position Counter Reset on the First Index Event (QEPCTL[PCRM] = 10)

If the index event occurs during forward movement, then the position counter is reset to 0 on the next eQEP clock. If the index event occurs during the reverse movement, then the position counter is reset to the value in the QPOSMAX register on the next eQEP clock. Note that this is done only on the first occurrence and subsequently the position counter value is not reset on an index event; rather, it is reset based on maximum position as described in Section 20.4.2.4.1.2.

First index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (QEPSTS[FIMF]) and direction on the first index event marker (QEPSTS[FIDF]) in QEPSTS registers. It also remembers the quadrature edge on the first index marker so that same relative quadrature transition is used for software index marker (QEPCTL[IEL]=11).

20.4.2.4.1.4 Position Counter Reset on Unit Time out Event (QEPCTL[PCRM] = 11)

In this mode, the QPOSCNT value is latched to the QPOSLAT register and then the QPOSCNT is reset (to 0 or QPOSMAX, depending on the direction mode selected by QDECCTL[QSRC] bits on a unit time event). This is useful for frequency measurement.

20.4.2.4.2 Position Counter Latch

The eQEP index and strobe input can be configured to latch the position counter (QPOSCNT) into QPOSILAT and QPOSSLAT, respectively, on occurrence of a definite event on these pins.

20.4.2.4.2.1 Index Event Latch

In some applications, it may not be desirable to reset the position counter on every index event and instead it may be required to operate the position counter in full 32-bit mode (QEPCTL[PCRM] = 01 and QEPCTL[PCRM] = 10 modes).

In such cases, the eQEP position counter can be configured to latch on the following events and direction information is recorded in the QEPSTS[QDLF] bit on every index event marker.

- Latch on Rising edge (QEPCTL[IEL] = 01)
- Latch on Falling edge (QEPCTL[IEL] = 10)
- Latch on Index Event Marker (QEPCTL[IEL] = 11)

This is particularly useful as an error checking mechanism to check if the position counter accumulated the correct number of counts between index events. As an example, the 1000-line encoder must count 4000 times when moving in the same direction between the index events.

The index event latch interrupt flag (QFLG[IEL]) is set when the position counter is latched to the QPOSILAT register. The index event latch configuration bits (QEPCTZ[IEL]) are ignored when QEPCTL[PCRM] = 00.

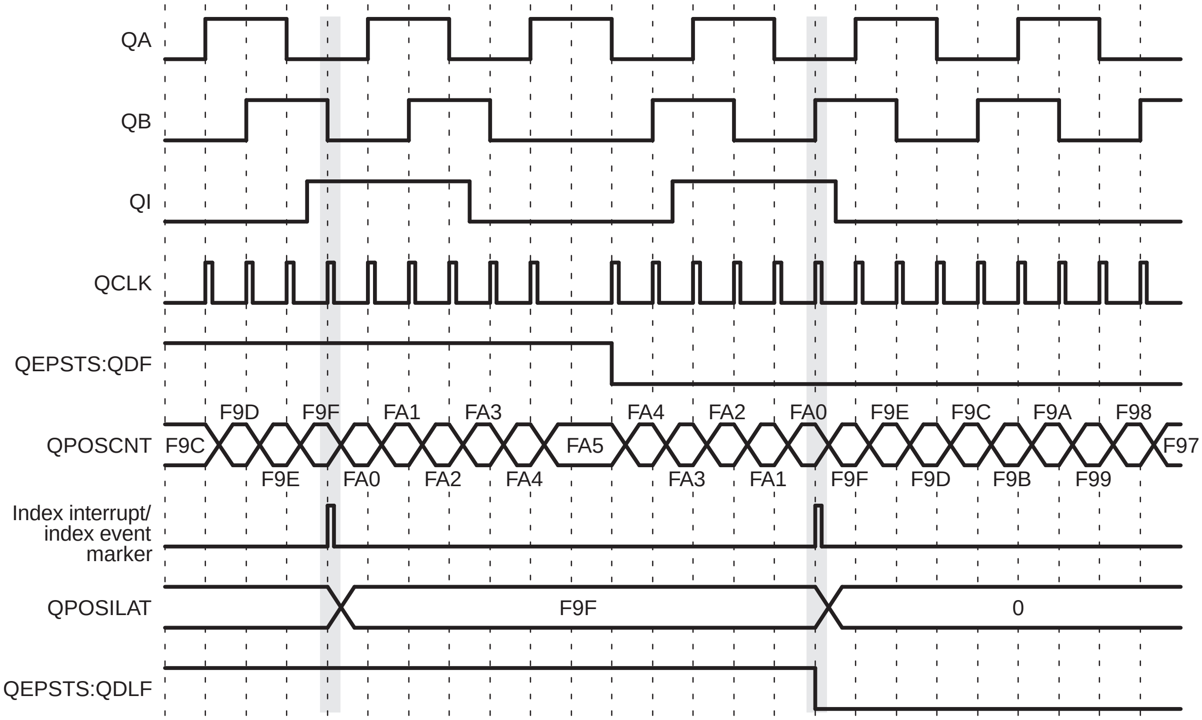
Latch on Rising Edge (QEPCTL[IEL] = 01)— The position counter value (QPOSCNT) is latched to the QPOSILAT register on every rising edge of an index input.

Latch on Falling Edge (QEPCTL[IEL] = 10)— The position counter value (QPOSCNT) is latched to the QPOSILAT register on every falling edge of index input.

Latch on Index Event Marker/Software Index Marker (QEPCTL[IEL] = 11)— The first index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (QEPSTS[FIMF]) and direction on the first index event marker (QEPSTS[FIDF]) in the QEPSTS registers. It also remembers the quadrature edge on the first index marker so that same relative quadrature transition is used for latching the position counter (QEPCTL[IEL] = 11).

Figure 20-139 shows the position counter latch using an index event marker.

Figure 20-139. Software Index Marker for 1000-line Encoder (QEPCTL[IEL] = 1)



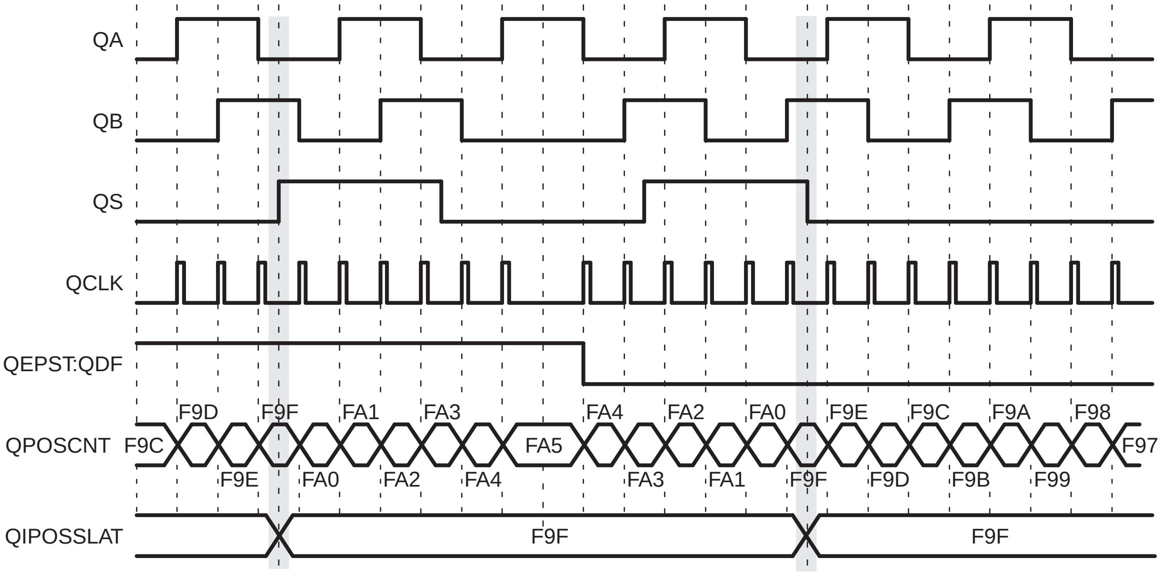
20.4.2.4.2.2 Strobe Event Latch

The position-counter value is latched to the QPOSSLAT register on the rising edge of the strobe input by clearing the QEPCTL[SEL] bit.

If the QEPCTL[SEL] bit is set, then the position counter value is latched to the QPOSSLAT register on the rising edge of the strobe input for forward direction and on the falling edge of the strobe input for reverse direction as shown in Figure 20-140.

The strobe event latch interrupt flag (QFLG[SEL]) is set when the position counter is latched to the QPOSSLAT register.

Figure 20-140. Strobe Event Latch (QEPCTL[SEL] = 1)



20.4.2.4.3 Position Counter Initialization

The position counter can be initialized using following events:

- Index event
- Strobe event
- Software initialization

Index Event Initialization (IEI)— The QEPI index input can be used to trigger the initialization of the position counter at the rising or falling edge of the index input.

If the QEPCTL[IEI] bits are 10, then the position counter (QPOSCNT) is initialized with a value in the QPOSINIT register on the rising edge of strobe input for forward direction and on the falling edge of strobe input for reverse direction.

The index event initialization interrupt flag (QFLG[IEI]) is set when the position counter is initialized with a value in the QPOSINIT register.

Strobe Event Initialization (SEI)— If the QEPCTL[SEI] bits are 10, then the position counter is initialized with a value in the QPOSINIT register on the rising edge of strobe input.

If the QEPCTL[SEL] bits are 11, then the position counter (QPOSCNT) is initialized with a value in the QPOSINIT register on the rising edge of strobe input for forward direction and on the falling edge of strobe input for reverse direction.

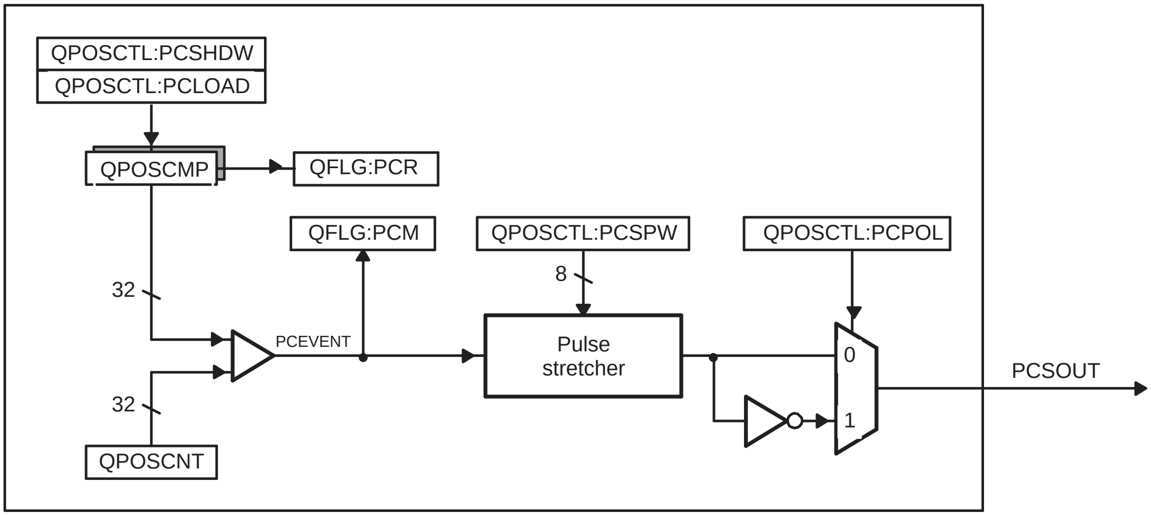
The strobe event initialization interrupt flag (QFLG[SEI]) is set when the position counter is initialized with a value in the QPOSINIT register.

Software Initialization (SWI)— The position counter can be initialized in software by writing a 1 to the QEPCTL[SWI] bit, which will automatically be cleared after initialization.

20.4.2.4.4 eQEP Position-compare Unit

The eQEP peripheral includes a position-compare unit that is used to generate a sync output and/or interrupt on a position-compare match. Figure 20-141 shows a diagram. The position-compare (QPOSCMP) register is shadowed and shadow mode can be enabled or disabled using the QPOSCTL[PSSHDW] bit. If the shadow mode is not enabled, the CPU writes directly to the active position compare register.

Figure 20-141. eQEP Position-compare Unit



In shadow mode, you can configure the position-compare unit (QPOSCTL[PCLOAD]) to load the shadow register value into the active register on the following events and to generate the position-compare ready (QFLG[PCR]) interrupt after loading.

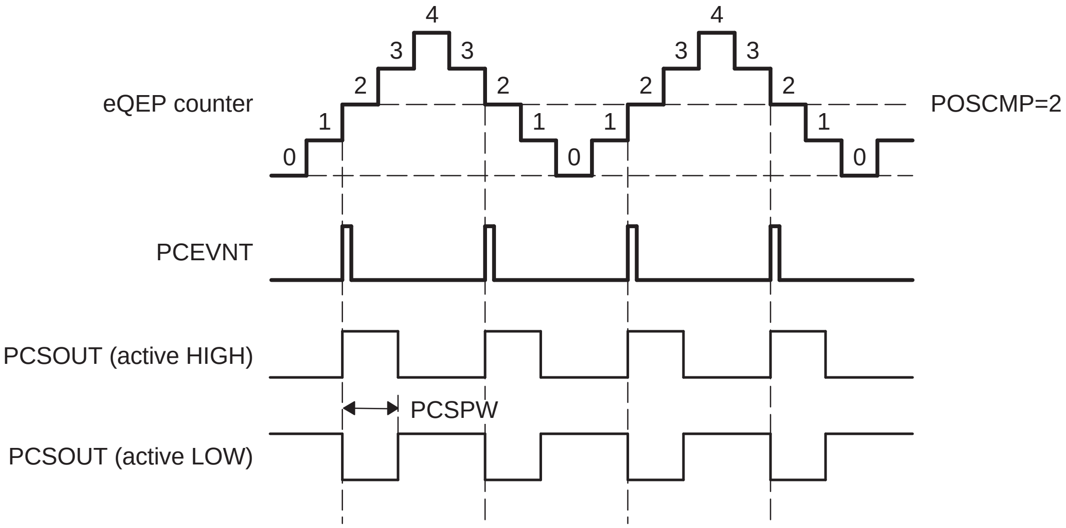
- Load on compare match
- Load on position-counter zero event

The position-compare match (QFLG[PCM]) is set when the position-counter value (QPOSCNT) matches with the active position-compare register (QPOSCMP) and the position-compare sync output of the programmable pulse width is generated on compare match to trigger an external device.

For example, if QPOSCMP = 2, the position-compare unit generates a position-compare event on 1 to 2 transitions of the eQEP position counter for forward counting direction and on 3 to 2 transitions of the eQEP position counter for reverse counting direction (see Figure 20-142).

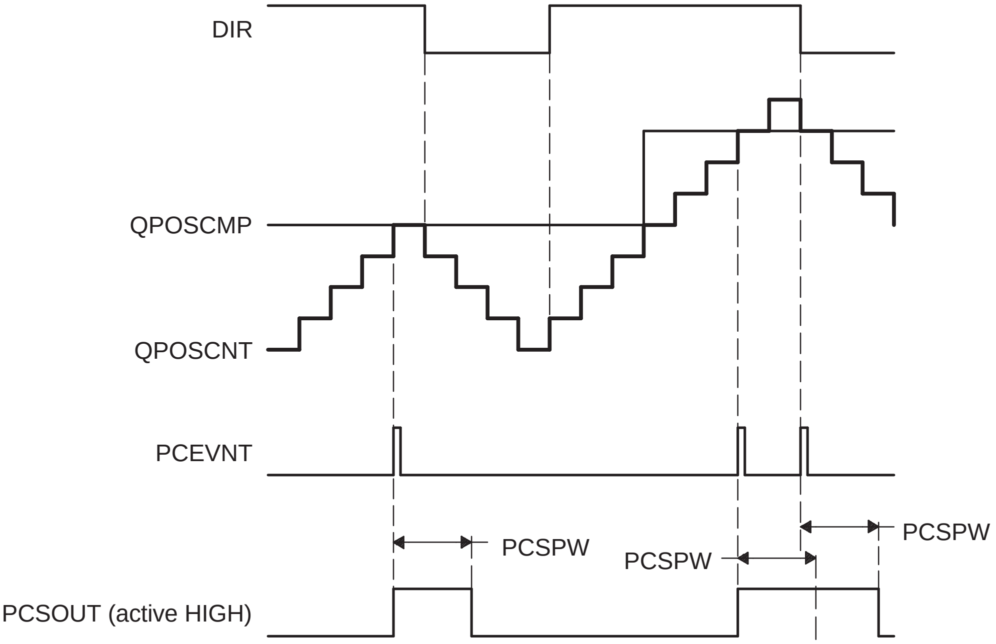
shows the layout of the eQEP Position-Compare Control Register (QPOSCTL) and describes the QPOSCTL bit fields.

Figure 20-142. eQEP Position-compare Event Generation Points



The pulse stretcher logic in the position-compare unit generates a programmable position-compare sync pulse output on the position-compare match. In the event of a new position-compare match while a previous position-compare pulse is still active, then the pulse stretcher generates a pulse of specified duration from the new position-compare event as shown in Figure 20-143.

Figure 20-143. eQEP Position-compare Sync Output Pulse Stretcher



20.4.2.5 eQEP Edge Capture Unit

The eQEP peripheral includes an integrated edge capture unit to measure the elapsed time between the unit position events as shown in Figure 20-144. This feature is typically used for low speed measurement using the following equation:

$$v(k) = \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T} \tag{24}$$

where,

- X - Unit position is defined by integer multiple of quadrature edges (see Figure 20-145)
- ΔT - Elapsed time between unit position events
- v(k) - Velocity at time instant "k"

The eQEP capture timer (QCTMR) runs from prescaled SYSCLKOUT and the prescaler is programmed by the QCAPCTL[CCPS] bits. The capture timer (QCTMR) value is latched into the capture period register (QCPRD) on every unit position event and then the capture timer is reset, a flag is set in QEPSTS[UPEVNT] to indicate that new value is latched into the QCPRD register. Software can check this status flag before reading the period register for low speed measurement and clear the flag by writing 1.

Time measurement (ΔT) between unit position events will be correct if the following conditions are met:

- No more than 65,535 counts have occurred between unit position events.
- No direction change between unit position events.

The capture unit sets the eQEP overflow error flag (QEPSTS[COEF]) in the event of capture timer overflow between unit position events. If a direction change occurs between the unit position events, then an error flag is set in the status register (QEPSTS[CDEF]).

Capture Timer (QCTMR) and Capture period register (QCPRD) can be configured to latch on following events.

- CPU read of QPOSCNT register
- Unit time-out event

If the QEPCTL[QCLM] bit is cleared, then the capture timer and capture period values are latched into the QCTMRLAT and QCPRDLAT registers, respectively, when the CPU reads the position counter (QPOSCNT).

If the QEPCTL[QCLM] bit is set, then the position counter, capture timer, and capture period values are latched into the QPOSLAT, QCTMRLAT and QCPRDLAT registers, respectively, on unit time out.

Figure 20-146 shows the capture unit operation along with the position counter.

NOTE: The QCAPCTL register should not be modified dynamically (such as switching CAPCLK prescaling mode from QCLK/4 to QCLK/8). The capture unit must be disabled before changing the prescaler.

Figure 20-144. eQEP Edge Capture Unit

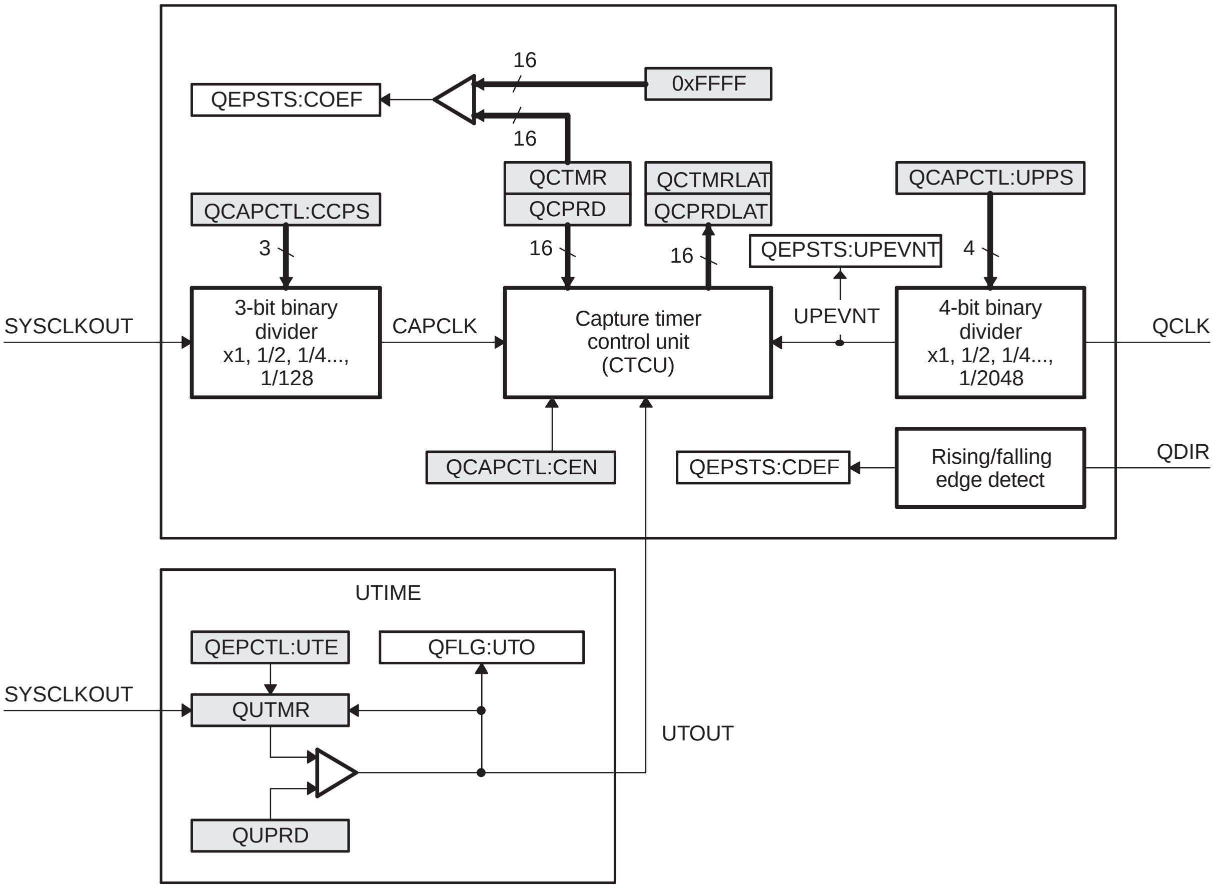
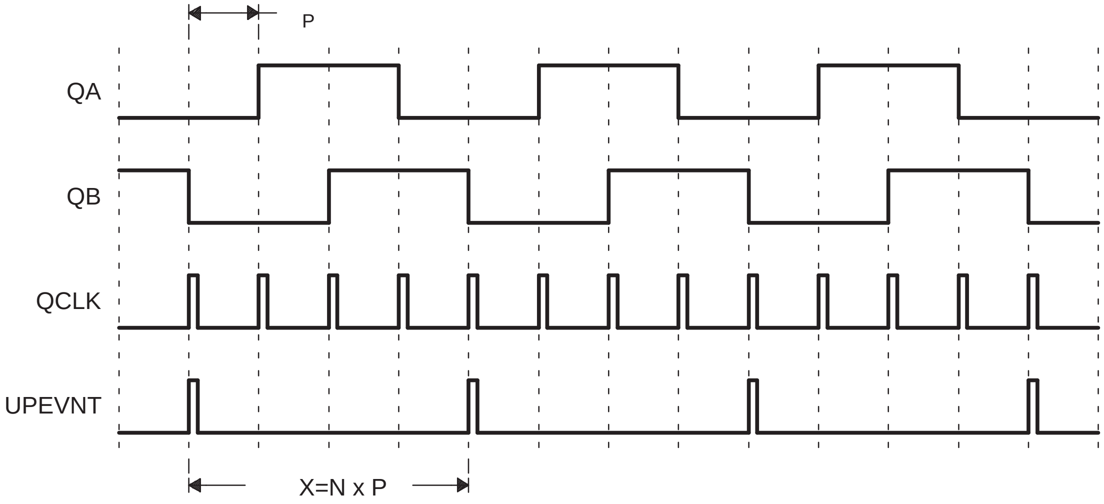
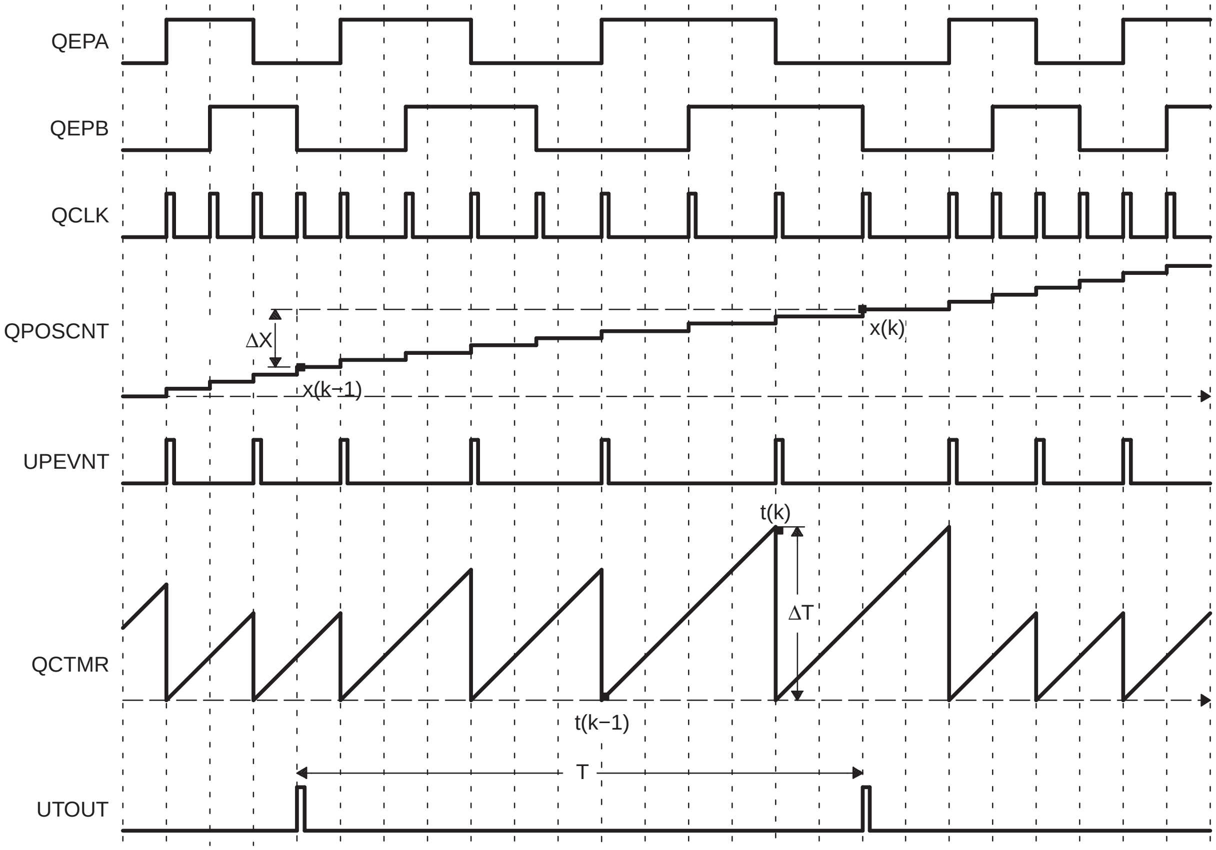


Figure 20-145. Unit Position Event for Low Speed Measurement (QCAPCTL[UPPS] = 0010)



N - Number of quadrature periods selected using QCAPCTL[UPPS] bits

Figure 20-146. eQEP Edge Capture Unit - Timing Details



Velocity Calculation Equations:

$$v(k) = \frac{x(k) - x(k-1)}{T} = \frac{\Delta X}{T} \text{ o} \qquad (25)$$

where

v(k): Velocity at time instant k

x(k): Position at time instant k

x(k-1): Position at time instant k - 1

T: Fixed unit time or inverse of velocity calculation rate

ΔX: Incremental position movement in unit time

X: Fixed unit position

ΔT: Incremental time elapsed for unit position movement

t(k): Time instant "k"

t(k-1): Time instant "k - 1"

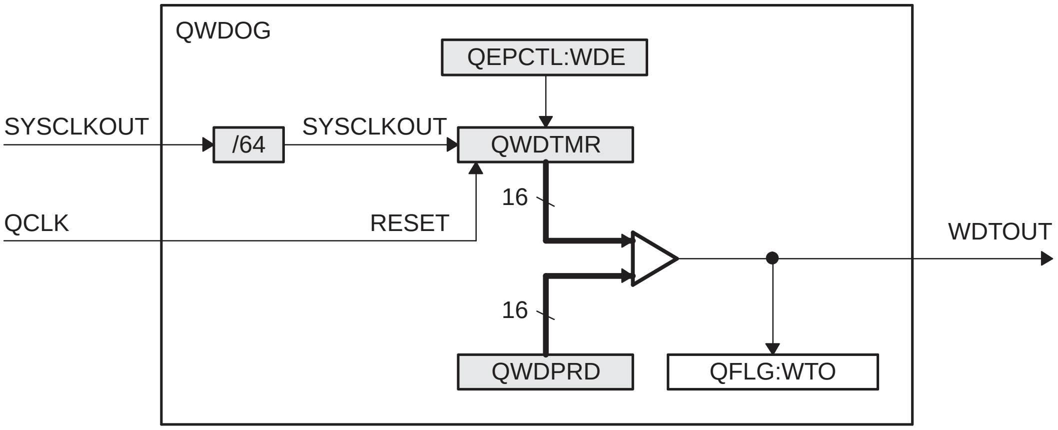
Unit time (T) and unit period (X) are configured using the QUPRD and QCAPCTL[UPPS] registers. Incremental position output and incremental time output is available in the QPOSLAT and QCPRDLAT registers.

Parameter	Relevant Register to Configure or Read the Information
T	Unit Period Register (QUPRD)
ΔX	Incremental Position = QPOSLAT(k) - QPOSLAT(K - 1)
X	Fixed unit position defined by sensor resolution and ZCAPCTL[UPPS] bits
ΔT	Capture Period Latch (QCPRDLAT)

20.4.2.6 eQEP Watchdog

The eQEP peripheral contains a 16-bit watchdog timer that monitors the quadrature-clock to indicate proper operation of the motion-control system. The eQEP watchdog timer is clocked from SYSCLKOUT/64 and the quadrate clock event (pulse) resets the watchdog timer. If no quadrature-clock event is detected until a period match (QWDPRD = QWDTMR), then the watchdog timer will time out and the watchdog interrupt flag will be set (QFLG[WTO]). The time-out value is programmable through the watchdog period register (QWDPRD).

Figure 20-147. eQEP Watchdog Timer

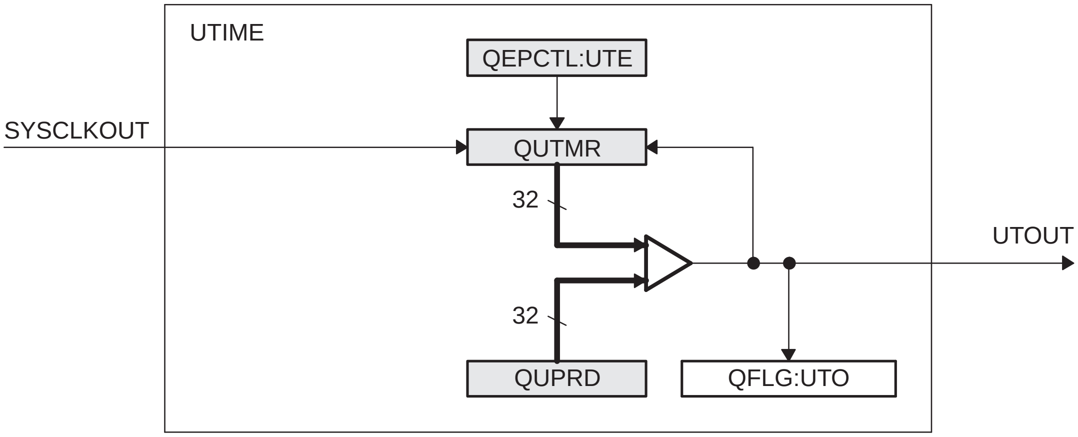


20.4.2.7 Unit Timer Base

The eQEP peripheral includes a 32-bit timer (QUTMR) that is clocked by SYSCLKOUT to generate periodic interrupts for velocity calculations. The unit time out interrupt is set (QFLG[UTO]) when the unit timer (QUTMR) matches the unit period register (QUPRD).

The eQEP peripheral can be configured to latch the position counter, capture timer, and capture period values on a unit time out event so that latched values are used for velocity calculation as described in Section Section 20.4.2.5.

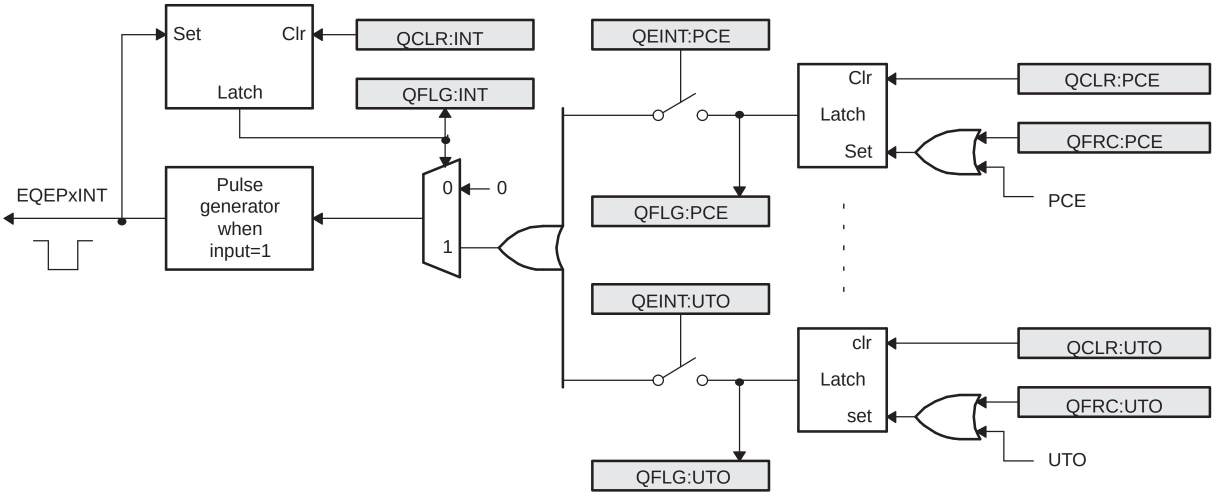
Figure 20-148. eQEP Unit Time Base



20.4.2.8 eQEP Interrupt Structure

Figure 20-149 shows how the interrupt mechanism works in the EQEP module.

Figure 20-149. EQEP Interrupt Generation



Eleven interrupt events (PCE, PHE, QDC, WTO, PCU, PCO, PCR, PCM, SEL, IEL, and UTO) can be generated. The interrupt control register (QEINT) is used to enable/disable individual interrupt event sources. The interrupt flag register (QFLG) indicates if any interrupt event has been latched and contains the global interrupt flag bit (INT). An interrupt pulse is generated only to the interrupt controller if any of the interrupt events is enabled, the flag bit is 1 and the INT flag bit is 0. The interrupt service routine will need to clear the global interrupt flag bit and the serviced event, via the interrupt clear register (QCLR), before any other interrupt pulses are generated. You can force an interrupt event by way of the interrupt force register (QFRC), which is useful for test purposes.

Note that the interrupts coming from the eQEP module are also used as DMA events. The interrupt registers should be used to enable and clear the current DMA event in order for the eQEP module to generate subsequent DMA events.

20.4.3 PWMSS_EQEP Registers

Table 20-116 lists the memory-mapped registers for the PWMSS_EQEP. All register offset addresses not listed in Table 20-116 should be considered as reserved locations and the register contents should not be modified.

Table 20-116. PWMSS_EQEP REGISTERS

Offset	Acronym	Register Name	Section
0h	QPOSCNT	eQEP Position Counter Register	Section 20.4.3.1
4h	QPOSINIT	eQEP Position Counter Initialization Register	Section 20.4.3.2
8h	QPOSMAX	eQEP Maximum Position Count Register	Section 20.4.3.3
Ch	QPOSCMP	eQEP Position-Compare Register	Section 20.4.3.4
10h	QPOSILAT	eQEP Index Position Latch Register	Section 20.4.3.5
14h	QPOSSLAT	eQEP Strobe Position Latch Register	Section 20.4.3.6
18h	QPOSLAT	eQEP Position Counter Latch Register	Section 20.4.3.7
1Ch	QUTMR	eQEP Unit Timer Register	Section 20.4.3.8
20h	QUPRD	eQEP Unit Period Register	Section 20.4.3.9
24h	QWDTMR	eQEP Watchdog Timer Register	Section 20.4.3.10
26h	QWDPRD	eQEP Watchdog Period Register	Section 20.4.3.11
28h	QDECCTL	eQEP Decoder Control Register	Section 20.4.3.12
2Ah	QEPCTL	eQEP Control Register	Section 20.4.3.13
2Ch	QCAPCTL	eQEP Capture Control Register	Section 20.4.3.14
2Eh	QPOSCTL	eQEP Position-Compare Control Register	Section 20.4.3.15
30h	QEINT	eQEP Interrupt Enable Register	Section 20.4.3.16
32h	QFLG	eQEP Interrupt Flag Register	Section 20.4.3.17
34h	QCLR	eQEP Interrupt Clear Register	Section 20.4.3.18
36h	QFRC	eQEP Interrupt Force Register	Section 20.4.3.19
38h	QEPSTS	eQEP Status Register	Section 20.4.3.20
3Ah	QCTMR	eQEP Capture Timer Register	Section 20.4.3.21
3Ch	QCPRD	eQEP Capture Period Register	Section 20.4.3.22
3Eh	QCTMRLAT	eQEP Capture Timer Latch Register	Section 20.4.3.23
40h	QCPRDLAT	eQEP Capture Period Latch Register	Section 20.4.3.24
5Ch	REVID	eQEP Revision ID Register	Section 20.4.3.25

20.4.3.1 QPOSCNT Register (offset = 0h) [reset = 0h]

QPOSCNT is shown in Figure 20-150 and described in Table 20-117.

Figure 20-150. QPOSCNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSCNT																															
R/W-0h																															

Table 20-117. QPOSCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSCNT	R/W	0h	This 32 bit position counter register counts up/down on every eQEP pulse based on direction input. This counter acts as a position integrator whose count value is proportional to position from a give reference point.

20.4.3.2 QPOSINIT Register (offset = 4h) [reset = 0h]

QPOSINIT is shown in Figure 20-151 and described in Table 20-118.

Figure 20-151. QPOSINIT Register

31-0
QPOSINIT
R/W-0h

Table 20-118. QPOSINIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSINIT	R/W	0h	This register contains the position value that is used to initialize the position counter based on external strobe or index event. The position counter can be initialized through software.

20.4.3.3 QPOSMAX Register (offset = 8h) [reset = 0h]

QPOSMAX is shown in Figure 20-152 and described in Table 20-119.

Figure 20-152. QPOSMAX Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">QPOSMAX</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 20-119. QPOSMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSMAX	R/W	0h	This register contains the maximum position counter value.

20.4.3.4 QPOSCMP Register (offset = Ch) [reset = 0h]

QPOSCMP is shown in Figure 20-153 and described in Table 20-120.

Figure 20-153. QPOSCMP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSCMP																															
R/W-0h																															

Table 20-120. QPOSCMP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSCMP	R/W	0h	The position-compare value in this register is compared with the position counter (QPOSCNT) to generate sync output and/or interrupt on compare match.

20.4.3.5 QPOSILAT Register (offset = 10h) [reset = 0h]

QPOSILAT is shown in Figure 20-154 and described in Table 20-121.

Figure 20-154. QPOSILAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSILAT																															
R-0h																															

Table 20-121. QPOSILAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSILAT	R	0h	The position-counter value is latched into this register on an index event as defined by the QEPCTL[IEL] bits.

20.4.3.6 QPOSSLAT Register (offset = 14h) [reset = 0h]

QPOSSLAT is shown in Figure 20-155 and described in Table 20-122.

Figure 20-155. QPOSSLAT Register

31–0
QPOSSLAT
R-0h

Table 20-122. QPOSSLAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSSLAT	R	0h	The position-counter value is latched into this register on strobe event as defined by the QEPCTL[SEL] bits.

20.4.3.7 QPOSLAT Register (offset = 18h) [reset = 0h]

QPOSLAT is shown in Figure 20-156 and described in Table 20-123.

Figure 20-156. QPOSLAT Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

31-0
QPOSLAT
R-0h

Table 20-123. QPOSLAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSLAT	R	0h	The position-counter value is latched into this register on unit time out event.

20.4.3.8 QUTMR Register (offset = 1Ch) [reset = 0h]

QUTMR is shown in Figure 20-157 and described in Table 20-124.

Figure 20-157. QUTMR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">QUTMR</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 20-124. QUTMR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QUTMR	R/W	0h	This register acts as time base for unit time event generation. When this timer value matches with unit time period value, unit time event is generated.

20.4.3.9 QUPRD Register (offset = 20h) [reset = 0h]

QUPRD is shown in Figure 20-158 and described in Table 20-125.

Figure 20-158. QUPRD Register

31–0
QUPRD
R/W-0h

Table 20-125. QUPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QUPRD	R/W	0h	This register contains the period count for unit timer to generate periodic unit time events to latch the eQEP position information at periodic interval and optionally to generate interrupt.

20.4.3.10 QWDTMR Register (offset = 24h) [reset = 0h]

QWDTMR is shown in Figure 20-159 and described in Table 20-126.

Figure 20-159. QWDTMR Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QWDTMR															
R/W-0h															

Table 20-126. QWDTMR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QWDTMR	R/W	0h	This register acts as time base for watch dog to detect motor stalls. When this timer value matches with watch dog period value, watch dog timeout interrupt is generated. This register is reset upon edge transition in quadrature-clock indicating the motion.

20.4.3.11 QWDPRD Register (offset = 26h) [reset = 0h]

QWDPRD is shown in Figure 20-160 and described in Table 20-127.

Figure 20-160. QWDPRD Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QWDPRD															
R/W-0h															

Table 20-127. QWDPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QWDPRD	R/W	0h	This register contains the time-out count for the eQEP peripheral watch dog timer. When the watchdog timer value matches the watchdog period value, a watchdog timeout interrupt is generated.

20.4.3.12 QDECCTL Register (offset = 28h) [reset = 0h]

QDECCTL is shown in Figure 20-161 and described in Table 20-128.

Figure 20-161. QDECCTL Register

15	14	13	12	11	10	9	8
QSRC		SOEN	SPSEL	XCR	SWAP	IGATE	QAP
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
QBP	QIP	QSP	RESERVED				
R/W-0h	R/W-0h	R/W-0h	R-0h				

Table 20-128. QDECCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	QSRC	R/W	0h	Position-counter source selection. 0h = Quadrature count mode (QCLK = iCLK, QDIR = iDIR) 1h = Direction-count mode (QCLK = xCLK, QDIR = xDIR) 2h = UP count mode for frequency measurement (QCLK = xCLK, QDIR = 1) 3h = DOWN count mode for frequency measurement (QCLK = xCLK, QDIR = 0)
13	SOEN	R/W	0h	Sync output-enable 0h = Disable position-compare sync output 1h = Enable position-compare sync output
12	SPSEL	R/W	0h	Sync output pin selection 0h = Index pin is used for sync output 1h = Strobe pin is used for sync output
11	XCR	R/W	0h	External clock rate 0h = 2x resolution: Count the rising/falling edge 1h = 1x resolution: Count the rising edge only
10	SWAP	R/W	0h	Swap quadrature clock inputs. This swaps the input to the quadrature decoder, reversing the counting direction. 0h = Quadrature-clock inputs are not swapped 1h = Quadrature-clock inputs are swapped
9	IGATE	R/W	0h	Index pulse gating option 0h = Disable gating of Index pulse 1h = Gate the index pin with strobe
8	QAP	R/W	0h	QEPA input polarity 0h = No effect 1h = Negates QEPA input
7	QBP	R/W	0h	QEPB input polarity 0h = No effect 1h = Negates QEPB input
6	QIP	R/W	0h	QEPI input polarity 0h = No effect 1h = Negates QEPI input
5	QSP	R/W	0h	QEPS input polarity 0h = No effect 1h = Negates QEPS input
4-0	RESERVED	R	0h	

20.4.3.13 QEPCTL Register (offset = 2Ah) [reset = 0h]

QEPCTL is shown in Figure 20-162 and described in Table 20-129.

Figure 20-162. QEPCTL Register

15	14	13	12	11	10	9	8
FREE_SOFT		PCRM		SEI		IEI	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

7	6	5	4	3	2	1	0
SWI	SEL	IEL		PHEN	QCLM	UTE	WDE
R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-129. QEPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	FREE_SOFT	R/W	0h	Emulation Control Bits. In the values 0 through 3 listed below, x is different for the four following behaviors. QPOSCNT behavior, x refers to the Position counter. QWDTMR behavior, x refers to the Watchdog counter. QUTMR behavior, x refers to the Unit timer. QCTMR behavior, x refers to the Capture timer. 0h = x stops immediately. For QPOSCNT behavior, the stop is on emulation suspend. 1h = x continues to count until the rollover. 2h = x is unaffected by emulation suspend. 3h = x is unaffected by emulation suspend.
13-12	PCRM	R/W	0h	Position counter reset mode 0h = Position counter reset on an index event 1h = Position counter reset on the maximum position 2h = Position counter reset on the first index event 3h = Position counter reset on a unit time event
11-10	SEI	R/W	0h	Strobe event initialization of position counter 0h = Does nothing (action disabled) 1h = Does nothing (action disabled) 2h = Initializes the position counter on rising edge of the QEPS signal 3h = Clockwise Direction: Initializes the position counter on the rising edge of QEPS strobe. Counter Clockwise Direction: Initializes the position counter on the falling edge of QEPS strobe
9-8	IEI	R/W	0h	Index event initialization of position counter 0h = Do nothing (action disabled) 1h = Do nothing (action disabled) 2h = Initializes the position counter on the rising edge of the QEPI signal (QPOSCNT = QPOSINIT) 3h = Initializes the position counter on the falling edge of QEPI signal (QPOSCNT = QPOSINIT)
7	SWI	R/W	0h	Software initialization of position counter 0h = Do nothing (action disabled) 1h = Initialize position counter, this bit is cleared automatically
6	SEL	R/W	0h	Strobe event latch of position counter 0h = The position counter is latched on the rising edge of QEPS strobe (QPOSSLAT = POSCCNT). Latching on the falling edge can be done by inverting the strobe input using the QSP bit in the QDECCTL register. 1h = Clockwise Direction: Position counter is latched on rising edge of QEPS strobe. Counter Clockwise Direction: Position counter is latched on falling edge of QEPS strobe.

Table 20-129. QEPCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	IEL	R/W	0h	Index event latch of position counter (software index marker) 0h = Reserved 1h = Latches position counter on rising edge of the index signal 2h = Latches position counter on falling edge of the index signal 3h = Software index marker. Latches the position counter and quadrature direction flag on index event marker. The position counter is latched to the QPOSILAT register and the direction flag is latched in the QEPSTS[QDLF] bit. This mode is useful for software index marking.
3	PHEN	R/W	0h	Quadrature position counter enable/software reset 0h = Reset the eQEP peripheral internal operating flags/read-only registers. Control/configuration registers are not disturbed by a software reset. 1h = eQEP position counter is enabled
2	QCLM	R/W	0h	eQEP capture latch mode 0h = Latch on position counter read by CPU. Capture timer and capture period values are latched into QCTMRLAT and QCPRDLAT registers when CPU reads the QPOSCNT register. 1h = Latch on unit time out. Position counter, capture timer and capture period values are latched into QPOSLAT, QCTMRLAT and QCPRDLAT registers on unit time out.
1	UTE	R/W	0h	eQEP unit timer enable 0h = Disable eQEP unit timer 1h = Enable unit timer
0	WDE	R/W	0h	eQEP watchdog enable 0h = Disable the eQEP watchdog timer 1h = Enable the eQEP watchdog timer

20.4.3.14 QCAPCTL Register (offset = 2Ch) [reset = 0h]

QCAPCTL is shown in Figure 20-163 and described in Table 20-130.

Figure 20-163. QCAPCTL Register

15	14	13	12	11	10	9	8
CEN	RESERVED						
R/W-0h	R-0h						

7	6	5	4	3	2	1	0
RESERVED	CCPS			UPPS			
R-0h	R/W-0h			R/W-0h			

Table 20-130. QCAPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	CEN	R/W	0h	Enable eQEP capture 0h = eQEP capture unit is disabled 1h = eQEP capture unit is enabled
14-7	RESERVED	R	0h	
6-4	CCPS	R/W	0h	eQEP capture timer clock prescaler 0h = CAPCLK = SYSCLKOUT/1 1h = CAPCLK = SYSCLKOUT/2 2h = CAPCLK = SYSCLKOUT/4 3h = CAPCLK = SYSCLKOUT/8 4h = CAPCLK = SYSCLKOUT/16 5h = CAPCLK = SYSCLKOUT/32 6h = CAPCLK = SYSCLKOUT/64 7h = CAPCLK = SYSCLKOUT/128
3-0	UPPS	R/W	0h	Unit position event prescaler 0h = UPEVNT = QCLK/1 1h = UPEVNT = QCLK/2 2h = UPEVNT = QCLK/4 3h = UPEVNT = QCLK/8 4h = UPEVNT = QCLK/16 5h = UPEVNT = QCLK/32 6h = UPEVNT = QCLK/64 7h = UPEVNT = QCLK/128 8h = UPEVNT = QCLK/256 9h = UPEVNT = QCLK/512 Ah = UPEVNT = QCLK/1024 Bh = UPEVNT = QCLK/2048 Ch = Reserved Dh = Reserved Eh = Reserved Fh = Reserved

20.4.3.15 QPOSCTL Register (offset = 2Eh) [reset = 0h]

QPOSCTL is shown in Figure 20-164 and described in Table 20-131.

Figure 20-164. QPOSCTL Register

15	14	13	12	11	10	9	8
PCSHDW	PCLOAD	PCPOL	PCE	PCSPW			
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h			

7	6	5	4	3	2	1	0
PCSPW							
R/W-0h							

Table 20-131. QPOSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	PCSHDW	R/W	0h	Position-compare shadow enable 0h = Shadow disabled, load Immediate 1h = Shadow enabled
14	PCLOAD	R/W	0h	Position-compare shadow load mode 0h = Load on QPOSCNT = 0 1h = Load when QPOSCNT = QPOSCMP
13	PCPOL	R/W	0h	Polarity of sync output 0h = Active HIGH pulse output 1h = Active LOW pulse output
12	PCE	R/W	0h	Position-compare enable/disable 0h = Disable position compare unit 1h = Enable position compare unit
11-0	PCSPW	R/W	0h	Select-position-compare sync output pulse width ... 0h = 1 x 4 x SYSCLKOUT cycles 1h = 2 x 4 x SYSCLKOUT cycles 2h = 3 x 4 x SYSCLKOUT cycles to 4096 x 4 x SYSCLKOUT cycles FFFh = 3 x 4 x SYSCLKOUT cycles to 4096 x 4 x SYSCLKOUT cycles

20.4.3.16 QEINT Register (offset = 30h) [reset = 0h]

QEINT is shown in Figure 20-165 and described in Table 20-132.

Figure 20-165. QEINT Register

15	14	13	12	11	10	9	8
RESERVED				UTO	IEL	SEL	PCM
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	PHE	PCE	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 20-132. QEINT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	
11	UTO	R/W	0h	Unit time out interrupt enable 0h = Interrupt is disabled 1h = Interrupt is enabled
10	IEL	R/W	0h	Index event latch interrupt enable 0h = Interrupt is disabled 1h = Interrupt is enabled
9	SEL	R/W	0h	Strobe event latch interrupt enable 0h = Interrupt is disabled 1h = Interrupt is enabled
8	PCM	R/W	0h	Position-compare match interrupt enable 0h = Interrupt is disabled 1h = Interrupt is enabled
7	PCR	R/W	0h	Position-compare ready interrupt enable 0h = Interrupt is disabled 1h = Interrupt is enabled
6	PCO	R/W	0h	Position counter overflow interrupt enable 0h = Interrupt is disabled 1h = Interrupt is enabled
5	PCU	R/W	0h	Position counter underflow interrupt enable 0h = Interrupt is disabled 1h = Interrupt is enabled
4	WTO	R/W	0h	Watchdog time out interrupt enable 0h = Interrupt is disabled 1h = Interrupt is enabled
3	QDC	R/W	0h	Quadrature direction change interrupt enable 0h = Interrupt is disabled 1h = Interrupt is enabled
2	PHE	R/W	0h	Quadrature phase error interrupt enable 0h = Interrupt is disabled 1h = Interrupt is enabled
1	PCE	R/W	0h	Position counter error interrupt enable 0h = Interrupt is disabled 1h = Interrupt is enabled
0	RESERVED	R	0h	

20.4.3.17 QFLG Register (offset = 32h) [reset = 0h]

QFLG is shown in Figure 20-166 and described in Table 20-133.

Figure 20-166. QFLG Register

15	14	13	12	11	10	9	8
RESERVED				UTO	IEL	SEL	PCM
R-0h				R-0h	R-0h	R-0h	R-0h

7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	PHE	PCE	INT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-133. QFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	
11	UTO	R	0h	Unit time out interrupt flag 0h = No interrupt generated 1h = Set by eQEP unit timer period match
10	IEL	R	0h	Index event latch interrupt flag 0h = No interrupt generated 1h = This bit is set after latching the QPOSCNT to QPOSILAT
9	SEL	R	0h	Strobe event latch interrupt flag 0h = No interrupt generated 1h = This bit is set after latching the QPOSCNT to QPOSSLAT
8	PCM	R	0h	eQEP compare match event interrupt flag 0h = No interrupt generated 1h = This bit is set on position-compare match
7	PCR	R	0h	Position-compare ready interrupt flag 0h = No interrupt generated 1h = This bit is set after transferring the shadow register value to the active position compare register.
6	PCO	R	0h	Position counter overflow interrupt flag 0h = No interrupt generated 1h = This bit is set on position counter overflow.
5	PCU	R	0h	Position counter underflow interrupt flag 0h = No interrupt generated 1h = This bit is set on position counter underflow.
4	WTO	R	0h	Watchdog timeout interrupt flag 0h = No interrupt generated 1h = Set by watch dog timeout
3	QDC	R	0h	Quadrature direction change interrupt flag 0h = No interrupt generated 1h = This bit is set during change of direction
2	PHE	R	0h	Quadrature phase error interrupt flag 0h = No interrupt generated 1h = Set on simultaneous transition of QEPA and QEPB
1	PCE	R	0h	Position counter error interrupt flag 0h = No interrupt generated 1h = Position counter error
0	INT	R	0h	Global interrupt status flag 0h = No interrupt generated 1h = Interrupt was generated

20.4.3.18 QCLR Register (offset = 34h) [reset = 0h]

QCLR is shown in Figure 20-167 and described in Table 20-134.

Figure 20-167. QCLR Register

15	14	13	12	11	10	9	8
RESERVED				UTO	IEL	SEL	PCM
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	PHE	PCE	INT
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-134. QCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	
11	UTO	R/W	0h	Clear unit time out interrupt flag 0h = No effect 1h = Clears the interrupt flag
10	IEL	R/W	0h	Clear index event latch interrupt flag 0h = No effect 1h = Clears the interrupt flag
9	SEL	R/W	0h	Clear strobe event latch interrupt flag 0h = No effect 1h = Clears the interrupt flag
8	PCM	R/W	0h	Clear eQEP compare match event interrupt flag 0h = No effect 1h = Clears the interrupt flag
7	PCR	R/W	0h	Clear position-compare ready interrupt flag 0h = No effect 1h = Clears the interrupt flag
6	PCO	R/W	0h	Clear position counter overflow interrupt flag 0h = No effect 1h = Clears the interrupt flag
5	PCU	R/W	0h	Clear position counter underflow interrupt flag 0h = No effect 1h = Clears the interrupt flag
4	WTO	R/W	0h	Clear watchdog timeout interrupt flag 0h = No effect 1h = Clears the interrupt flag
3	QDC	R/W	0h	Clear quadrature direction change interrupt flag 0h = No effect 1h = Clears the interrupt flag
2	PHE	R/W	0h	Clear quadrature phase error interrupt flag 0h = No effect 1h = Clears the interrupt flag
1	PCE	R/W	0h	Clear position counter error interrupt flag 0h = No effect 1h = Clears the interrupt flag
0	INT	R/W	0h	Global interrupt clear flag 0h = No effect 1h = Clears the interrupt flag and enables further interrupts to be generated if an event flags is set to 1.

20.4.3.19 QFRC Register (offset = 36h) [reset = 0h]

QFRC is shown in Figure 20-168 and described in Table 20-135.

Figure 20-168. QFRC Register

15	14	13	12	11	10	9	8
RESERVED				UTO	IEL	SEL	PCM
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	PHE	PCE	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 20-135. QFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	
11	UTO	R/W	0h	Force unit time out interrupt 0h = No effect 1h = Force the interrupt
10	IEL	R/W	0h	Force index event latch interrupt 0h = No effect 1h = Force the interrupt
9	SEL	R/W	0h	Force strobe event latch interrupt 0h = No effect 1h = Force the interrupt
8	PCM	R/W	0h	Force position-compare match interrupt 0h = No effect 1h = Force the interrupt
7	PCR	R/W	0h	Force position-compare ready interrupt 0h = No effect 1h = Force the interrupt
6	PCO	R/W	0h	Force position counter overflow interrupt 0h = No effect 1h = Force the interrupt
5	PCU	R/W	0h	Force position counter underflow interrupt 0h = No effect 1h = Force the interrupt
4	WTO	R/W	0h	Force watchdog time out interrupt 0h = No effect 1h = Force the interrupt
3	QDC	R/W	0h	Force quadrature direction change interrupt 0h = No effect 1h = Force the interrupt
2	PHE	R/W	0h	Force quadrature phase error interrupt 0h = No effect 1h = Force the interrupt
1	PCE	R/W	0h	Force position counter error interrupt 0h = No effect 1h = Force the interrupt
0	RESERVED	R	0h	

20.4.3.20 QEPSTS Register (offset = 38h) [reset = 0h]

QEPSTS is shown in Figure 20-169 and described in Table 20-136.

Figure 20-169. QEPSTS Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
UPEVNT	FDF	QDF	QDLF	COEF	CDEF	FIMF	PCEF
R-0h	R-0h	R-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 20-136. QEPSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7	UPEVNT	R	0h	Unit position event flag 0h = No unit position event detected 1h = Unit position event detected. Write 1 to clear.
6	FDF	R	0h	Direction on the first index marker. Status of the direction is latched on the first index event marker. 0h = Counter-clockwise rotation (or reverse movement) on the first index event 1h = Clockwise rotation (or forward movement) on the first index event
5	QDF	R	0h	Quadrature direction flag 0h = Counter-clockwise rotation (or reverse movement) 1h = Clockwise rotation (or forward movement)
4	QDLF	R	0h	eQEP direction latch flag. Status of direction is latched on every index event marker. 0h = Counter-clockwise rotation (or reverse movement) on index event marker 1h = Clockwise rotation (or forward movement) on index event marker
3	COEF	R/W	0h	Capture overflow error flag 0h = Sticky bit, cleared by writing 1 1h = Overflow occurred in eQEP Capture timer (QEPCTMR)
2	CDEF	R/W	0h	Capture direction error flag 0h = Sticky bit, cleared by writing 1 1h = Direction change occurred between the capture position event.
1	FIMF	R/W	0h	First index marker flag 0h = Sticky bit, cleared by writing 1 1h = Set by first occurrence of index pulse
0	PCEF	R	0h	Position counter error flag. This bit is not sticky and it is updated for every index event. 0h = No error occurred during the last index transition. 1h = Position counter error

20.4.3.21 QCTMR Register (offset = 3Ah) [reset = 0h]

QCTMR is shown in Figure 20-170 and described in Table 20-137.

Figure 20-170. QCTMR Register

15-0
QCTMR
R/W-0h

Table 20-137. QCTMR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QCTMR	R/W	0h	This register provides time base for edge capture unit.

20.4.3.22 QCPRD Register (offset = 3Ch) [reset = 0h]

QCPRD is shown in Figure 20-171 and described in Table 20-138.

Figure 20-171. QCPRD Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QCPRD															
R/W-0h															

Table 20-138. QCPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QCPRD	R/W	0h	This register holds the period count value between the last successive eQEP position events

20.4.3.23 QCTMRLAT Register (offset = 3Eh) [reset = 0h]

QCTMRLAT is shown in Figure 20-172 and described in Table 20-139.

Figure 20-172. QCTMRLAT Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QCTMRLAT															
R-0h															

Table 20-139. QCTMRLAT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QCTMRLAT	R	0h	The eQEP capture timer value can be latched into this register on two events, that is, unit timeout event, reading the eQEP position counter.

20.4.3.24 QCPRDLAT Register (offset = 40h) [reset = 0h]

QCPRDLAT is shown in Figure 20-173 and described in Table 20-140.

Figure 20-173. QCPRDLAT Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QCPRDLAT															
R/W-0h															

Table 20-140. QCPRDLAT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QCPRDLAT	R/W	0h	eQEP capture period value can be latched into this register on two events, that is, unit timeout event, reading the eQEP position counter.

20.4.3.25 REVID Register (offset = 5Ch) [reset = 44D31103h]

REVID is shown in Figure 20-174 and described in Table 20-141.

Figure 20-174. REVID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REV																															
R-44D31103h																															

Table 20-141. REVID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REV	R	44D31103h	eQEP revision ID

Universal Asynchronous Receiver/Transmitter (UART)

This chapter describes the UART of the device.

Topic **Page**

21.1 Introduction

21.1.1 UART Mode Features

The general features of the UART/IrDA module when operating in UART mode are:

- 16C750 compatibility
- Baud rate from 300 bps up to 3.6864 Mbps
- Auto-baud between 1200 bps and 115.2 Kbps
- Software/Hardware flow control
  - Programmable Xon/Xoff characters
  - Programmable Auto-RTS and Auto CTS
- Programmable serial interface characteristics
  - 5, 6, 7, or 8-bit characters
  - Even, odd, mark (always 1), space (always 0), or no parity (non-parity bit frame) bit generation and detection
  - 1, 1.5, or 2 stop bit generation
- False start bit detection
- Line break generation and detection
- Modem control functions (CTS, RTS, DSR, DTR, RI, and DCD)
- Fully prioritized interrupt system controls
- Internal test and loopback capabilities

21.1.2 IrDA Mode Features

The general features of the UART/IrDA when operating in IrDA mode are:

- Support of IrDA 1.4 slow infrared (SIR), medium infrared (MIR) and fast infrared (FIR) communications (very fast infrared (VFIR) is not supported)
- Frame formatting: addition of variable xBOF characters and EOF characters
- Uplink/downlink CRC generation/detection
- Asynchronous transparency (automatic insertion of break character)
- 8-entry status FIFO (with selectable trigger levels) available to monitor frame length and frame errors
- Framing error, cyclic redundancy check (CRC) error, illegal symbol (FIR), abort pattern (SIR, MIR) detection

21.1.3 CIR Mode Features

The general features of the UART/IrDA when operating in CIR mode are:

- Support of consumer infrared (CIR) for remote control applications
- Transmit and receive
- Free data format (supports any remote control private standards)
- Selectable bit rate
- Configurable carrier frequency
- 1/2, 5/12, 1/3 or 1/4 carrier duty cycle

21.1.4 Unsupported UART Features

The following UART/IrDA module features are not supported in this device.

Table 21-1. Unsupported UART Features

Feature	Reason
Full modem control on UART0	DCD, DSR, DTR, RI not pinned-out
Full modem control on UART2-5	DCD, DSR, DTR, RI not pinned-out
Device wake-up on UART1-5	Wake-up not supported - no SWake connection

21.2 Integration

This device contains 6 instantiations of the UART/IrDA (UARTIRDAOCP) peripheral. There are six UART modules called UART0 – UART5. UART0 provides wakeup capability. Only UART 1 provides full modem control signals. All UARTs support IrDA and CIR modes and RTS/CTS flow control (subject to pin muxing configuration). Figure 21-1 shows an example of system connectivity using UART communication with hardware handshake.

Figure 21-1. UART/IrDA Module — UART Application

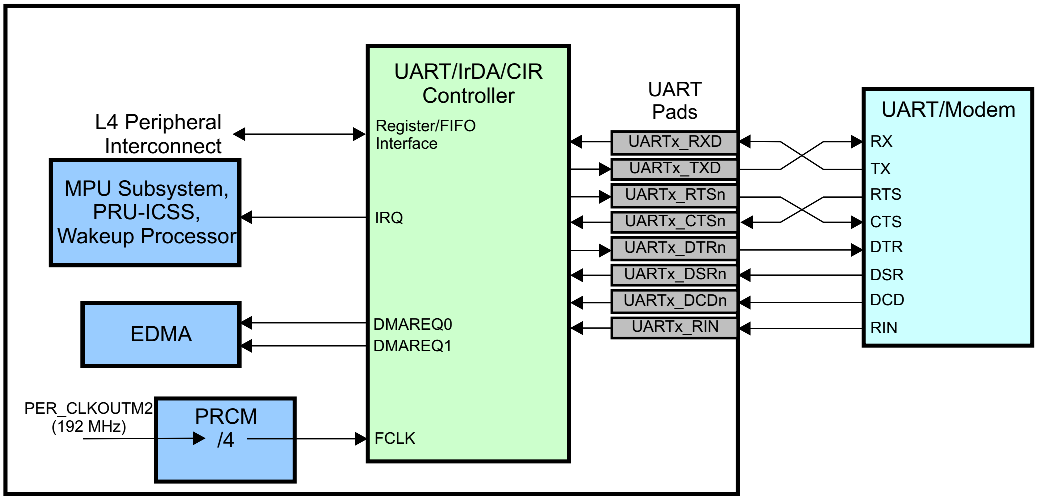
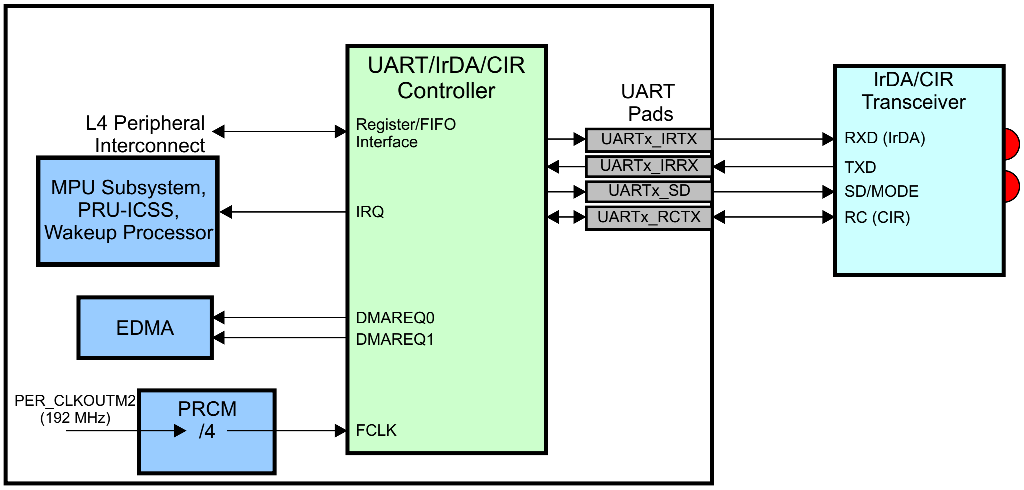


Figure 21-2 shows an example of system connectivity using infrared communication with remote control (consumer infrared).

Figure 21-2. UART/IrDA Module — IrDA/CIR Application



21.2.1 UART Connectivity Attributes

The general connectivity attributes for each of the UART modules are shown in Table 21-2 and Table 21-3.

Table 21-2. UART0 Connectivity Attributes

Attributes	Type
Power Domain	Wake-Up Domain
Clock Domain	PD_WKUP_L4_WKUP_GCLK (OCP) PD_WKUP_UART0_GFCLK (Func)

Table 21-2. UART0 Connectivity Attributes (continued)

Attributes	Type
Reset Signals	WKUP_DOM_RST_N
Idle/Wakeup Signals	Smart Idle / Wakeup
Interrupt Requests	1 interrupt to MPU Subsystem (UART0INT), PRU-ICSS (nirq) and Wakeup Processor SWakeup to Wakeup Processor
DMA Requests	2 DMA requests to EDMA (TX – UTXEVT0, RX – URXEVT0)
Physical Address	L4 Wakeup slave port

Table 21-3. UART1–5 Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L4LS_GCLK (OCP) PD_PER_UART_GFCLK (Func)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	UART1-2 1 interrupt per instance to MPU Subsystem (UART1INT, UART2INT) and PRU-ICSS (nirq) UART3-5 1 interrupt per instance to only MPU Subsystem (UART3INT, UART4INT, UART5INT)
DMA Requests	2 DMA requests per instance to EDMA (TX – UTXEVTx, RX – URXEVTx)
Physical Address	L4 Peripheral slave port

21.2.2 UART Clock and Reset Management

The UART modules use separate functional and bus interface clocks.

Table 21-4. UART0 Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
CLK Interface clock From PRCM	26 MHz	M_OSC_CLK	pd_wkup_l4_wkup_gclk
FCLK Functional clock From PRCM	48 MHz	PER_CLKOUTM2 / 4	pd_wkup_uart0_gfclk

Table 21-5. UART1–5 Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
CLK Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk From PRCM
FCLK Functional clock	48 MHz	PER_CLKOUTM2 / 4	pd_per_uart_gfclk From PRCM

For UART operation, the functional clock is used to produce a baud rate up to 3.6M bits/s. Table 21-6 lists the supported baud rates, the requested divider, and the corresponding error versus the standard baud rate.

Table 21-6. UART Mode Baud and Error Rates

Baud rate	Over sampling	Divisor	Error (%)
300	16	10000	0
600	16	5000	0
1200	16	2500	0
2400	16	1250	0
4800	16	625	0
9600	16	313	0.16
14400	16	208	0.16
19200	16	156	0.16
28800	16	104	0.16
38400	16	78	0.16
57600	16	52	0.16
115200	16	26	0.16
230400	16	13	0.16
460800	13	8	0.16
921600	13	4	0.16
1843200	13	2	0.16
3000000	16	1	0
3686400	13	1	0.16

For IrDA operation, the internal functional clock divisor allows generation of SIR, MIR, or FIR baud rates as shown in Table 21-7.

Table 21-7. IrDA Mode Baud and Error Rates

Baud rate	IR mode	Encoding	Divisor	Error (%)
2400	SIR	3/16	1250	0
9600	SIR	3/16	312	0.16
19200	SIR	3/16	156	0.16
38400	SIR	3/16	78	0.16
57600	SIR	3/16	52	0.16
115200	SIR	3/16	26	0.16
576000	MIR	1/4	2	0
1152000	MIR	1/4	1	0
4000000	FIR	4PPM	1	0

21.2.3 UART Pin List

The UART interface pins are listed in Table 21-8. Pin functionality depends on the selected operating mode of the module.

Table 21-8. UART Pin List

Pin	Type	Description
UARTx_RXD / IRRX / RCRX	I	UART / IrDA / CIR Receive Data
UARTx_TXD / IRTX / RCTX	O/Z	UART / IrDA / CIR Transmit Data
UARTx_RTSn / SD	O/Z	UART Request to Send / IrDA Mode
UARTx_CTSn	I	UART Clear to Send
UARTx_DTRn(1)	O/Z	UART Data Terminal Ready
UARTx_DSRn(1)	I	UART Data Set Ready
UARTx_DCDn(1)	I	UART Data Carrier Detect
UARTx_RIn(1)	I	UART Ring Indicator

(1) UART1 only

The UART module can operate in three different modes based on the MODE_SELECT bits. The signal muxing based on these mode bits is shown in Table 21-9.

Table 21-9. UART Muxing Control

UARTx_TXD / IRTX / RCTX Function	UARTx_RXD / IRRX / RCRX Function	UARTx_RTSn / SD Function	UARTx_CTSn Function	Mode
TXD	RXD	RTSn	CTSn	UART
IRTX	IRRX	SD	not used	IrDA (SIR, MIR, FIR)
RCTX	RCRX	SD	not used	CIR

21.3 Functional Description

21.3.1 Block Diagram

The UART/IrDA/CIR module can be divided into three main blocks:

- FIFO management
- Mode selection
- Protocol formatting

FIFO management is common to all functions and enables the transmission and reception of data from the host processor point of view.

There are two modes:

- Function mode: Routes the data to the chosen function (UART, IrDA, or CIR) and enables the mechanism corresponding to the chosen function
- Register mode: Enables conditional access to registers

For more information about mode configuration, see Section 21.3.7, *Mode Selection*.

Protocol formatting has three subcategories:

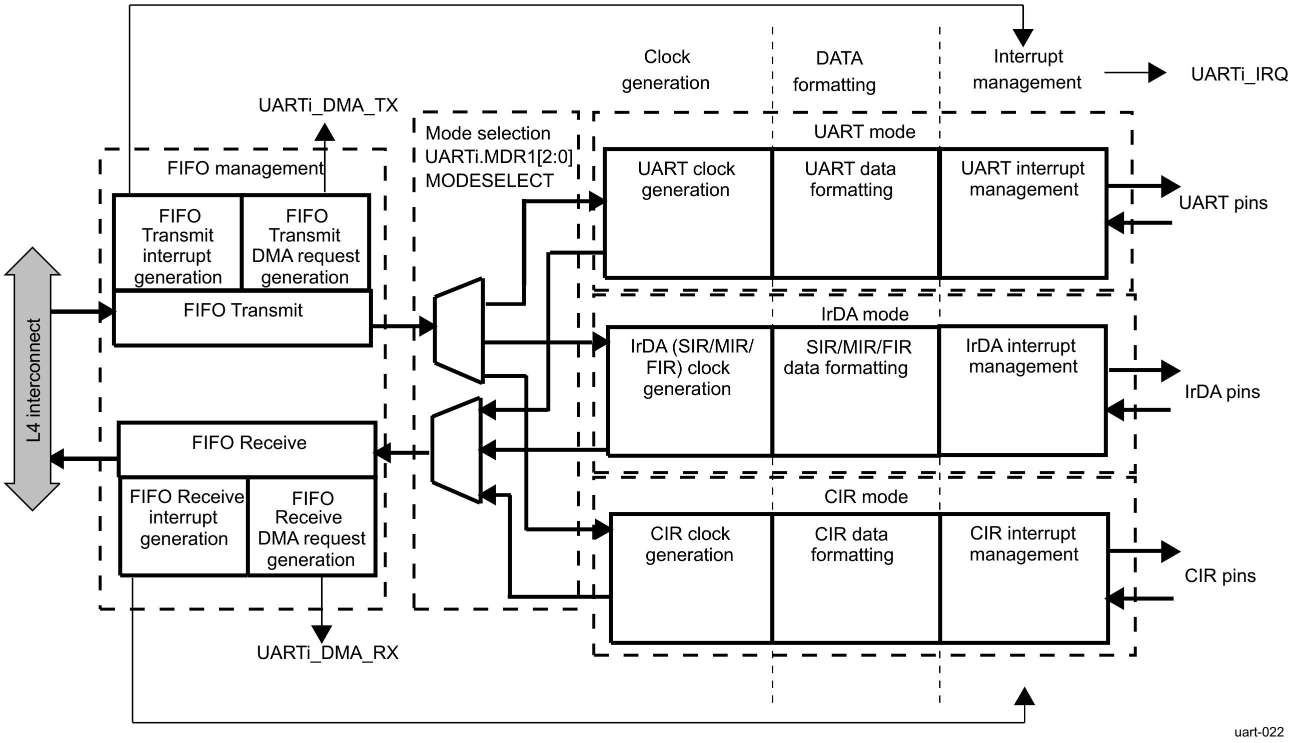
- Clock generation: The 48-MHz input clock generates all necessary clocks.
- Data formatting: Each function uses its own state-machine that is responsible for the transition between FIFO data and frame data associated with it.
- Interrupt management: Different interrupt types are generated depending on the chosen function:
  - UART mode interrupts: Seven interrupts prioritized in six different levels
  - IrDA mode interrupts: Eight interrupts. The interrupt line is activated when any interrupt is generated (there is no priority).
  - CIR mode interrupts: A subset of existing IrDA mode interrupts is used.

  In each mode, when an interrupt is generated, the UART_IIR register indicates the interrupt type.

In parallel with these functional blocks, a power-saving strategy exists for each function.

Figure 21-3 is the UART/IrDA/CIR block diagram.

Figure 21-3. UART/IrDA/CIR Functional Specification Block Diagram



21.3.2 Clock Configuration

Each UART uses a 48-MHz functional clock for its logic and to generate external interface signals. Each UART uses an interface clock for register accesses. The PRCM module generates and controls all these clocks (for more information, see *Clock Domain Module Attributes*, in Chapter 6, *Power, Reset, and Clock Management*).

The idle and wake-up processes use a handshake protocol between the PRCM and the UART (for a description of the protocol, see *Module-Level Clock Management* in Chapter 6, *Power, Reset, and Clock Management*). The UARTi.UART_SYSC[4:3] IDLEMODE bit field controls UART idle mode.

21.3.3 Software Reset

The UARTi.UART_SYSC[1] SOFTRESET bit controls the software reset; setting this bit to 1 triggers a software reset functionally equivalent to hardware reset.

21.3.4 Power Management

21.3.4.1 UART Mode Power Management

21.3.4.1.1 Module Power Saving

In UART modes, sleep mode is enabled by setting the UARTi.UART_IER[4] SLEEP_MODE bit to 1 (when the UARTi.UART_EFR[4] ENHANCED_EN bit is set to 1).

Sleep mode is entered when all the following conditions exist:

- The serial data input line, uarti_rx, is idle.
- The TX FIFO and TX shift register are empty.
- The RX FIFO is empty.
- The only pending interrupts are THR interrupts.

Sleep mode is a good way to lower UART power consumption, but this state can be achieved only when the UART is set to modem mode. Therefore, even if the UART has no key role functionally, it must be initialized in a functional mode to take advantage of sleep mode.

In sleep mode, the module clock and baud rate clock are stopped internally. Because most registers are clocked by these clocks, this greatly reduces power consumption. The module wakes up when a change is detected on the uarti_rx line, when data is written to the TX FIFO, and when there is a change in the state of the modem input pins.

An interrupt can be generated on a wake-up event by setting the UARTi.UART_SCR[4] RX_CTS_WU_EN bit to 1. To understand how to manage the interrupt, see Section 21.3.5.2, *Wake-Up Interrupt*.

NOTE: There must be no writing to the divisor latches, UARTi.UART_DLL and UARTi.UART_DLH, to set the baud clock (BCLK) while in sleep mode. It is advisable to disable sleep mode using the UARTi.UART_IER[4] SLEEP_MODE bit before writing to the UARTi.UART_DLL register or the UARTi.UART_DLH register.

21.3.4.1.2 System Power Saving

Sleep and auto-idle modes are embedded power-saving features. Power-reduction techniques can be applied at the system level by shutting down certain internal clock and power domains of the device.

The UART supports an idle req/idle ack handshaking protocol used at the system level to shut down the UART clocks in a clean and controlled manner and to switch the UART from interrupt-generation mode to wake-up generation mode for unmasked events (see the UARTi.UART_SYSC[2] ENAWAKEUP bit and the UARTi.UART_WER register).

For more information, see *Module Level Clock Management* in Chapter 6, *Power, Reset, and Clock Management*.

21.3.4.2 IrDA/CIR Mode Power Management

21.3.4.2.1 Module Power Saving

In IrDA/CIR modes, sleep mode is enabled by setting the UARTi.MDR[3] IR_SLEEP bit to 1.

Sleep mode is entered when all the following conditions exist:

- The serial data input line, uarti.rx_irrx, is idle.
- The TX FIFO and TX shift register are empty.
- The RX FIFO is empty.
- No interrupts are pending except THR interrupts.

The module wakes up when a change is detected on the uarti_rx_irrx line or when data is written to the TX FIFO.

21.3.4.2.2 System Power Saving

System power saving for the IrDA/CIR mode has the same function as for the UART mode (see Section 21.3.4.1.2, *System Power Saving*).

21.3.4.3 Local Power Management

Table 21-10 describes power-management features available for the UART.

NOTE: For information about source clock gating and sleep/wake-up transitions description, see *Module-Level Clock Management* in Chapter 6, *Power, Reset, and Clock Management*.

Table 21-10. Local Power-Management Features

Feature	Registers	Description
Clock autogating	UART_SYSC[0] AUTOIDLE	This bit allows local power optimization in the module by gating the UARTi_ICLK clock on interface activity or gating the UARTi_FCLK clock on internal activity.
Slave idle modes	UART_SYSC[4:3] IDLEMODE	Force-idle, no-idle, smart-idle, and smart-idle wakeup-capable modes are available
Clock activity	N/A	Feature not available
Master standby modes	N/A	Feature not available
Global wake-up enable	UART_SYSC[2] ENAWAKEUP	This bit enables the wake-up feature at module level.
Wake-Up sources enable	N/A	Feature not available

21.3.5 Interrupt Requests

The UART IrDA CIR module generates interrupts. All interrupts can be enabled/disabled by writing to the appropriate bit in the interrupt enable register (IER). The interrupt status of the device can be checked at any time by reading the interrupt identification register (IIR). The UART, IrDA, and CIR modes have different interrupts in the UART IrDA CIR module and therefore have different IER and IIR mappings according to the selected mode.

21.3.5.1 UART Mode Interrupt Management

21.3.5.1.1 UART Interrupts

UART mode includes seven possible interrupts prioritized to six levels.

When an interrupt is generated, the interrupt identification register (UARTi.UART_IIR) sets the UARTi.UART_IIR[0] IT_PENDING bit to 0 to indicate that an interrupt is pending, and indicates the type of interrupt through the UARTi.UART_IIR[5:1] bit field. Table 21-11 summarizes the interrupt control functions.

Table 21-11. UART Mode Interrupts

UART_IIR[5:0]	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Method
000001	None	None	None	None
000110	1	Receiver line status	OE, FE, PE, or BI errors occur in characters in the RX FIFO.	FE, PE, BI: Read the UART_RHR register. OE: Read the UART_LSR register.
001100	2	RX time-out	Stale data in RX FIFO	Read the UART_RHR register.
000100	2	RHR interrupt	DRDY (data ready) (FIFO disable) RX FIFO above trigger level (FIFO enable)	Read the UART_RHR register until the interrupt condition disappears.
000010	3	THR interrupt	TFE (UART_THR empty) (FIFO disable) TX FIFO below trigger level (FIFO enable)	Write to the UART_THR until the interrupt condition disappears.
000000	4	Modem status	See the UART_MSR register.	Read the UART_MSR register.
010000	5	XOFF interrupt/special character interrupt	Receive XOFF characters/special character	Receive XON character(s), if XOFF interrupt/read of the UART_IIR register, if special character interrupt.
100000	6	CTS, RTS, DSR	RTS pin or CTS pin or DSR change state from active (low) to inactive (high).	Read the UART_IIR register.

For the receiver-line status interrupt, the RX_FIFO_STS bit (UARTi.UART_LSR[7]) generates the interrupt.

For the XOFF interrupt, if an XOFF flow character detection caused the interrupt, the interrupt is cleared by an XON flow character detection. If special character detection caused the interrupt, the interrupt is cleared by a read of the UARTi.UART_IIR register.

21.3.5.2 Wake-Up Interrupt

Wake-up interrupt is a special interrupt that works differently from other interrupts. This interrupt is enabled when the UARTi.UART_SCR[4] RXCTSDSRWAKEUPENABLE bit is set to 1. The UARTi.UART_IIR register is not modified when this occurs; the UARTi.UART_SSR[1] RXCTSDSRWAKEUPSTS bit must be checked to detect a wake-up event.

When a wake-up interrupt occurs, it can be cleared only by resetting the UARTi.UART_SCR[4] RXCTSDSRWAKEUPENABLE bit. This bit must be re-enabled (set to 1) after the current wake-up interrupt event is processed to detect the next incoming wake-up event.

A wake-up interrupt can also occur if the WER[7] TXWAKEUPEN bit is set to 1 and one of the following occurs:

- THR interrupt occurred if it is enabled (omitted if TX DMA request is enabled).
- TX DMA request occurred if it is enabled.
- TX_STATUS_IT occurred if it is enabled (only IrDA and CIR modes). Cannot be used with THR interrupt.

CAUTION

Wake-Up interface implementation in IrDA mode is based on the UARTi_SIDLEACK low-to-high transition instead of the UARTi_SIDLEACK state.

This does not ensure wake-up event generation as expected when configured in smart-idle mode, and the system wakes up for a short period.

21.3.5.3 IrDA Mode Interrupt Management

21.3.5.3.1 IrDA Interrupts

The IrDA function generates interrupts. All interrupts can be enabled and disabled by writing to the appropriate bit in the interrupt enable register (UARTi.UART_IER). The interrupt status of the device can be checked by reading the interrupt identification register (UARTi.UART_IIR).

UART, IrDA, and CIR modes have different interrupts in the UART/IrDA/CIR module and, therefore, different UARTi.UART_IER and UARTi.UART_IIR mappings, depending on the selected mode.

IrDA modes have eight possible interrupts (see Table 21-12). The interrupt line is activated when any interrupt is generated (there is no priority).

Table 21-12. IrDA Mode Interrupts

UART_IIR Bit	Interrupt Type	Interrupt Source	Interrupt Reset Method
0	RHR interrupt	DRDY (data ready) (FIFO disable) RX FIFO above trigger level (FIFO enable)	Read the UART_RHR register until the interrupt condition disappears.
1	THR interrupt	TFE (UART_THR empty) (FIFO disable) TX FIFO below trigger level (FIFO enable)	Write to the UART_THR until the interrupt condition disappears.

Table 21-12. IrDA Mode Interrupts (continued)

UART_IIR Bit	Interrupt Type	Interrupt Source	Interrupt Reset Method
2	Last byte in RX FIFO	Last byte of frame in RX FIFO is available to be read at the RHR port.	Read the UART_RHR register.
3	RX overrun	Write to the UART_RHR register when the RX FIFO is full.	Read UART_RESUME register.
4	Status FIFO interrupt	Status FIFO triggers level reached.	Read STATUS FIFO.
5	TX status	1. UART_THR empty before EOF sent. Last bit of transmission of the IrDA frame occurred, but with an underrun error. OR 2. Transmission of the last bit of the IrDA frame completed successfully.	1. Read the UART_RESUME register. OR 2. Read the UART_IIR register.
6	Receiver line status interrupt	CRC, ABORT, or frame-length error is written into the STATUS FIFO.	Read the STATUS FIFO (read until empty - maximum of eight reads required).
7	Received EOF	Received end-of-frame	Read the UART_IIR register.

21.3.5.4 CIR Mode Interrupt Management

21.3.5.4.1 CIR Interrupts

The CIR function generates interrupts that can be enabled and disabled by writing to the appropriate bit in the interrupt enable register (UARTi.UART_IER). The interrupt status of the device can be checked by reading the interrupt identification register (UARTi.UART_IIR).

UART, IrDA, and CIR modes have different interrupts in the UART/IrDA/CIR module and, therefore, different UARTi.UART_IER and UARTi.UART_IIR mappings, depending on the selected mode.

Table 21-13 lists the interrupt modes to be maintained. In CIR mode, the sole purpose of the UARTi.UART_IIR[5] bit is to indicate that the last bit of infrared data was passed to the uart_cts_rctx pin.

Table 21-13. CIR Mode Interrupts

UART_IIR Bit Number	Interrupt Type	Interrupt Source	Interrupt Reset Method
0	RHR interrupt	DRDY (data ready) (FIFO disable) RX FIFO above trigger level (FIFO enable)	Read UART_RHR until interrupt condition disappears.
1	THR interrupt	TFE (UART_THR empty) (FIFO disable) TX FIFO below trigger level (FIFO enable)	Write to the UART_THR register until the interrupt condition disappears.
2	RX_STOP_IT	Receive stop interrupt (depending on value set in the BOF Length Register (UART_EBLR).	Read IIR
3	RX overrun	Write to RHR when RX FIFO is full.	Read RESUME register.
4	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode
5	TX status	Transmission of the last bit of the frame is complete successfully.	Read the UART_IIR register.
6	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode
7	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode

21.3.6 FIFO Management

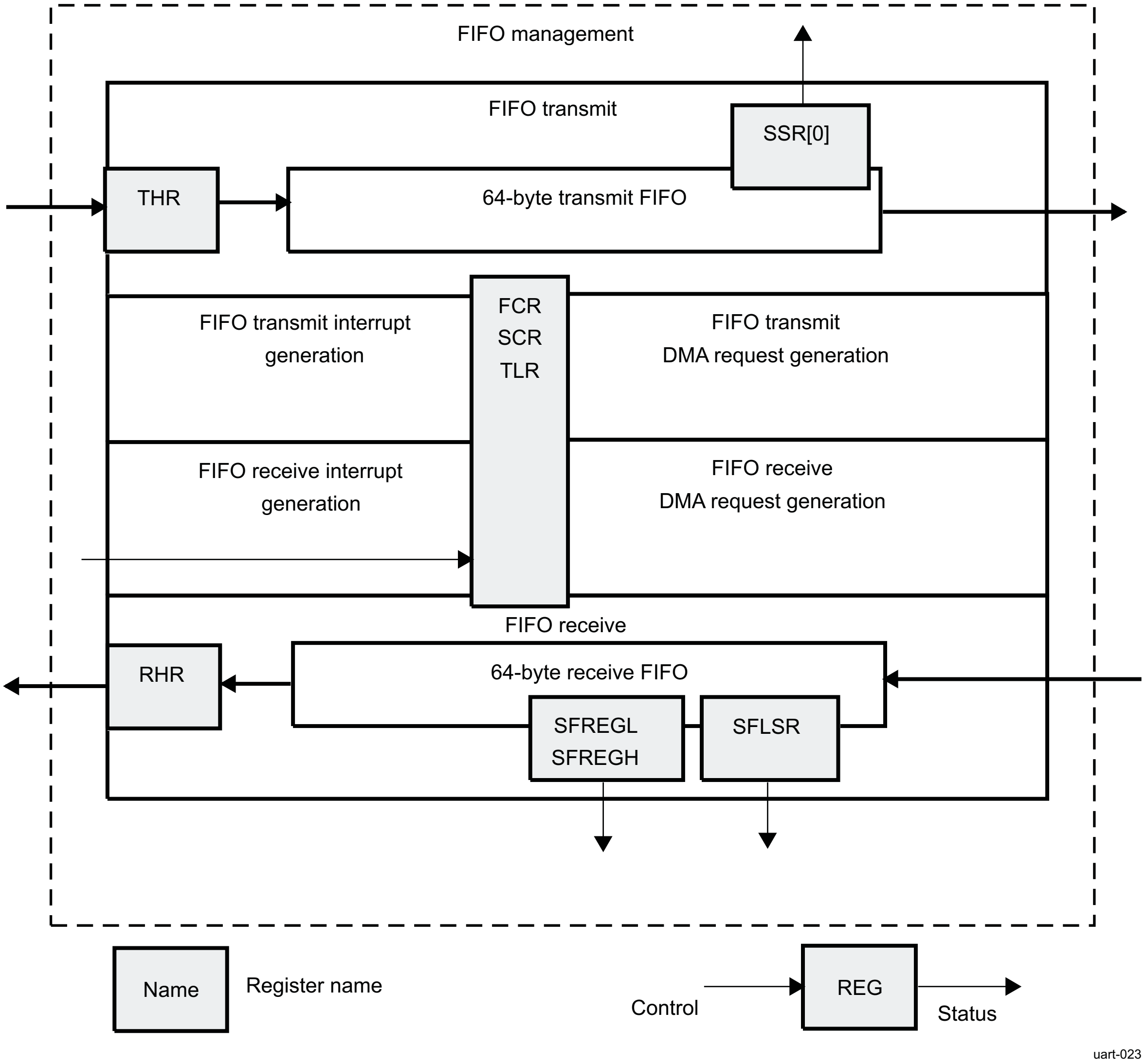
The FIFO is accessed by reading and writing the UARTi.UART_RHR and UARTi.UART_THR registers. Parameters are controlled using the FIFO control register (UARTi.UART_FCR) and supplementary control register (UARTi.UART_SCR). Reading the UARTi.UART_SSR[0] TX_FIFO_FULL bit at 1 means the FIFO is full.

The UARTi.UART_TLR register controls the FIFO trigger level, which enables DMA and interrupt generation. After reset, transmit (TX) and receive (RX) FIFOs are disabled; thus, the trigger level is the default value of 1 byte. Figure 21-4 shows the FIFO management registers.

NOTE: Data in the UARTi.UART_RHR register is not overwritten when an overflow occurs.

NOTE: The UARTi.UART_SFLSR, UARTi.UART_SFREGL, and UARTi.UART_SFREGH status registers are used in IrDA mode only. For use, see Section 21.3.8.2.6, *IrDA Data Formatting*.

Figure 21-4. FIFO Management Registers



21.3.6.1 FIFO Trigger

21.3.6.1.1 Transmit FIFO Trigger

Table 21-14 lists the TX FIFO trigger level settings.

Table 21-14. TX FIFO Trigger Level Setting Summary

UART_SCR[6]	UART_TLR[3:0]	TX FIFO Trigger Level
0	= 0x0	Defined by the UARTi.UART_FCR[5:4] TX_FIFO_TRIG bit field (8,16, 32, or 56 spaces)
0	!= 0x0	Defined by the UARTi.UART_TLR[3:0] TX_FIFO_TRIG_DMA bit field (from 4 to 60 spaces with a granularity of 4 spaces)
1	Value	Defined by the concatenated value of TX_FIFO_TRIG_DMA and TX_FIFO_TRIG (from 1 to 63 spaces with a granularity of 1 space) **Note:** The combination of TX_FIFO_TRIG_DMA = 0x0 and TX_FIFO_TRIG = 0x0 (all zeros) is not supported (minimum of one space required). All zeros result in unpredictable behavior.

21.3.6.1.2 Receive FIFO Trigger

Table 21-15 lists the RX FIFO trigger level settings.

Table 21-15. RX FIFO Trigger Level Setting Summary

UART_SCR[7]	UART_TLR[7:4]	RX FIFO Trigger Level
0	= 0x0	Defined by the UARTi.UART_FCR[7:6] RX_FIFO_TRIG bit field (8,16, 56, or 60 characters)
0	!= 0x0	Defined by the UARTi.UART_TLR[7:4] RX_FIFO_TRIG_DMA bit field (from 4 to 60 characters with a granularity of 4 characters)
1	Value	Defined by the concatenated value of RX_FIFO_TRIG_DMA and RX_FIFO_TRIG (from 1 to 63 characters with a granularity of 1 character) **Note:** The combination of RX_FIFO_TRIG_DMA = 0x0 and RX_FIFO_TRIG = 0x0 (all zeros) is not supported (minimum of one character required). All zeros result in unpredictable behavior.

The receive threshold is programmed using the UARTi.UART_TCR[7:4] RX_FIFO_TRIG_START and UARTi.UART_TCR[3:0] RX_FIFO_TRIG_HALT bit fields:

- Trigger levels from 0 to 60 bytes are available with a granularity of 4 (trigger level = 4 x [4-bit register value]).
- To ensure correct device operation, ensure that RX_FIFO_TRIG_HALT RX_FIFO_TRIG when auto-RTS is enabled.

  Delay = [4 + 16 x (1 + CHAR_LENGTH + Parity + Stop 0.5)] x Baud_rate + 4 x FCLK

  NOTE: The RTS signal is deasserted after the UART module receives the data over RX_FIFO_TRIG_HALT. Delay means how long the UART module takes to deassert the RTS signal after reaching RX_FIFO_TRIG_HALT.

- In FIFO interrupt mode with flow control, ensure that the trigger level to HALT transmission is greater than or equal to the RX FIFO trigger level (the UARTi.UART_TCR[7:4] RX_FIFO_TRIG_START bit field or the UARTi.UART_FCR[7:6] RX_FIFO_TRIG bit field); otherwise, FIFO operation stalls. In FIFO DMA mode with flow control, this concept does not exist, because a DMA request is sent when a byte is received.

21.3.6.2 FIFO Interrupt Mode

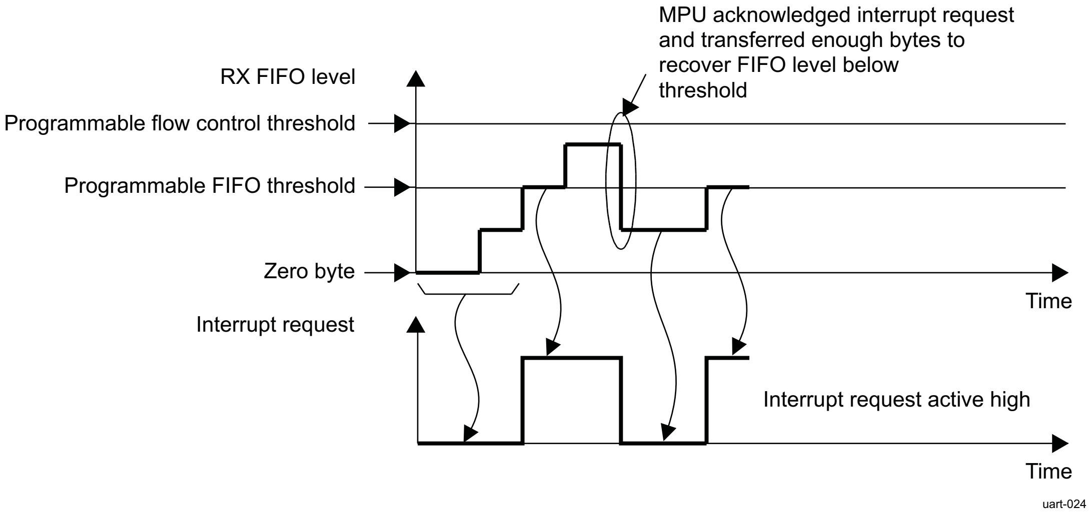
In FIFO interrupt mode (the FIFO control register UARTi.UART_FCR[0] FIFO_EN bit is set to 1 and relevant interrupts are enabled by the UARTi.UART_IER register), an interrupt signal informs the processor of the status of the receiver and transmitter. These interrupts are raised when the RX/TX FIFO threshold (the UARTi.UART_TLR[7:4] RX_FIFO_TRIG_DMA and UARTi.UART_TLR[3:0] TX_FIFO_TRIG_DMA bit fields or the UARTi.UART_FCR[7:6] RX_FIFO_TRIG and UARTi.UART_FCR[5:4] TX_FIFO_TRIG bit fields, respectively) is reached.

The interrupt signals instruct the MPU to transfer data to the destination (from the UART in receive mode and/or from any source to the UART FIFO in transmit mode).

When UART flow control is enabled with interrupt capabilities, the UART flow control FIFO threshold (the UARTi.UART_TCR[3:0] RX_FIFO_TRIG_HALT bit field) must be greater than or equal to the RX FIFO threshold.

Figure 21-5 shows the generation of the RX FIFO interrupt request.

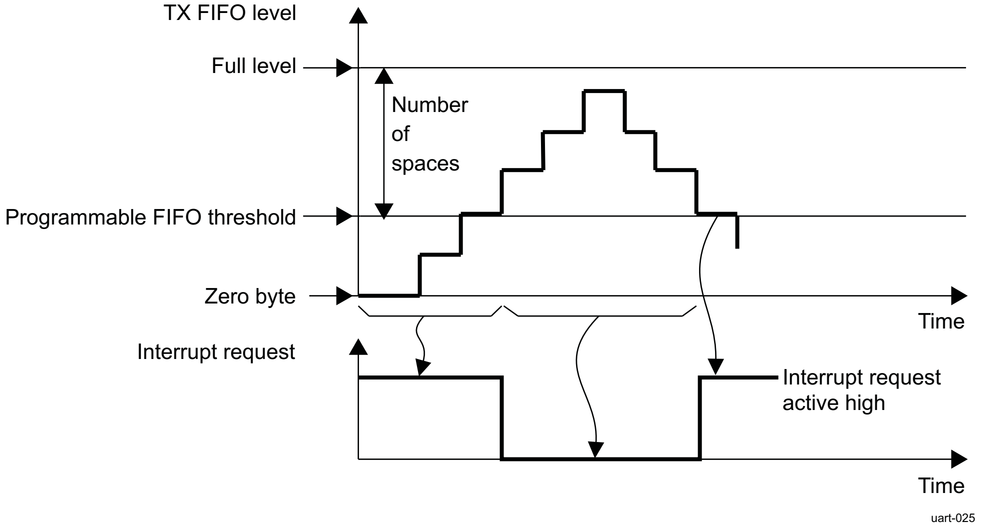
Figure 21-5. RX FIFO Interrupt Request Generation



In receive mode, no interrupt is generated until the RX FIFO reaches its threshold. Once low, the interrupt can be deasserted only when the MPU has handled enough bytes to put the FIFO level below threshold. The flow control threshold is set at a higher value than the FIFO threshold.

Figure 21-6 shows the generation of the TX FIFO interrupt request.

Figure 21-6. TX FIFO Interrupt Request Generation



In transmit mode, an interrupt request is automatically asserted when the TX FIFO is empty. This request is deasserted when the TX FIFO crosses the threshold level. The interrupt line is deasserted until a sufficient number of elements is transmitted to go below the TX FIFO threshold.

21.3.6.3 FIFO Polled Mode Operation

In FIFO polled mode (the UARTi.UART_FCR[0] FIFO_EN bit is set to 0 and the relevant interrupts are disabled by the UARTi.UART_IER register), the status of the receiver and transmitter can be checked by polling the line status register (UARTi.UART_LSR).

This mode is an alternative to the FIFO interrupt mode of operation in which the status of the receiver and transmitter is automatically determined by sending interrupts to the MPU.

21.3.6.4 FIFO DMA Mode Operation

Although DMA operation includes four modes (DMA modes 0 through 3), assume that mode 1 is used. (Mode 2 and mode 3 are legacy modes that use only one DMA request for each module.)

In mode 2, the remaining DMA request is used for RX. In mode 3, the remaining DMA request is used for TX.

DMA requests in mode 2 and mode 3 use the following signals:

- S_DMA_48
- S_DMA_50
- S_DMA_52/D_DMA_10
- S_DMA_54

The following signals are not used by the module in mode 2 and mode 3:

- S_DMA_49
- S_DMA_51
- S_DMA_53/D_DMA_11
- S_DMA_55

These signals can be selected as follows:

- When the UARTi.UART_SCR[0] DMA_MODE_CTL bit is set to 0, setting the UARTi.UART_FCR[3]DMA_MODE bit to 0 enables DMA mode 0. Setting the DMA_MODE bit to 1 enables DMA mode 1.
- When the DMA_MODE_CTL bit is set to 1, the UARTi.UART_SCR[2:1]DMA_MODE_2 bit field determines DMA mode 0 to mode 3 based on the supplementary control register (UART_SCR) description.

For example:

- If no DMA operation is desired, set the DMA_MODE_CTL bit to 1 and the DMA_MODE_2 bit field to 0x0. (The DMA_MODE bit is discarded.)
- If DMA mode 1 is desired, set the DMA_MODE_CTL bit to 0 and the DMA_MODE bit to 1, or set the DMA_MODE_CTL bit to 1 and the DMA_MODE_2 bit field to 01. (The DMA_MODE bit is discarded.)

If the FIFOs are disabled (the UARTi.UART_FCR[0] FIFO_EN bit is set to 0), the DMA occurs in single-character transfers.

When DMA mode 0 is programmed, the signals associated with DMA operation are not active.

Depending on UART_MDR3[2] SET_DMA_TX_THRESHOLD, the threshold can be programmed different ways:

- SET_TX_DMA_THRESHOLD = 1:

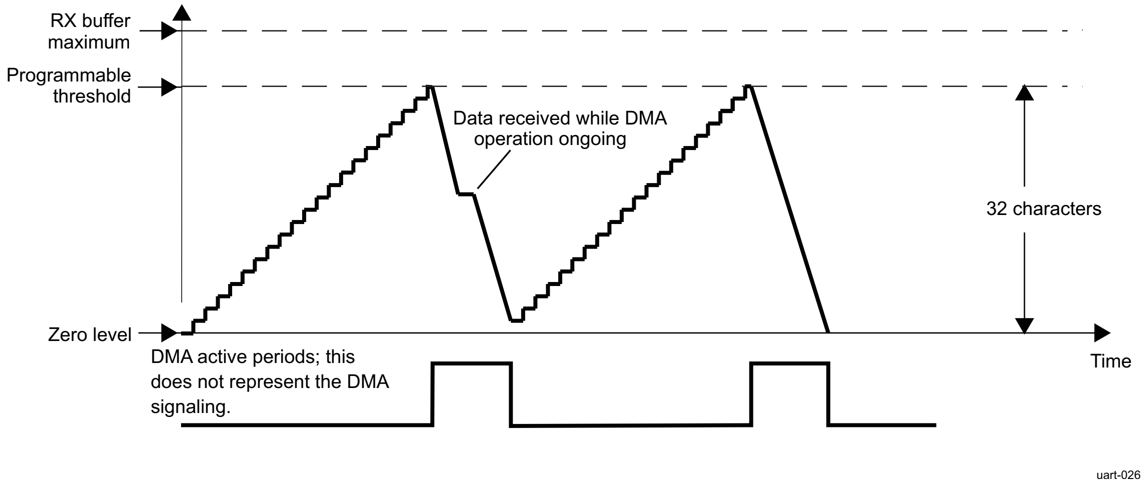
  The threshold value will be the value of the UART_TX_DMA_THRESHOLD register. If SET_TX_DMA_THRESHOLD + TX trigger spaces 64, then the default method of threshold is used: threshold value = TX FIFO size.
- SET_TX_DMA_THRESHOLD = 0:

  The threshold value = TX FIFO size - TX trigger space. The TX DMA line is asserted if the TX FIFO level is lower then the threshold. It remains asserted until TX trigger spaces number of bytes are written into the FIFO. The DMA line is then deasserted and the FIFO level is compared with the threshold value.

21.3.6.4.1 DMA Transfers (DMA Mode 1, 2, or 3)

Figure 21-7 through Figure 21-10 show the supported DMA operations.

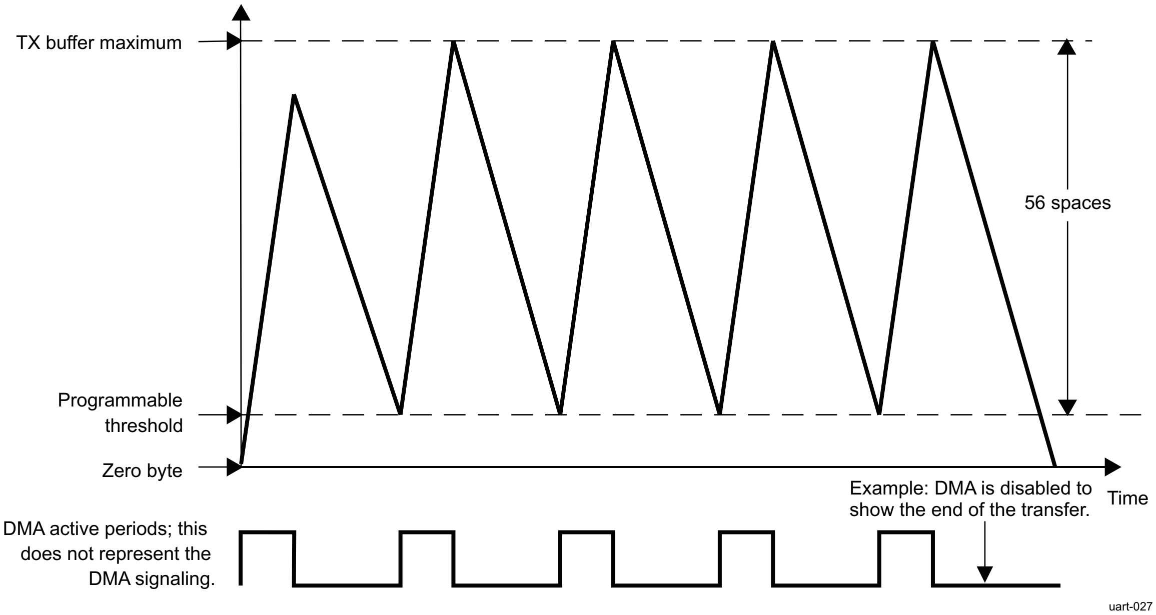
Figure 21-7. Receive FIFO DMA Request Generation (32 Characters)



In receive mode, a DMA request is generated when the RX FIFO reaches its threshold level defined in the trigger level register (UARTi.UART_TLR). This request is deasserted when the number of bytes defined by the threshold level is read by the sDMA.

In transmit mode, a DMA request is automatically asserted when the TX FIFO is empty. This request is deasserted when the number of bytes defined by the number of spaces in the UARTi.UART_TLR register is written by the sDMA. If an insufficient number of characters is written, the DMA request stays active.

Figure 21-8. Transmit FIFO DMA Request Generation (56 Spaces)



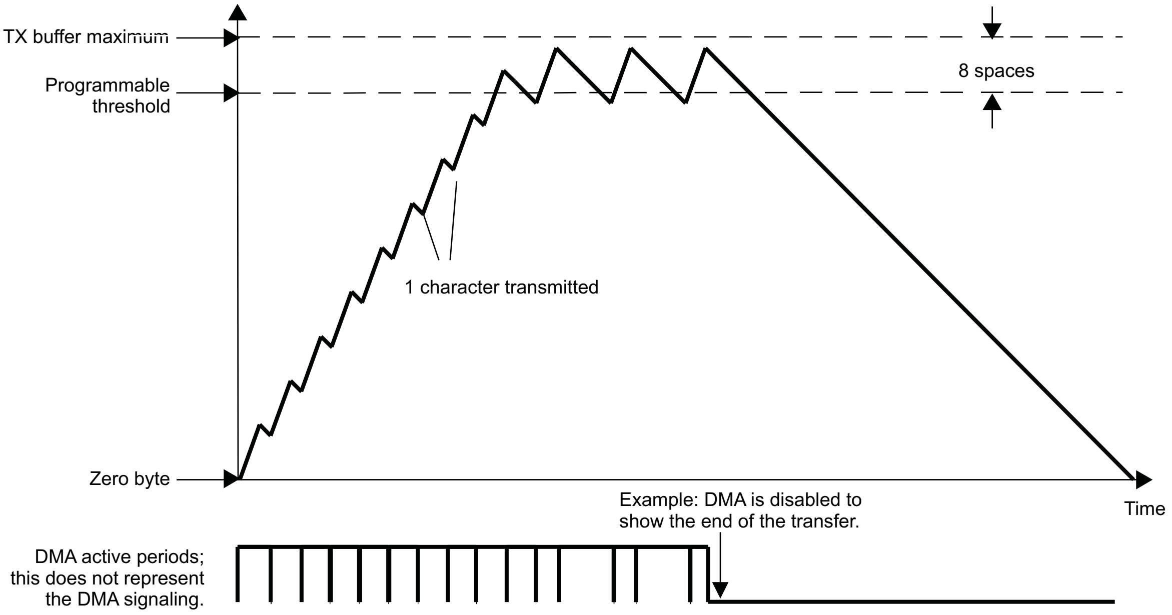
The DMA request is again asserted if the FIFO can receive the number of bytes defined by the UARTi.UART_TLR register.

The threshold can be programmed in a number of ways. Figure 21-8 shows a DMA transfer operating with a space setting of 56 that can arise from using the auto settings in the UARTi.UART_FCR[5:4] TX_FIFO_TRIG bit field or the UARTi.UART_TLR[3:0] TX_FIFO_TRIG_DMA bit field concatenated with the TX_FIFO_TRIG bit field.

The setting of 56 spaces in the UART/IrDA/CIR module must correlate with the settings of the sDMA so that the buffer does not overflow (program the DMA request size of the LH controller to equal the number of spaces in the UART/IrDA/CIR module).

Figure 21-9 shows an example with eight spaces to show the buffer level crossing the space threshold. The LH DMA controller settings must correspond to those of the UART/IrDA/CIR module.

Figure 21-9. Transmit FIFO DMA Request Generation (8 Spaces)



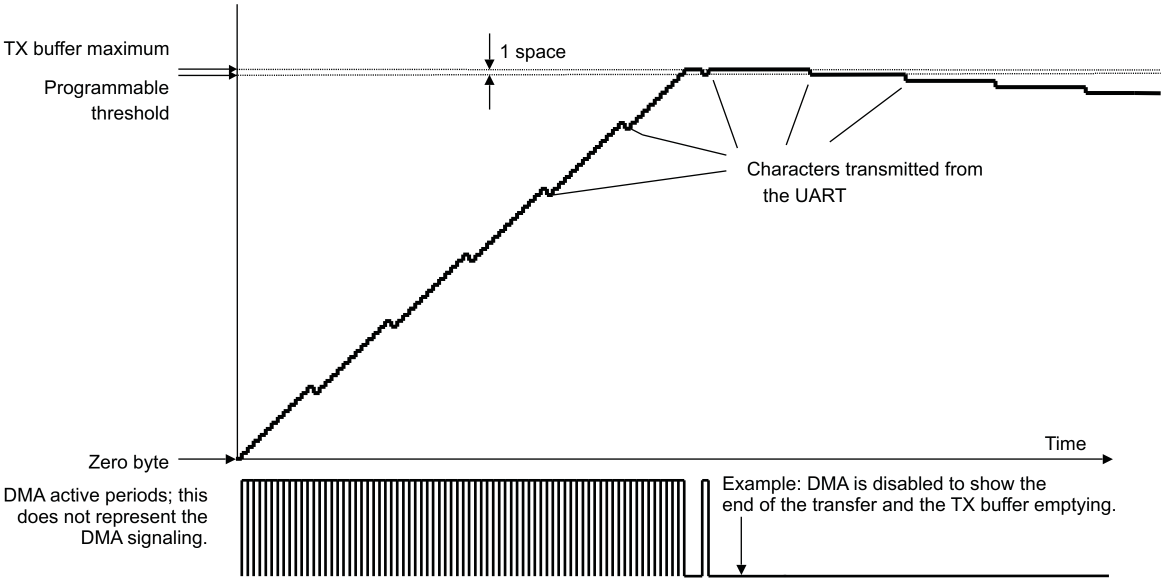
uart-028

The next example shows the setting of one space that uses the DMA for each transfer of one character to the transmit buffer (see Figure 21-10). The buffer is filled faster than the baud rate at which data is transmitted to the TX pin. Eventually, the buffer is completely full and the DMA operations stop transferring data to the transmit buffer.

On two occasions, the buffer holds the maximum amount of data words; shortly after this, the DMA is disabled to show the slower transmission of the data words to the TX pin. Eventually, the buffer is emptied at the rate specified by the baud rate settings of the UARTi.UART_DLL and UARTi.UART_DLH registers.

The DMA settings must correspond to the system LH DMA controller settings to ensure correct operation of this logic.

Figure 21-10. Transmit FIFO DMA Request Generation (1 Space)



uart-029

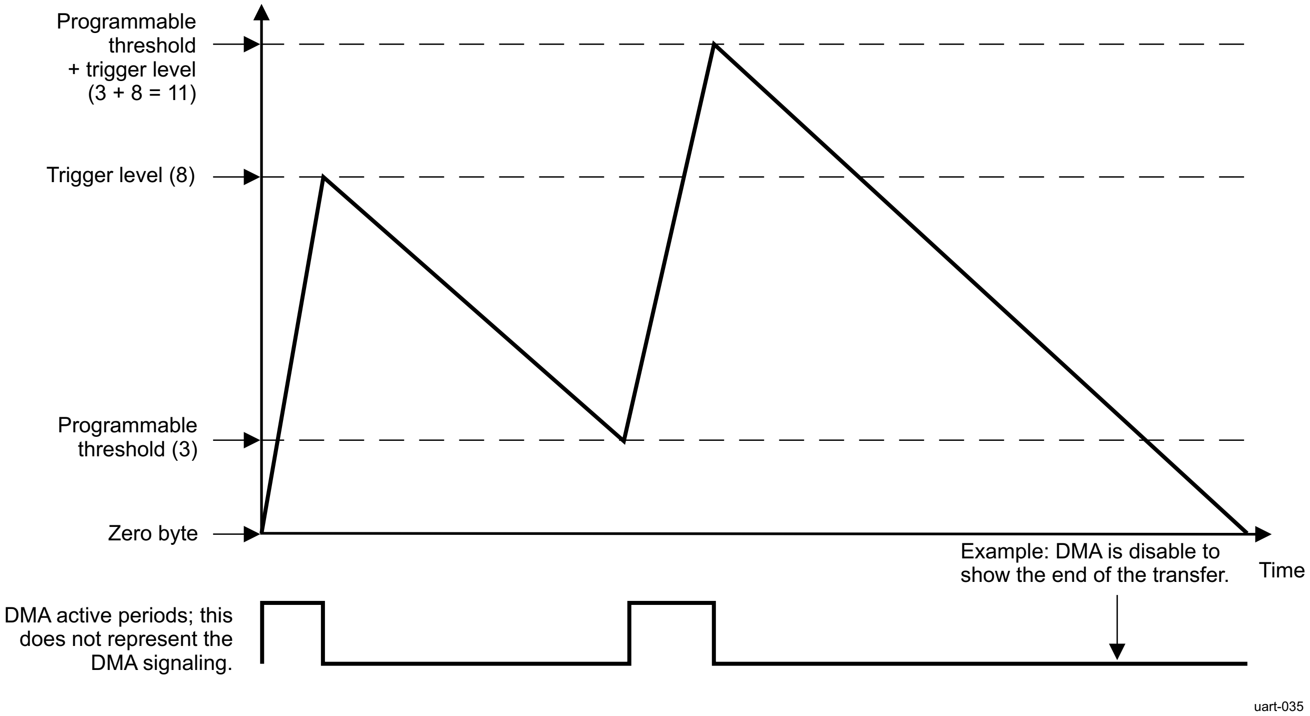
The final example shows the setting of eight spaces, but setting the TX DMA threshold directly by setting the UART_MDR3[1]SET_DMA_RX_THRESHOLD bit and the UART_TX_DMA_THRESHOLD register (see Figure 21-11). In the example, UART_TX_DMA_THRESHOLD[2:0]TX_DMA_THRESHOLD = 3 and the trigger level is 8. The buffer is filled at a faster rate than the baud rate transmits data to the TX pin. The buffer is filled with 8 bytes and the DMA operations stop transferring data to the transmit buffer. When the buffer is emptied to the threshold level by transmission, the DMA operation activates again to fill the buffer with 8 bytes.

Eventually, the buffer is emptied at the rate specified by the baud rate settings of the UART_DLL and UART_DLH registers.

If the selected threshold level plus the trigger level exceed the maximum buffer size, the original TX DMA threshold method is used to prevent TX overrun, regardless of the value of the UART_MDR3[1]SET_DMA_RX_THRESHOLD bit.

The DMA settings must correspond to the settings of the system local host DMA controller to ensure the correct operation of this logic.

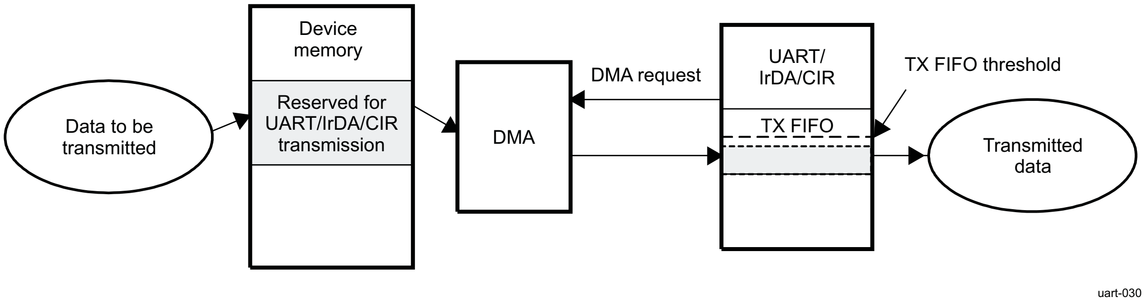
Figure 21-11. Transmit FIFO DMA Request Generation Using Direct TX DMA Threshold Programming. (Threshold = 3; Spaces = 8)



21.3.6.4.2 DMA Transmission

Figure 21-12 shows DMA transmission.

Figure 21-12. DMA Transmission



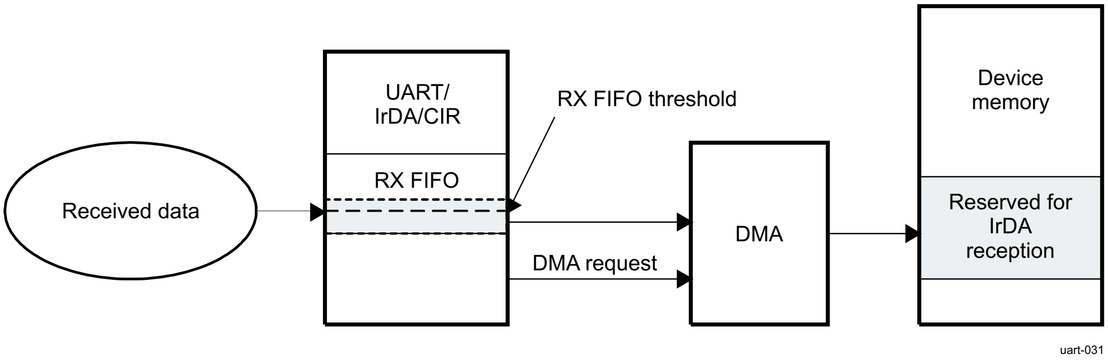
1. Data to be transmitted are put in the device memory reserved for UART/IrDA/CIR transmission by the DMA:
   a. Until the TX FIFO trigger level is not reached, a DMA request is generated
   b. An element (1 byte) is transferred from the SDRAM to the TX FIFO at each DMA request (DMA element synchronization).
2. Data in the TX FIFO are automatically transmitted.
3. The end of the transmission is signaled by the UARTi.UART_THR empty (TX FIFO empty).

NOTE: In IrDA mode, the transmission does not end immediately after the TX FIFO empties, at which point the last data byte, the CRC field, and the stop flag still must be transmitted; thus, the end of transmission occurs a few milliseconds after the UARTi.UART_THR register empties.

21.3.6.4.3 *DMA Reception*

Figure 21-13 shows DMA reception.

Figure 21-13. DMA Reception



1. Enable the reception.
2. Received data are put in the RX FIFO.
3. Data are transferred from the RX FIFO to the device memory by the DMA:
   a. At each received byte, the RX FIFO trigger level (one character) is reached and a DMA request is generated.
   b. An element (1 byte) is transferred from the RX FIFO to the SDRAM at each DMA request (DMA element synchronization).
4. The end of the reception is signaled by the EOF interrupt.

21.3.7 Mode Selection

21.3.7.1 Register Access Modes

21.3.7.1.1 *Operational Mode and Configuration Modes*

Register access depends on the register access mode, although register access modes are not correlated to functional mode selection. Three different modes are available:

- Operational mode
- Configuration mode A
- Configuration mode B

Operational mode is the selected mode when the function is active; serial data transfer can be performed in this mode.

Configuration mode A and configuration mode B are used during module initialization steps. These modes enable access to configuration registers, which are hidden in the operational mode. The modes are used when the module is inactive (no serial data transfer processed) and only for initialization or reconfiguration of the module.

The value of the UARTi.UART_LCR register determines the register access mode (see Table 21-16).

Table 21-16. UART/IrDA/CIR Register Access Mode Programming (Using UART_LCR)

Mode	Condition
Configuration mode A	UART_LCR[7] = 0x1 and UART_LCR[7:0] != 0xBF
Configuration mode B	UART_LCR[7] = 0x1 and UART_LCR[7:0] = 0xBF
Operational mode	UART_LCR[7] = 0x0

21.3.7.1.2 Register Access Submode

In each access register mode (operational mode or configuration mode A/B), some register accesses are conditional on the programming of a submode (MSR_SPR, TCR_TLR, and XOFF).

Table 21-17 through Table 21-19 summarize the register access submodes.

Table 21-17. Subconfiguration Mode A Summary

Mode	Condition
MSR_SPR	(UART_EFR[4] = 0x0 or UART_MCR[6] = 0x0)
TCR_TLR	UART_EFR[4] = 0x1 and UART_MCR[6] = 0x1

Table 21-18. Subconfiguration Mode B Summary

Mode	Condition
TCR_TLR	UART_EFR[4] = 0x1 and UART_MCR[6] = 0x1
XOFF	(UART_EFR[4] = 0x0 or UART_MCR[6] = 0x0)

Table 21-19. Suboperational Mode Summary

Mode	Condition
MSR_SPR	UART_EFR[4] = 0x0 or UART_MCR[6] = 0x0
TCR_TLR	UART_EFR[4] = 0x1 and UART_MCR[6] = 0x1

21.3.7.1.3 Registers Available for the Register Access Modes

Table 21-20 lists the names of the register bits in each access register mode. Gray shading indicates that the register does not depend on the register access mode (available in all modes).

Table 21-20. UART/IrDA/CIR Register Access Mode Overview

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x000	UART_DLL	UART_DLL	UART_DLL	UART_DLL	UART_RHR	UART_THR
0x004	UART_DLH	UART_DLH	UART_DLH	UART_DLH	UART_IER	UART_IER
0x008	UART_IIR	UART_FCR	UART_EFR	UART_EFR	UART_IIR	UART_FCR
0x00C	UART_LCR	UART_LCR	UART_LCR	UART_LCR	UART_LCR	UART_LCR
0x010	UART_MCR	UART_MCR	UART_XON1_ADDR1	UART_XON1_ADDR1	UART_MCR	UART_MCR

Table 21-20. UART/IrDA/CIR Register Access Mode Overview (continued)

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x014	UART_LSR	–	UART_XON2_ADDR2	UART_XON2_ADDR2	UART_LSR	–
0x018	UART_MSR (1)/UART_TCR (2)	UART_TCR (2)	UART_TCR (2)/UART_XOFF1 (3)	UART_TCR (2)/UART_XOFF1 (3)	UART_MSR (1)/UART_TCR (2)	UART_TCR (2)
0x01C	UART_SPR (1)/UART_TLR (2)	UART_SPR (1)/UART_TLR (2)	UART_TLR (2)/UART_XOFF2 (3)	UART_TLR (2)/UART_XOFF2 (3)	UART_SPR (1)/UART_TLR (2)	UART_SPR (1)/UART_TLR (2)
0x020	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1
0x024	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2
0x028	UART_SFLSR	UART_TXFLL	UART_SFLSR	UART_TXFLL	UART_SFLSR	UART_TXFLL
0x02C	UART_RESUME	UART_TXFLH	UART_RESUME	UART_TXFLH	UART_RESUME	UART_TXFLH
0x030	UART_SFREGL	UART_RXFLL	UART_SFREGL	UART_RXFLL	UART_SFREGL	UART_RXFLL
0x034	UART_SFREGH	UART_RXFLH	UART_SFREGH	UART_RXFLH	UART_SFREGH	UART_RXFLH
0x038	UART_UASR	–	UART_UASR	–	UART_BLR	UART_BLR
0x03C	–	–	–	–	UART_ACREG	UART_ACREG
0x040	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR
0x044	UART_SSR	–	UART_SSR	–	UART_SSR	–
0x048	–	–	–	–	UART_EBLR	UART_EBLR
0x050	UART_MVR	–	UART_MVR	–	UART_MVR	–
0x054	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC
0x058	UART_SYSS	–	UART_SYSS	–	UART_SYSS	–
0x05C	UART_WER	UART_WER	UART_WER	UART_WER	UART_WER	UART_WER
0x060	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS
0x064	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL
0x068	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL
0x06C	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2
0x070	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2
0x074	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL
0x080	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3
0x084	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD

(1) MSR_SPR mode is active (see Section 21.3.7.1.2, *Register Access Submode*)

(2) TCR_TLR mode is active (see Section 21.3.7.1.2, *Register Access Submode*)

(3) XOFF mode is active (see Section 21.3.7.1.2, *Register Access Submode*)

21.3.7.2 UART/IrDA (SIR, MIR, FIR)/CIR Mode Selection

To select a mode, set the UARTi.UART_MDR1[2:0] MODESELECT bit field (see Table 21-21).

Table 21-21. UART Mode Selection

Value	Mode
0x0:	UART 16x mode
0x1:	SIR mode
0x2:	UART 16x auto-baud
0x3:	UART 13x mode
0x4:	MIR mode
0x5:	FIR mode
0x6:	CIR mode

MODESELECT is effective when the module is in operational mode (see Section 21.3.7.1, *Register Access Modes*).

21.3.7.2.1 Registers Available for the UART Function

Only the registers listed in Table 21-22 are used for the UART function.

Table 21-22. UART Mode Register Overview[1] [2]

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x000	UART_DLL	UART_DLL	UART_DLL	UART_DLL	UART_RHR	UART_THR
0x004	UART_DLH	UART_DLH	UART_DLH	UART_DLH	UART_IER(UART)	UART_IER(UART)
0x008	UART_IIR	UART_FCR	UART_EFR[4]	UART_EFR[4]	UART_IIR(UART)	UART_FCR(UART)
0x00C	UART_LCR	UART_LCR	UART_LCR	UART_LCR	UART_LCR	UART_LCR
0x010	UART_MCR	UART_MCR	UART_XON1_ADDR1	UART_XON1_ADDR1	UART_MCR	UART_MCR
0x014	UART_LSR(UART)	–	UART_XON2_ADDR2	UART_XON2_ADDR2	UART_LSR(UART)	–
0x018	UART_MSR/UART_TCR	UART_TCR	UART_XOFF1/UART_TCR	UART_XOFF1/UART_TCR	UART_MSR/UART_TCR	UART_TCR
0x01C	UART_TLR/UART_SPR	UART_TLR/UART_SPR	UART_TLR/UART_XOFF2	UART_TLR/UART_XOFF2	UART_TLR/UART_SPR	UART_TLR/UART_SPR
0x020	UART_MDR1	UART_MDR1[2:0]	UART_MDR1[2:0]	UART_MDR1[2:0]	UART_MDR1[2:0]	UART_MDR1[2:0]
0x024	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2
0x028	–	–	–	–	–	–
0x02C	–	–	–	–	–	–
0x030	–	–	–	–	–	–
0x034	–	–	–	–	–	–
0x038	UART_UASR	–	UART_UASR	–	–	–
0x03C	–	–	–	–	–	–
0x040	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR
0x044	UART_SSR	–	UART_SSR	–	UART_SSR	–
0x048	–	–	–	–	–	–
0x050	UART_MVR	–	UART_MVR	–	UART_MVR	–
0x054	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC
0x058	UART_SYSS	–	UART_SYSS	–	UART_SYSS	–

[1] REGISTER_NAME(UART) notation indicates that the register exists for other functions (IrDA or CIR), but fields have different meanings for other functions.

[2] REGISTER_NAME[m:n] notation indicates that only register bits numbered m to n apply to the UART function.

Table 21-22. UART Mode Register Overview(1) (2) (continued)

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x05C	UART_WER	UART_WER	UART_WER	UART_WER	UART_WER	UART_WER
0x060	–	–	–	–	–	–
0x064	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL
0x068	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL
0x06C	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2
0x070	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2
0x074	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL
0x080	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3
0x084	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD

21.3.7.2.2 *Registers Available for the IrDA Function*

Only the registers listed in Table 21-23 are used for the IrDA function.

Table 21-23. IrDA Mode Register Overview(1) (2)

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x000	UART_DLL	UART_DLL	UART_DLL	UART_DLL	UART_RHR	UART_THR
0x004	UART_DLH	UART_DLH	UART_DLH	UART_DLH	UART_IER(IrDA)	UART_IER(IrDA)
0x008	UART_IIR	UART_FCR	UART_EFR[4]	UART_EFR[4]	UART_IIR(IrDA)	UART_FCR(IrDA)
0x00C	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]
0x010	–	–	UART_XON1_ADDR1	UART_XON1_ADDR1	–	–
0x014	UART_LSR(IrDA)	–	UART_XON2_ADDR2	UART_XON2_ADDR2	UART_LSR(IrDA)	–
0x018	UART_MSR/UART_TCR	UART_TCR	UART_TCR	UART_TCR	UART_MSR/UART_TCR	UART_TCR
0x01C	UART_TLR/UART_SPR	UART_TLR/UART_SPR	UART_TLR	UART_TLR	UART_TLR/UART_SPR	UART_TLR/UART_SPR
0x020	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1
0x024	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2
0x028	UART_SFLSR	UART_TXFLL	UART_SFLSR	UART_TXFLL	UART_SFLSR	UART_TXFLL
0x02C	UART_RESUME	UART_TXFLH	UART_RESUME	UART_TXFLH	UART_RESUME	UART_TXFLH
0x030	UART_SFREGL	UART_RXFLL	UART_SFREGL	UART_RXFLL	UART_SFREGL	UART_RXFLL
0x034	UART_SFREGH	UART_RXFLH	UART_SFREGH	UART_RXFLH	UART_SFREGH	UART_RXFLH

(1) REGISTER_NAME(UART) notation indicates that the register exists for other functions (IrDA or CIR), but fields have different meanings for other functions.

(2) REGISTER_NAME[m:n] notation indicates that only register bits numbered m to n apply to the UART function.

Table 21-23. IrDA Mode Register Overview[1] [2] (continued)

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x038	–	–	–	–	UART_BLR	UART_BLR
0x03C	–	–	–	–	UART_ACREG	UART_ACREG
0x040	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR
0x044	UART_SSR	–	UART_SSR	–	UART_SSR	–
0x048	–	–	–	–	UART_EBLR	UART_EBLR
0x050	UART_MVR	–	UART_MVR	–	UART_MVR	–
0x054	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC
0x058	UART_SYSS	–	UART_SYSS	–	UART_SYSS	–
0x05C	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]
0x060	–	–	–	–	–	–
0x064	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL
0x068	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL
0x06C	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2
0x070	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2
0x074	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL
0x080	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3
0x084	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD

21.3.7.2.3 Registers Available for the CIR Function

Only the registers listed in Table 21-24 are used for the CIR function.

Table 21-24. CIR Mode Register Overview[1] [2]

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x000	UART_DLL	UART_DLL	UART_DLL	UART_DLL	–	UART_THR
0x004	UART_DLH	UART_DLH	UART_DLH	UART_DLH	UART_IER(CIR)	UART_IER(CIR)
0x008	UART_IIR	UART_FCR	UART_EFR	UART_EFR	UART_IIR(CIR)	UART_FCR(CIR)
0x00C	UART_LCR	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]
0x010	–	–	–	–	–	–
0x014	UART_LSR(IrDA)	–	–	–	UART_LSR(IrDA)	–
0x018	UART_MSR/UART_TCR	UART_TCR	UART_TCR	UART_TCR	UART_MSR/UART_TCR	UART_TCR
0x01C	UART_TLR/UART_SPR	UART_TLR/UART_SPR	UART_TLR	UART_TLR	UART_TLR/UART_SPR	UART_TLR/UART_SPR

---

[1] REGISTER_NAME(UART) notation indicates that the register exists for other functions (IrDA or CIR), but fields have different meanings for other functions.

[2] REGISTER_NAME[m:n] notation indicates that only register bits numbered m to n apply to the UART function.

Table 21-24. CIR Mode Register Overview[1] [2] (continued)

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x020	UART_MDR1[3:0]	UART_MDR1[3:0]	UART_MDR1[3:0]	UART_MDR1[3:0]	UART_MDR1[3:0]	UART_MDR1[3:0]
0x024	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2
0x028	–	–	–	–	–	–
0x02C	UART_RESUME	–	UART_RESUME	–	UART_RESUME	–
0x030	–	–	–	–	–	–
0x034	–	–	–	–	–	–
0x038	–	–	–	–	–	–
0x03C	–	–	–	–	UART_ACREG	UART_ACREG
0x040	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR
0x044	UART_SSR	–	UART_SSR	–	UART_SSR	–
0x048	–	–	–	–	UART_EBLR	UART_EBLR
0x050	UART_MVR	–	UART_MVR	–	UART_MVR	–
0x054	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC
0x058	UART_SYSS	–	UART_SYSS	–	UART_SYSS	–
0x05C	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]
0x060	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS
0x064	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL
0x068	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL
0x06C	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2
0x070	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2
0x074	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL
0x080	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3
0x084	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD

21.3.8 Protocol Formatting

The UART/IRDA module can operate in seven different modes:

1. UART 16x mode (≤230.4 Kbits/s), UART16x ≤460Kbits/s if MDR3[1] is set
2. UART 16x mode with autobauding (≥1200 bits/s and ≤115.2 Kbits/s) if MDR3[1] is not set
3. UART 13x mode (≥460.8 Kbits/s) if MDR3[1] is not set
4. IrDA SIR mode (≤115.2 Kbits/s) if MDR3[1] is not set
5. IrDA MIR mode (0.576 and 1.152 Mbits/s) if MDR3[1] is not set
6. IrDA FIR mode (4 Mbits/s) if MDR3[1] is not set
7. CIR mode (programmable modulation rates specific to remote control applications) if MDR3[1] is not set

The module performs a serial-to-parallel conversion on received data characters and a parallel-to-serial conversion on transmitted data characters by the processor. The complete status of each channel of the module and each received character/frame can be read at any time during functional operation via the line status register (LSR).

The module can be placed in an alternate mode (FIFO mode) to relieve the processor of excessive software overhead by buffering received/transmitted characters.

Both the receiver and transmitter FIFOs can store up to 64 bytes of data (plus three additional bits of error status per byte for the receiver FIFO) and have selectable trigger levels. Both interrupts and DMA are available to control the data flow between the LH and the module.

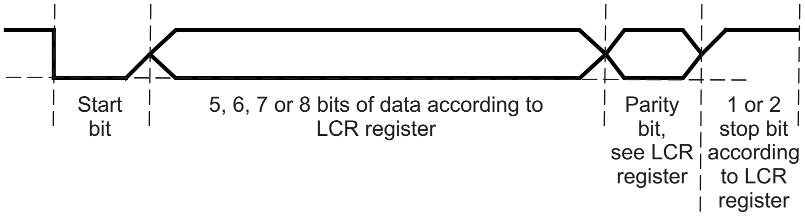
21.3.8.1 UART Mode

The UART uses a wired interface for serial communication with a remote device.

The UART module is functionally compatible with the TL16C750 UART and is also functionally compatible to earlier designs, such as the TL16C550. The UART module can use hardware or software flow control to manage transmission and reception. Hardware flow control significantly reduces software overhead and increases system efficiency by automatically controlling serial data flow using the RTS output and CTS input signals. Software flow control automatically controls data flow by using programmable XON/XOFF characters.

The UART modem module is enhanced with an autobauding functionality which in control mode allows to automatically set the speed, the number of bit per character, the parity selected.

Figure 21-14. UART Data Format

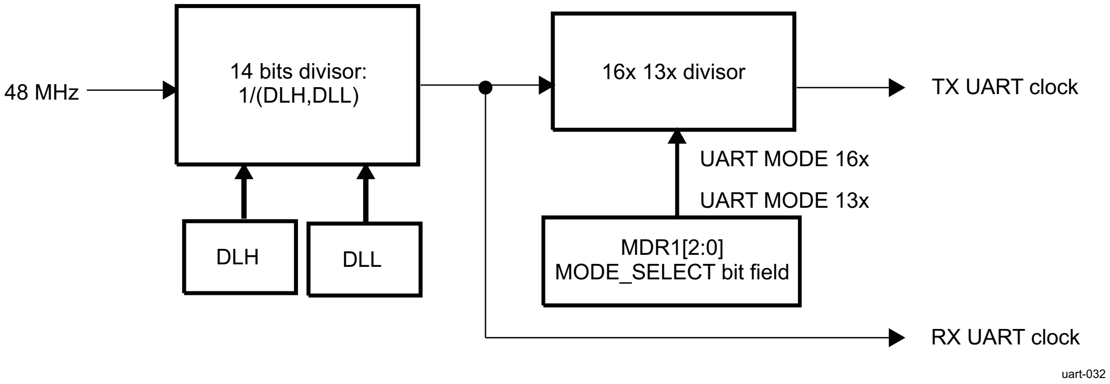


21.3.8.1.1 *UART Clock Generation: Baud Rate Generation*

The UART function contains a programmable baud generator and a set of fixed dividers that divide the 48-MHz clock input down to the expected baud rate.

Figure 21-15 shows the baud rate generator and associated controls.

Figure 21-15. Baud Rate Generation



CAUTION

Before initializing or modifying clock parameter controls (UARTi.UART_DLH, UARTi.UART_DLL), MODE_SELECT = DISABLE (UARTi.UART_MDR1[2:0]) must be set to 0x7. Failure to observe this rule can result in unpredictable module behavior.

21.3.8.1.2 Choosing the Appropriate Divisor Value

Two divisor values are:

- UART 16x mode: Divisor value = Operating frequency/(16x baud rate)
- UART 13x mode: Divisor value = Operating frequency/(13x baud rate)

Table 21-25 describes the UART baud rate settings.

Table 21-25. UART Baud Rate Settings (48-MHz Clock)

Baud Rate	Baud Multiple	DLH,DLL (Decimal)	DLH,DLL (Hex)	Actual Baud Rate	Error (%)
0.3 kbps	16x	10000	0x27, 0x10	0.3 kbps	0
0.6 kbps	16x	5000	0x13, 0x88	0.6 kbps	0
1.2 kbps	16x	2500	0x09, 0xC4	1.2 kbps	0
2.4 kbps	16x	1250	0x04, 0xE2	2.4 kbps	0
4.8 kbps	16x	625	0x02, 0x71	4.8 kbps	0
9.6 kbps	16x	312	0x01, 0x38	9.6153 kbps	+0.16
14.4 kbps	16x	208	0x00, 0xD0	14.423 kbps	+0.16
19.2 kbps	16x	156	0x00, 0x9C	19.231 kbps	+0.16
28.8 kbps	16x	104	0x00, 0x68	28.846 kbps	+0.16
38.4 kbps	16x	78	0x00, 0x4E	38.462 kbps	+0.16
57.6 kbps	16x	52	0x00, 0x34	57.692 kbps	+0.16
115.2 kbps	16x	26	0x00, 0x1A	115.38 kbps	+0.16
230.4 kbps	16x	13	0x00, 0x0D	230.77 kbps	+0.16
460.8 kbps	13x	8	0x00, 0x08	461.54 kbps	+0.16
921.6 kbps	13x	4	0x00, 0x04	923.08 kbps	+0.16
1.843 Mbps	13x	2	0x00, 0x02	1.846 Mbps	+0.16
3.6884 Mbps	13x	1	0x00, 0x01	3.6923 Mbps	+0.16

21.3.8.1.3 UART Data Formatting

The UART can use hardware flow control to manage transmission and reception. Hardware flow control significantly reduces software overhead and increases system efficiency by automatically controlling serial data flow using the RTS output and CTS input signals.

The UART is enhanced with the autobauding function. In control mode, autobauding lets the speed, the number of bits per character, and the parity selected be set automatically.

21.3.8.1.3.1 Frame Formatting

When autobauding is not used, frame format attributes must be defined in the UARTi.UART_LCR register.

Character length is specified using the UARTi.UART_LCR[1:0] CHAR_LENGTH bit field.

The number of stop-bits is specified using the UARTi.UART_LCR[2] NB_STOP bit.

The parity bit is programmed using the UARTi.UART_LCR[5:3] PARITY_EN, PARITY_TYPE_1, and PARITY_TYPE_2 bit fields (see Table 21-26).

Table 21-26. UART Parity Bit Encoding

PARITY_EN	PARITY_TYPE_1	PARITY_TYPE_2	Parity
0	N/A	N/A	No parity
1	0	0	Odd parity
1	1	0	Even parity
1	0	1	Forced 1
1	1	1	Forced 0

21.3.8.1.3.2 Hardware Flow Control

Hardware flow control is composed of auto-CTS and auto-RTS. Auto-CTS and auto-RTS can be enabled and disabled independently by programming the UARTi.UART_EFR[7:6] AUTO_CTS_EN and AUTO_RTS_EN bit fields, respectively.

With auto-CTS, uarti_cts must be active before the module can transmit data.

Auto-RTS activates the uarti_rts output only when there is enough room in the RX FIFO to receive data. It deactivates the uarti_rts output when the RX FIFO is sufficiently full. The HALT and RESTORE trigger levels in the UARTi.UART_TCR register determine the levels at which uarti_rts is activated and deactivated.

If auto-CTS and auto-RTS are enabled, data transmission does not occur unless the RX FIFO has empty space. Thus, overrun errors are eliminated during hardware flow control. If auto-CTS and auto-RTS are not enabled, overrun errors occur if the transmit data rate exceeds the RX FIFO latency.

- Auto-RTS:

  Auto-RTS data flow control originates in the receiver block. The RX FIFO trigger levels used in auto-RTS are stored in the UARTi.UART_TCR register. uarti_rts is active if the RX FIFO level is below the HALT trigger level in the UARTi.UART_TCR[3:0] RX_FIFO_TRIG_HALT bit field. When the RX FIFO HALT trigger level is reached, uarti_rts is deasserted. The sending device (for example, another UART) can send an additional byte after the trigger level is reached because it may not recognize the deassertion of RTS until it begins sending the additional byte.

  uarti_rts is automatically reasserted when the RX FIFO reaches the RESUME trigger level programmed by the UARTi.UART_TCR[7:4] RX_FIFO_TRIG_START bit field. This reassertion requests the sending device to resume transmission.

  In this case, uarti_rts is an active-low signal.

- Auto-CTS:

  The transmitter circuitry checks uarti_cts before sending the next data byte. When uarti_cts is active, the transmitter sends the next byte. To stop the transmitter from sending the next byte, uarti_cts must be deasserted before the middle of the last stop-bit currently sent.

  The auto-CTS function reduces interrupts to the host system. When auto-CTS flow control is enabled, the uarti_cts state changes do not have to trigger host interrupts because the device automatically controls its own transmitter. Without auto-CTS, the transmitter sends any data present in the transmit FIFO, and a receiver overrun error can result.

  In this case, uarti_cts is an active-low signal.

21.3.8.1.3.3 Software Flow Control

Software flow control is enabled through the enhanced feature register (UARTi.UART_EFR) and the modem control register (UARTi.UART_MCR). Different combinations of software flow control can be enabled by setting different combinations of the UARTi.UART_EFR[3:0] bit field (see Table 21-27).

Two other enhanced features relate to software flow control:

- XON any function (UARTi.UART_MCR[5]): Operation resumes after receiving any character after the XOFF character is recognized. If special character detect is enabled and special character is received after XOFF1, it does not resume transmission. The special character is stored in the RX FIFO.

NOTE: The XON-any character is written into the RX FIFO even if it is a software flow character.

- Special character (UARTi.UART_EFR[5]): Incoming data is compared to XOFF2. When the special character is detected, the XOFF interrupt (UARTi.UART_IIR[4]) is set, but it does not halt transmission. The XOFF interrupt is cleared by a read of UARTi.UART_IIR. The special character is transferred to the RX FIFO. Special character does not work with XON2, XOFF2, or sequential XOFFs.

Table 21-27. UART_EFR[3:0] Software Flow Control Options

Bit 3	Bit 2	Bit 1	Bit 0	TX, RX Software Flow Controls
0	0	X	X	No transmit flow control
1	0	X	X	Transmit XON1, XOFF1
0	1	X	X	Transmit XON2, XOFF2
1	1	X	X	Transmit XON1, XON2: XOFF1, XOFF2(1)
X	X	0	0	No receive flow control
X	X	1	0	Receiver compares XON1, XOFF1
X	X	0	1	Receiver compares XON2, XOFF2
X	X	1	1	Receiver compares XON1, XON2: XOFF1, XOFF2(1)

(1) In these cases, the XON1 and XON2 characters or the XOFF1 and XOFF2 characters must be transmitted/received sequentially with XON1/XOFF1 followed by XON2/XOFF2.
XON1 is defined in the UARTi.UART_XON1_ADDR1[7:0] XON_WORD1 bit field. XON2 is defined in the UARTi.UART_XON2_ADDR2[7:0] XON_WORD2 bit field.
XOFF1 is defined in the UARTi.UART_XOFF1[7:0] XOFF_WORD1 bit field. XOFF2 is defined in the UARTi.UART_XOFF2[7:0] XOFF_WORD2 bit field.

21.3.8.1.3.3.1 Receive (RX)

When software flow control operation is enabled, the UART compares incoming data with XOFF1/2 programmed characters (in certain cases, XOFF1 and XOFF2 must be received sequentially). When the correct XOFF characters are received, transmission stops after transmission of the current character completes. Detection of XOFF also sets the UARTi.UART_IIR[4] bit (if enabled by UARTi.UART_IER[5]) and causes the interrupt line to go low.

To resume transmission, an XON1/2 character must be received (in certain cases, XON1 and XON2 must be received sequentially). When the correct XON characters are received, the UARTi.UART_IIR[4] bit is cleared and the XOFF interrupt disappears.

NOTE: When a parity, framing, or break error occurs while receiving a software flow control character, this character is treated as normal data and is written to the RX FIFO.

When XON-any and special character detect are disabled and software flow control is enabled, no valid XON or XOFF characters are written to the RX FIFO. For example, when UARTi.UART_EFR[1:0] = 0x2, if XON1 and XOFF1 characters are received, they are not written to the RX FIFO.

When pairs of software flow characters are programmed to be received sequentially (UARTi.UART_EFR[1:0] = 0x3), the software flow characters are not written to the RX FIFO if they are received sequentially. However, received XON1/XOFF1 characters must be written to the RX FIFO if the subsequent character is not XON2/XOFF2.

21.3.8.1.3.3.2 Transmit (TX)

Two XOFF1 characters are transmitted when the RX FIFO passes the trigger level programmed by UARTi.UART_TCR[3:0]. As soon as the RX FIFO reaches the trigger level programmed by UARTi.UART_TCR[7:4], two XON1 characters are sent, so the data transfer recovers.

NOTE: If software flow control is disabled after an XOFF character is sent, the module transmits XON characters automatically to enable normal transmission.

The transmission of XOFF(s)/XON(s) follows the same protocol as transmission of an ordinary byte from the TX FIFO. This means that even if the word length is 5, 6, or 7 characters, the 5, 6, or 7 LSBs of XOFF1/2 and XON1/2 are transmitted. The 5, 6, or 7 bits of a character are seldom transmitted, but this function is included to maintain compatibility with earlier designs.

It is assumed that software flow control and hardware flow control are never enabled simultaneously.

21.3.8.1.3.4 Autobauding Modes

In autobauding mode, the UART can extract transfer characteristics (speed, length, and parity) from an "at" (AT) command (ASCII code). These characteristics are used to receive data after an AT and to send data.

The following AT commands are valid:

AT	DATA	<CR>
at	DATA	<CR>
A/		
a/		

A line break during the acquisition of the sequence AT is not recognized, and an echo function is not implemented in hardware.

A/ and a/ are not used to extract characteristics, but they must be recognized because of their special meaning. A/ or a/ is used to instruct the software to repeat the last received AT command; therefore, an a/ always follows an AT, and transfer characteristics are not expected to change between an AT and an a/.

When a valid AT is received, AT and all subsequent data, including the final <CR> (0x0D), are saved to the RX FIFO. The autobaud state-machine waits for the next valid AT command. If an a/ (A/) is received, the a/ (A/) is saved in the RX FIFO and the state-machine waits for the next valid AT command.

On the first successful detection of the baud rate, the UART activates an interrupt to signify that the AT (upper or lower case) sequence is detected. The UARTi.UART_UASR register reflects the correct settings for the baud rate detected. Interrupt activity can continue in this fashion when a subsequent character is received. Therefore, it is recommended that the software enable the RHR interrupt when using the autobaud mode.

The following settings are detected in autobaud mode with a module clock of 48 MHz:

- Speed:
  - 115.2K baud
  - 57.6K baud
  - 38.4K baud
  - 28.8K baud
  - 19.2K baud
  - 14.4K baud
  - 9.6K baud
  - 4.8K baud
  - 2.4K baud
  - 1.2K baud
- Length: 7 or 8 bits
- Parity: Odd, even, or space

NOTE: The combination of 7-bit character plus space parity is not supported.

Autobauding mode is selected when the UARTi.UART_MDR1[2:0] MODE_SELECT bit field is set to 0x2. In UART autobauding mode, UARTi.UART_DLL, UARTi.UART_DLH, and UARTi.UART_LCR[5:0] bit field settings are not used; instead, UART_UASR is updated with the configuration detected by the autobauding logic.

UART_UASR Autobauding Status Register Use

This register is used to set up transmission according to the characteristics of the previous reception instead of the UARTi.UART_LCR, UARTi.UART_DLL, and UARTi.UART_DLH registers when the UART is in autobauding mode.

To reset the autobauding hardware (to start a new AT detection) or to set the UART in standard mode (no autobaud), the UARTi.UART_MDR1[2:0] MODE_ SELECT bit field must be set to reset state (0x7) and then to the UART in autobauding mode (0x2) or to the UART in standard mode (0x0).

Use limitation:

- Only 7- and 8-bit characters (5- and 6-bit not supported)
- 7-bit character with space parity not supported
- Baud rate between 1200 and 115,200 bps (10 possibilities)

21.3.8.1.3.5 Error Detection

When the UARTi.UART_LSR register is read, the UARTi.UART_LSR[4:2] bit field reflects the error bits (BI: break condition, FE: framing error, PE: parity error) of the character at the top of the RX FIFO (the next character to be read). Therefore, reading the UARTi.UART_LSR register and then reading the UARTi.UART_RHR register identifies errors in a character.

Reading the UARTi.UART_RHR register updates the BI, FE, and PE bits (see Table 21-11 for the UART mode interrupts).

The UARTi.UART_LSR[7] RX_FIFO_STS bit is set when there is an error in the RX FIFO and is cleared only when no errors remain in the RX FIFO.

NOTE: Reading the UARTi.UART_LSR register does not cause an increment of the RX FIFO read pointer. The RX FIFO read pointer is incremented by reading the UARTi.UART_RHR register.

Reading the UARTi.UART_LSR register clears the OE bit if it is set (see Table 21-11 for the UART mode interrupts).

21.3.8.1.3.6 Overrun During Receive

Overrun during receive occurs if the RX state-machine tries to write data into the RX FIFO when it is already full. When overrun occurs, the device interrupts the MPU with the UARTi.UART_IIR[5:1] IT_TYPE bit field set to 0x3 (receiver line status error) and discards the remaining portion of the frame.

Overrun also causes an internal flag to be set, which disables further reception. Before the next frame can be received, the MPU must:

- Reset the RX FIFO.
- Read the UARTi.UART_RESUME register, which clears the internal flag.

21.3.8.1.3.7 Time-Out and Break Conditions

21.3.8.1.3.7.1 Time-Out Counter

An RX idle condition is detected when the receiver line (uarti_rx) is high for a time that equals 4x the programmed word length + 12 bits. uarti_rx is sampled midway through each bit.

For sleep mode, the counter is reset when there is activity on uarti_rx.

For the time-out interrupt, the counter counts only when there is data in the RX FIFO, and the count is reset when there is activity on uarti_rx or when the UARTi.UART_RHR register is read.

21.3.8.1.3.7.2 Break Condition

When a break condition occurs, uarti_tx is pulled low. A break condition is activated by setting the UARTi.UART_LCR[6] BREAK_EN bit. The break condition is not aligned on word stream (a break condition can occur in the middle of a character). The only way to send a break condition on a full character is:

1. Reset the TX FIFO (if enabled).
2. Wait for the transmit shift register to empty (the UARTi.UART_LSR[6] TX_SR_E bit is set to 1).

3. Take a guard time according to stop-bit definition.
4. Set the BREAK_EN bit to 1.

The break condition is asserted while the BREAK_EN bit is set to 1.

The time-out counter and break condition apply only to UART modem operation and not to IrDA/CIR mode operation.

21.3.8.2 IrDA Mode

21.3.8.2.1 Slow Infrared (SIR) Mode

In SIR mode, data transfers take place between the LH and peripheral devices at speeds of up to 115200 bauds. A SIR transmit frame starts with start flags (either a single 0xC0, multiple 0xC0, or a single 0xC0 preceded by a number of 0xFF flags), followed by frame data, CRC-16, and ends with a stop flag (0xC1). The bit format for a single word uses a single start bit, eight data bits, and one stop bit. The format is unaffected by the use and settings of the LCR register.

Note that BLR[6] is used to select whether to use 0xC0 or 0xFF start patterns when multiple start flags are required.

The SIR transmit state machine attaches start flags, CRC-16, and stop flags. It checks the outgoing data to establish if data transparency is required.

SIR transparency is carried out if the outgoing data, between the start and stop flags, contains 0xC0, 0xC1, or 0x7D. If one of these is about to be transmitted, then the SIR state machine sends an escape character (0x7D) first, then inverts the fifth bit of the real data to be sent and sends this data immediately after the 0x7D character.

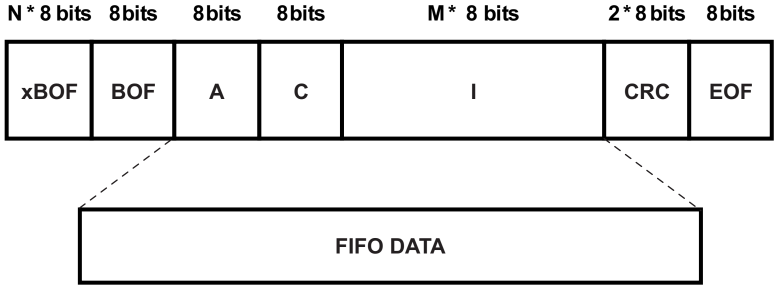
The SIR receive state machine recovers the receive clock, removes the start flags, removes any transparency from the incoming data, and determines frame boundary with reception of the stop flag. It also checks for errors, such as frame abort (0x7D character followed immediately by a 0xC1 stop flag, without transparency), CRC error, and frame-length error. At the end of a frame reception, the LH reads the line status register (LSR) to find out possible errors of the received frame.

Data can be transferred both ways by the module, but when the device is transmitting the IR RX circuitry is automatically disabled by hardware. See bit 5 in the auxiliary control register (ACREG) for a description of the logical operation. **Note:** This applies to all three modes SIR, MIR, and FIR.

The infrared output in SIR mode can either be 1.6µs or 3/16 encoding, selected by the PULSETYPE bit of the Auxiliary Control Register (ACREG[7]). In 1.6µs encoding, the infrared pulse width is 1.6µs and in 3/16 encoding the infrared pulse width is 3/16 of a bit duration (1/baud-rate). The receiver supports both 3/16 and 1.6µs pulse duration by default. The transmitting device must send at least two start flags at the start of each frame for back-to-back frames. **Note:** Reception supports variable-length stop bits.

21.3.8.2.1.1 Frame Format

Figure 21-16. IrDA SIR Frame Format



The CRC is applied on the address (A), control (C) and information (I) bytes.

Note: The two words of CRC are written in the FIFO in reception.

21.3.8.2.1.2 Asynchronous Transparency

Before transmitting a byte, the UART IrDA controller examines each byte of the payload and the CRC field (between BOF and EOF). For each byte equal to 0xC0 (BOF), 0xC1 (EOF), or 0x7D (control escape) it does the following.

In transmission

1. Inserts a control escape (CE) byte preceding the byte.
2. Complements bit 5 of the byte (i.e., exclusive OR's the byte with 0x20).

The byte sent for the CRC computation is the initial byte written in the TX FIFO (before the XOR with 0x20).

In reception

For the A, C, I, CRC field:

1. Compare the byte with CE byte, and if not equal send it to the CRC detector and store it in the RX FIFO.
2. If equal to CE, discard the CE byte.
3. Complements the bit 5 of the byte following the CE.
4. 4. Send the complemented byte to the CRC detector and store it in the RX FIFO.

21.3.8.2.1.3 Abort Sequence

The transmitter may decide to prematurely close a frame. The transmitter aborts by sending the following sequence: 0x7DC1. The abort pattern closes the frame without a CRC field or an ending flag.

It is possible to abort a transmission frame by programming the ABORTEN bit of the Auxiliary Control Register (ACREG[1]). When this bit is set to 1, 0x7D and 0xC1 are transmitted and the frame is not terminated with CRC or stop flags. The receiver treats a frame as an aborted frame when a 0x7D character, followed immediately by a 0xC1 character, has been received without transparency.

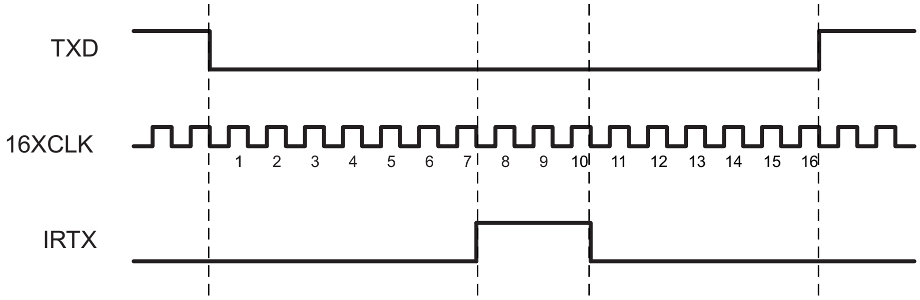
21.3.8.2.1.4 Pulse Shaping

In SIR mode, both the 3/16th and the 1.6μs pulse duration methods are supported in receive and transmit. ACREG[7] selects the pulse width method in transmit mode.

21.3.8.2.1.5 Encoder

Serial data from the transmit state machine is encoded to transmit data to the optoelectronics. While the serial data input to the (TXD) is high, the output (IRTX) is always low, and the counter used to form a pulse on IRTX is continuously cleared. After TXD resets to 0, IRTX rises on the falling edge of the 7th 16XCLK. On the falling edge of the 10th 16XCLK pulse, IRTX falls, creating a 3-clock-wide pulse. While TXD stays low, a pulse is transmitted during the 7th to the 10th clock of each 16-clock bit cycle.

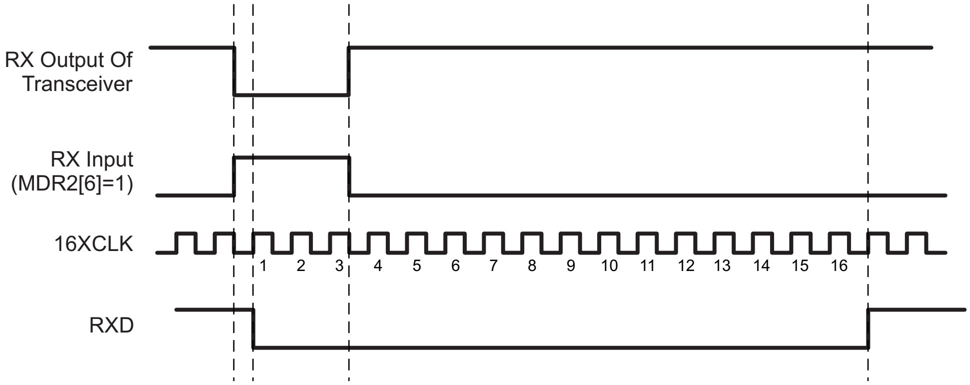
Figure 21-17. IrDA Encoding Mechanism



21.3.8.2.1.6 Decoder

After reset, RXD is high and the 4-bit counter is cleared. When a rising edge is detected on RX, RXD falls on the next rising edge of 16XCLK with sufficient setup time. RXD stays low for 16 cycles (16XCLK) and then returns to high as required by the IrDA specification. As long as no pulses (rising edges) are detected on the RX, RXD remains high.

Figure 21-18. IrDA Decoding Mechanism



The operation of the RX input can be disabled with DISIRRX bit of the Auxiliary Control Register (ACREG[5]). Furthermore, the MDR2[6] can be used to invert the signal from the transceiver (RX ouput) pin to the IRRX logic inside the UART.

21.3.8.2.1.7 IR Address Checking

In all IR modes, if address checking has been enabled, only frames intended for the device are written to the RX FIFO. This is to avoid receiving frames not meant for this device in a multi-point infrared environment. It is possible to program two frame addresses that the UART IrDA receives with XON1/ADDR1 and XON2/ADDR2 registers. Selecting address1 checking is done by setting EFR[0] to 1; address2 checking is done by setting EFR[1] to 1.

Setting EFR[1:0] to 0 disables all address checking operations. If both bits are set, then the incoming frame is checked for both private and public addresses. If address checking is disabled, then all received frames are written into the reception FIFO.

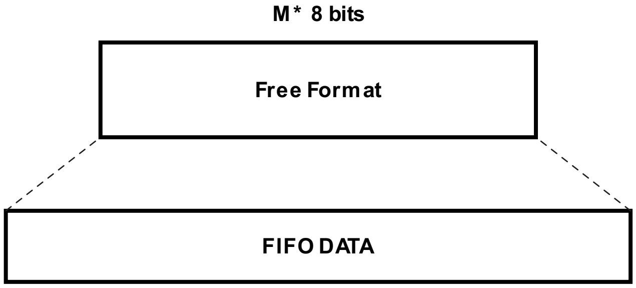
21.3.8.2.1.8 SIR Free Format Mode

To allow complete software flexibility in the transmission and reception of Infrared data packets, the SIR free format mode is a sub-function of the existing SIR mode such that all frames going to and from the FIFO buffers are untouched with respect to appending and removing control characters and CRC values. In the transmission phase, the mode uses the IRTX pin, as in SIR mode.

This mode corresponds to a UART mode with a pulse modulation of 3/16 of baud-rate pulse width.

For example, a normal SIR packet has BOF control and CRC error checking data appended (transmitting) or removed (receiving) from the data going to and from the FIFOs. In SIR free format mode, only the data termed the FIFO DATA area, illustrated in Figure 21-19, would be transmitted and received.

Figure 21-19. SIR Free Format Mode



In this mode, the entire FIFO data packet is to be constructed (encoded and decoded) by the LH software.

The SIR free format mode is selected by setting the module in UART mode (MDR1[2:0] = 000) and the MDR2[3] register bit to one to allow the pulse shaping. As the bit format is to remain the same, some UART mode configuration registers need to be set at specific value:

- LCR[1:0] = “11” (8 data bits)
- LCR[2] = 0 (1 stop bit)
- LCR[3] = 0 (no parity)
- ACREG[7] = 0 (3/16 of baud-rate pulse width)

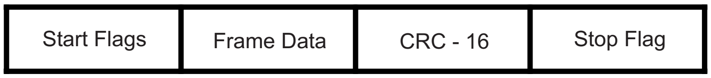
The features defined through MDR2[6] and ACREG[5] are also supported.

Note: - All other configuration registers need to be at the reset value. The UART mode interrupts are used for the SIR free format mode, but many of them are not relevant (e.g., XOFF, RTS, CTS, Modem status register).

21.3.8.2.2 Medium Infrared (MIR) Mode

In MIR mode, data transfers take place between LH and peripheral devices at 0.576 or 1.152 Mbits/s speed. A MIR transmit frame starts with start flags (at least two), followed by a frame data, CRC-16, and ends with a stop flag.

Figure 21-20. MIR Transmit Frame Format



On transmit, the MIR state machine attaches start flags, CRC-16, and stop flags. It also looks for five consecutive values of 1 in the frame data and automatically inserts a zero after five consecutive values of one (this is called bit stuffing).

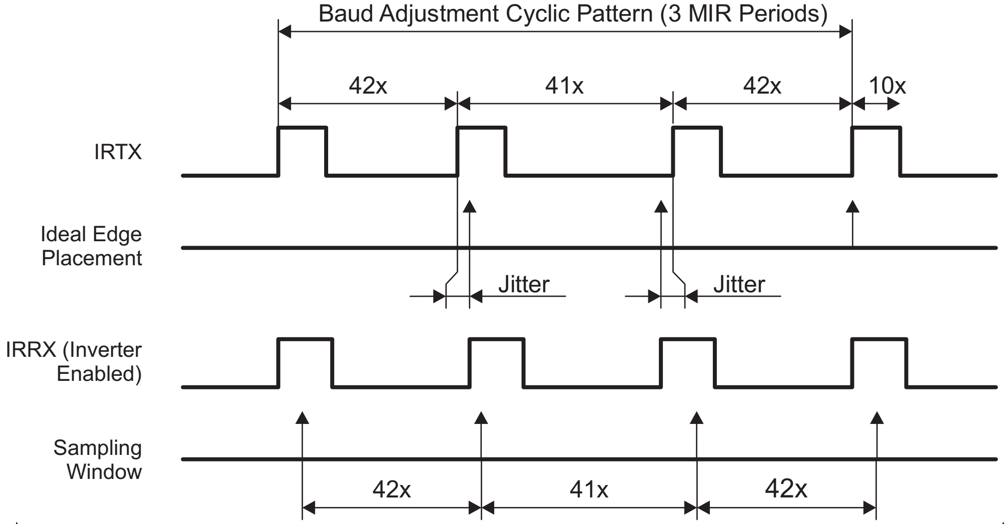
On receive, the MIR receive state machine recovers the receive clock, removes the start flags, de-stuffs the incoming data, and determines frame boundary with reception of the stop flag. It also checks for errors, such as frame abort, CRC error, or frame-length error. At the end of a frame reception, the LH reads the line status register (LSR) to find possible errors of received frame.

Data can be transferred both ways by the module but when the device is transmitting, the IR RX circuitry is automatically disabled by hardware. See bit 5 in the auxiliary control register (ACREG) for a description of the logical operation. **Note:** This applies to all three modes SIR, MIR and FIR.

21.3.8.2.2.1 MIR Encoder/Decoder

In order to meet MIR baud-rate tolerance of +/-0.1% with a 48-MHz clock input, a 42-41-42 encoding/decoding adjustment is performed. The reference start point is the first start flag and the 42-41-42 cyclic pattern is repeated until the stop flag is sent or detected. The jitter created this way is within MIR tolerances. The pulse width is not exactly 1/4 but within tolerances defined by the IrDA specifications.

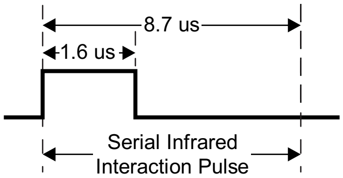
Figure 21-21. MIR BAUD Rate Adjustment Mechanism



21.3.8.2.2.2 SIP Generation

In MIR and FIR operation modes, the transmitter needs to send a serial infrared interaction pulse (SIP) at least once every 500 ms. The purpose of the SIP is to let slow devices (operating in SIR mode) know that the medium is currently occupied. The SIP pulse is shown in Figure 21-22

Figure 21-22. SIP Pulse



When the SIPMODE bit of Mode Definition Register 1 (MDR1[6]) equals 1, the TX state machine will always send one SIP at the end of a transmission frame. But when MDR1[6] = 0, the transmission of the SIP depends on the SENDSIP bit of the Auxiliary Control Register (ACREG[3]). The system (LH) can set ACREG[3] at least once every 500ms. The advantage of this approach over the default approach is that the TX state machine does not need to send the SIP at the end of each frame which may reduce the overhead required

21.3.8.2.3 Fast Infrared (FIR) Mode

In FIR mode, data transfers take place between LH and peripheral devices at 4 Mbits/s speed. A FIR transmit frame starts with a preamble, followed by a start flag, frame data, CRC-32, and ends with a stop flag.

Figure 21-23. FIR Transmit Frame Format

On transmit, the FIR transmit state machine attaches the preamble, start flag, CRC-32, and stop flag. It also encodes the transmit data into 4PPM format. It also generates the serial infrared interaction pulse (SIP).

On receive, the FIR receive state machine recovers the receive clock, removes the start flag, decodes the 4PPM incoming data, and determines frame boundary with a reception of the stop flag. It also checks for errors such as an illegal symbol, a CRC error, and a frame-length error. At the end of a frame reception, the LH reads the line status register (LSR) to find out possible errors of the received frame.

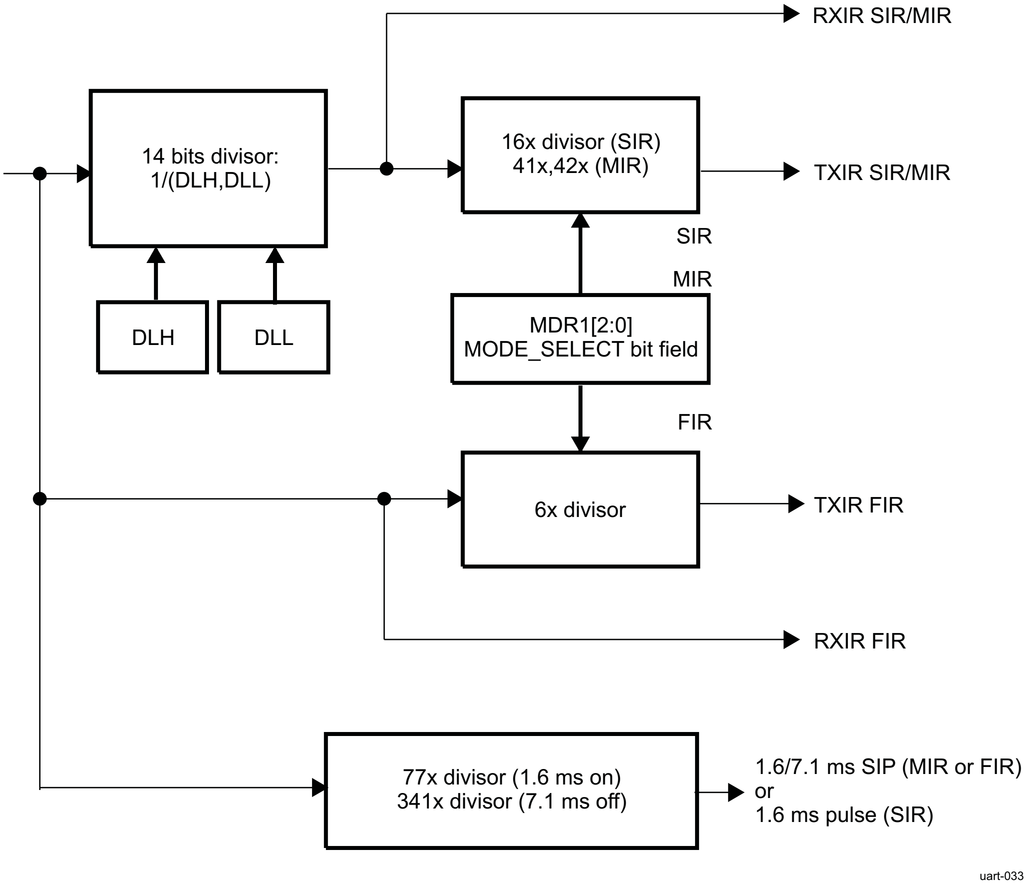
Data can be transferred both ways by the module but when the device is transmitting, the IR RX circuitry is automatically disabled by hardware. See bit 5 in the auxiliary control register (ACREG) for a description of the logical operation. **Note:** This applies to all three modes of SIR, MIR, and FIR.

21.3.8.2.4 IrDA Clock Generation: Baud Generator

The IrDA function contains a programmable baud generator and a set of fixed dividers that divide the 48-MHz clock input down to the expected baud rate.

Figure 21-24 shows the baud rate generator and associated controls.

Figure 21-24. Baud Rate Generator



CAUTION

Before initializing or modifying clock parameter controls (UARTi.UART_DLH, UARTi.UART_DLL), MODE_SELECT=DISABLE (UARTi.UART_MDR1[2:0]) must be set to 0x7). Failure to observe this rule can result in unpredictable module behavior.

21.3.8.2.5 Choosing the Appropriate Divisor Value

Three divisor values are:

- SIR mode: Divisor value = Operating frequency/(16x baud rate)
- MIR mode: Divisor value = Operating frequency/(41x/42x baud rate)
- FIR mode: Divisor value = None

Table 21-28 lists the IrDA baud rate settings.

Table 21-28. IrDA Baud Rate Settings

Baud Rate	IR Mode	Baud Multiple	Encoding	DLH, DLL (Decimal)	Actual Baud Rate	Error (%)	Source Jitter (%)	Pulse Duration
2.4 kbps	SIR	16x	3/16	1250	2.4 kbps	0	0	78.1 µs
9.6 kbps	SIR	16x	3/16	312	9.6153 kbps	+0.16	0	19.5 µs
19.2 kbps	SIR	16x	3/16	156	19.231 kbps	+0.16	0	9.75 µs
38.4 kbps	SIR	16x	3/16	78	38.462 kbps	+0.16	0	4.87 µs
57.6 kbps	SIR	16x	3/16	52	57.692 kbps	+0.16	0	3.25 µs
115.2 kbps	SIR	16x	3/16	26	115.38 kbps	+0.16	0	1.62 µs
0.576 Mbps	MIR	41x/42x	1/4	2	0.5756 Mbps(1)	0	+1.63/–0.80	416 ns
1.152 Mbps	MIR	41x/42x	1/4	1	1.1511 Mbps(1)	0	+1.63/–0.80	208 ns
4 Mbps	FIR	6x	4 PPM	–	4 Mbps	0	0	125 ns

(1) Average value

NOTE: Baud rate error and source jitter table values do not include 48-MHz reference clock error and jitter.

21.3.8.2.6 IrDA Data Formatting

The methods described in this section apply to all IrDA modes (SIR, MIR, and FIR).

21.3.8.2.6.1 IR RX Polarity Control

The UARTi.UART_MDR2[6] IRRXINVERT bit provides the flexibility to invert the uarti_rx_irrx pin in the UART to ensure that the protocol at the output of the transceiver has the same polarity at module level. By default, the uarti_rx_irrx pin is inverted because most transceivers invert the IR receive pin.

21.3.8.2.6.2 IrDA Reception Control

The module can transmit and receive data, but when the device is transmitting, the IR RX circuitry is automatically disabled by hardware.

Operation of the uarti_rx_irrx input can be disabled by the UARTi.UART_ACREG[5] DIS_IR_RX bit.

21.3.8.2.6.3 IR Address Checking

In all IR modes, when address checking is enabled, only frames intended for the device are written to the RX FIFO. This restriction avoids receiving frames not meant for this device in a multipoint infrared environment. It is possible to program two frame addresses that the UART IrDA receives, with the UARTi.UART_XON1_ADDR1[7:0] XON_WORD1 and UARTi.UART_XON2_ADDR2[7:0] XON_WORD2 bit fields.

Setting the UART_EFR[0] bit to 1 selects address1 checking. Setting the UART_EFR[1] bit to 1 selects address2 checking. Setting the UART_EFR[1:0] bit field to 0 disables all address checking operations. If both bits are set, the incoming frame is checked for private and public addresses.

If address checking is disabled, all received frames write to the RX FIFO.

21.3.8.2.6.4 Frame Closing

A transmission frame can be terminated in two ways:

- Frame-length method: Set the UARTi.UART_MDR1[7] FRAME_END_MODE bit to 0. The MPU writes the value of the frame length to the UARTi.UART_TXFLH and UARTi.UART_TXFLL registers. The device automatically attaches end flags to the frame when the number of bytes transmitted equals the value of the frame length.
- Set-EOT bit method: Set the FRAME_END_MODE bit to 1. The MPU writes 1 to the UARTi.UART_ACREG[0] EOT bit just before it writes the last byte to the TX FIFO. When the MPU writes the last byte to the TX FIFO, the device internally sets the tag bit for that character in the TX FIFO. As the TX state-machine reads data from the TX FIFO, it uses this tag-bit information to attach end flags and correctly terminate the frame.

21.3.8.2.6.5 Store and Controlled Transmission

In store and controlled transmission (SCT) mode, the MPU starts writing data to the TX FIFO. Then, after writing a part of a frame (for a bigger frame) or an entire frame (a small frame; that is, a supervisory frame), the MPU writes 1 to the UARTi.UART_ACREG[2] SCTX_EN bit (deferred TX start) to start transmission.

SCT mode is enabled by setting the UARTi.UART_MDR1[5] SCT bit to 1. This transmission method differs from normal mode, in which data transmission starts immediately after data is written to the TX FIFO. SCT mode is useful for sending short frames without TX underrun.

21.3.8.2.6.6 Error Detection

When the UARTi.UART_LSR register is read, the UARTi.UART_LSR[4:2] bit field reflects the error bits [FL, CRC, ABORT] of the frame at the top of the STATUS FIFO (the next frame status to be read).

The error is triggered by an interrupt (for IrDA mode interrupts, see Table 21-12). The STATUS FIFO must be read until empty (a maximum of eight reads is required).

21.3.8.2.6.7 Underrun During Transmission

Underrun during transmission occurs when the TX FIFO is empty before the end of the frame is transmitted. When underrun occurs, the device closes the frame with end flags but attaches an incorrect CRC value. The receiving device detects a CRC error and discards the frame; it can then ask for a retransmission.

Underrun also causes an internal flag to be set, which disables additional transmissions. Before the next frame can be transmitted, the MPU must:

- Reset the TX FIFO.
- Read the UARTi.UART_RESUME register, which clears the internal flag.

This function can be disabled by the UARTi.UART_ACREG[4] DIS_TX_UNDERRUN bit, compensated by the extension of the stop-bit in transmission if the TX FIFO is empty.

21.3.8.2.6.8 Overrun During Receive

Overrun during receive for the IrDA mode has the same function as that for the UART mode (see Section 21.3.8.1.3.6, *Overrun During Receive*).

21.3.8.2.6.9 Status FIFO

In IrDA modes, a status FIFO records the received frame status. When a complete frame is received, the length of the frame and the error bits associated with the frame are written to the status FIFO.

Reading the UARTi.UART_SFREGH[3:0] MSB and UARTi.UART_SFREGL[3:0] (LSB) bit fields obtains the frame length. The frame error status is read in the UARTi.UART_SFLSR register. Reading the UARTi.UART_SFLSR register increments the status FIFO read pointer. Because the status FIFO is eight entries deep, it can hold the status of eight frames.

The MPU uses the frame-length information to locate the frame boundary in the received frame data. The MPU can screen bad frames using the error status information and can later request the sender to resend only the bad frames.

This status FIFO can be used effectively in DMA mode because the MPU must be interrupted only when the programmed status FIFO trigger level is reached, not each time a frame is received.

21.3.8.2.7 SIR Mode Data Formatting

This section provides specific instructions for SIR mode programming.

21.3.8.2.7.1 Abort Sequence

The transmitter can prematurely close a frame (abort) by sending the sequence 0x7DC1. The abort pattern closes the frame without a CRC field or an ending flag.

A transmission frame can be aborted by setting the UARTi.UART_ACREG[1] ABORT_EN bit to 1. When this bit is set to 1, 0x7D and 0xC1 are transmitted and the frame is not terminated with CRC or stop flags.

When a 0x7D character followed immediately by a 0xC1 character is received without transparency, the receiver treats a frame as an aborted frame.

CAUTION

When the TX FIFO is not empty and the UARTi.UART_MDR1[5] SCT bit is set to 1, the UART IrDA starts a new transfer with data of a previous frame when the aborted frame is sent. Therefore, the TX FIFO must be reset before sending an aborted frame.

21.3.8.2.7.2 Pulse Shaping

SIR mode supports the 3/16 or the 1.6-µs pulse duration methods in receive and transmit. The UARTi.UART_ACREG[7] PULSE_TYPE bit selects the pulse width method in the transmit mode.

21.3.8.2.7.3 SIR Free Format Programming

The SIR FF mode is selected by setting the module in the UART mode (UARTi.UART_MDR1[2:0] MODE_SELECT = 0x0) and the UARTi.UART_MDR2[3] UART_PULSE bit to 1 to allow pulse shaping.

Because the bit format stays the same, some UART mode configuration registers must be set at specific values:

- UARTi.UART_LCR[1:0] CHAR_LENGTH bit field = 0x3 (8 data bits)
- UARTi.UART_LCR[2] NB_STOP bit = 0x0 (1 stop-bit)
- UARTi.UART_LCR[3] PARITY_EN bit = 0x0 (no parity)

The UART mode interrupts are used for the SIR FF mode, but many are not relevant (XOFF, RTS, CTS, modem status register, etc.).

21.3.8.2.8 MIR and FIR Mode Data Formatting

This section describes common instructions for FIR and MIR mode programming.

At the end of a frame reception, the MPU reads the line status register (UARTi.UART_LSR) to detect errors in the received frame.

When the UARTi.UART_MDR1[6] SIP_MODE bit is set to 1, the TX state-machine always sends one SIP at the end of a transmission frame. However, when the SIP_MODE bit is set to 0, SIP transmission depends on the UARTi.UART_ACREG[3] SEND_SIP bit.

The MPU can set the SEND_SIP bit at least once every 500 ms. The advantage of this approach over the default approach is that the TX state-machine does not have to send the SIP at the end of each frame, thus reducing the overhead required.

21.3.8.3 CIR Mode

In consumer infrared mode, the infrared operation is designed to function as a programmable (universal) remote control. By setting the MDR1 register, the UART can be set to CIR mode in the same way as the other IrDA modes are set using the MDR1 register.

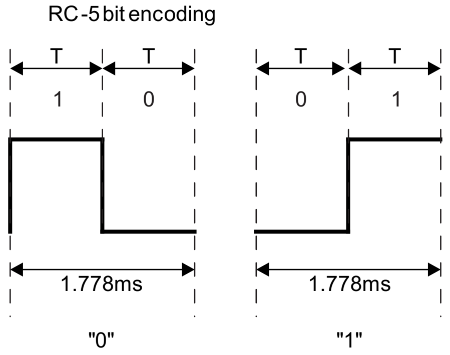
The CIR mode uses a variable pulse width modulation technique (based on multiples of a programmable T period) to encompass the various formats of infrared encoding for remote control applications. The CIR logic is to transmit and receive data packets according to the user definable frame structure and packet content.

21.3.8.3.1 Consumer IR Encoding

There are two distinct methods of encoding for remote control applications. The first uses time extended bit forms i.e. a variable pulse distance (or duration) whereby the difference between a logic one and logic zero is the length of the pulse width; and the second is the use of a bi-phase where the encoding of the logic zero and one is in the change of signal level from 1→0 or 0→1 respectively. Japanese manufacturers tend to favor the use of pulse duration encoding whereas European manufacturers favor the use of bi-phase encoding.

The CIR mode is designed to use a completely flexible free format encoding where a digit ‘1’ from the TX/RX FIFO is to be transmitted/received as a modulated pulse with duration T. Equally, a ‘0’ is to be transmitted/received as a blank duration T. The protocol of the data is to be constructed and deciphered by the host CPU. For example, the RC-5 protocol using Manchester encoding can be emulated as using a “01” pair for one and “10” pair for a zero.

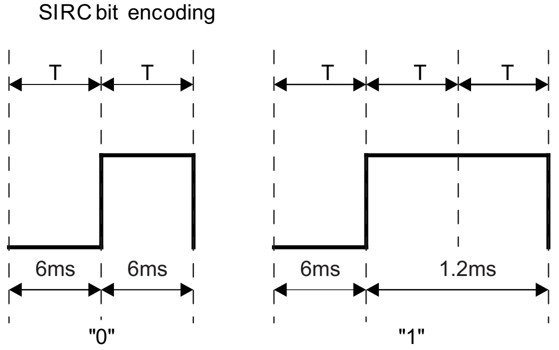
Figure 21-25. RC-5 Bit Encoding



Since the CIR mode logic does not impose a fixed format for infrared packets of data, the CPU software is at liberty to define the format through the use of simple data structures that will then be modulated into an industry standard, such as RC5 or SIRC. To send a sequence of “0101” in RC5, the host software must write an eight bit binary character of “10011001” to the data TX FIFO of the UART.

For SIRC, the modulation length (i.e. multiples of T) is the method to distinguish between a “1” or a “0”. The following SIRC digits show the difference in encoding between this and RC5 for example. Note: the pulse width is extended for “1” digits.

Figure 21-26. SIRC Bit Encoding



To construct comprehensive packets that constitute remote control commands, the host software must combine a number of eight bit data characters in a sequence that follows one of the universally accepted formats. For illustrative purposes, a standard RC5 frame is described below (the SIRC format follows this). Each of the above fields in RC-5 can be considered as two T pulses (digital bits) from the TX and RX FIFOs.

The standard RC5 format as seen by the UART_IrDA in CIR mode.

Figure 21-27. RC-5 Standard Packet Format

S1	S2	T	A4	A3	A2	A1	A0	C5	C4	C3	C2	C1	C0

Where:

- S1, S2: Start bits (always 1)
- T: Toggle bit
- A4–A0: Address (or system) bits
- C5–C0: Command bits

The toggle bit T changes each time a new command is transmitted to allow detection of pressing the same key twice (or effectively receiving the same data from the host consecutively). Since a code is being sent as long as the CPU transmits characters to the UART for transmission, a brief delay in the transmission of the same command would be detected by the use of the toggle bit. The address bits define the machine or device that the Infrared transmission is intended for and the command defines the operation.

To accommodate an extended RC5 format, the S2 bit is replaced by a further command bit (C6) that allows the command range to increase to 7-bits. This format is known as the extended RC-5 format.

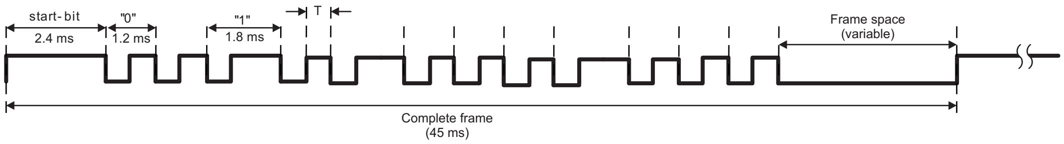
The SIRC encoding uses the duration of modulation for mark and space; hence the duration of data bits inside the standard frame length will vary depending upon the logic 1 content. The packet format and bit encoding is illustrated below. There is one start bit of two milliseconds and control codes followed by data that constitute the whole frame.

Figure 21-28. SIRC Packet Format

S	C0	C1	C2	C3	C4	C5	C6	D0	D1	D2	D3	D4

It should be noted that the encoding must take a standard duration but the contents of the data may vary. This implies that the control software for emitting and receiving data packets must exercise a scheme of inter-packet delay, where the emission of successive packets can only be done after a real time delay has expired.

Figure 21-29. SIRC Bit Transmission Example

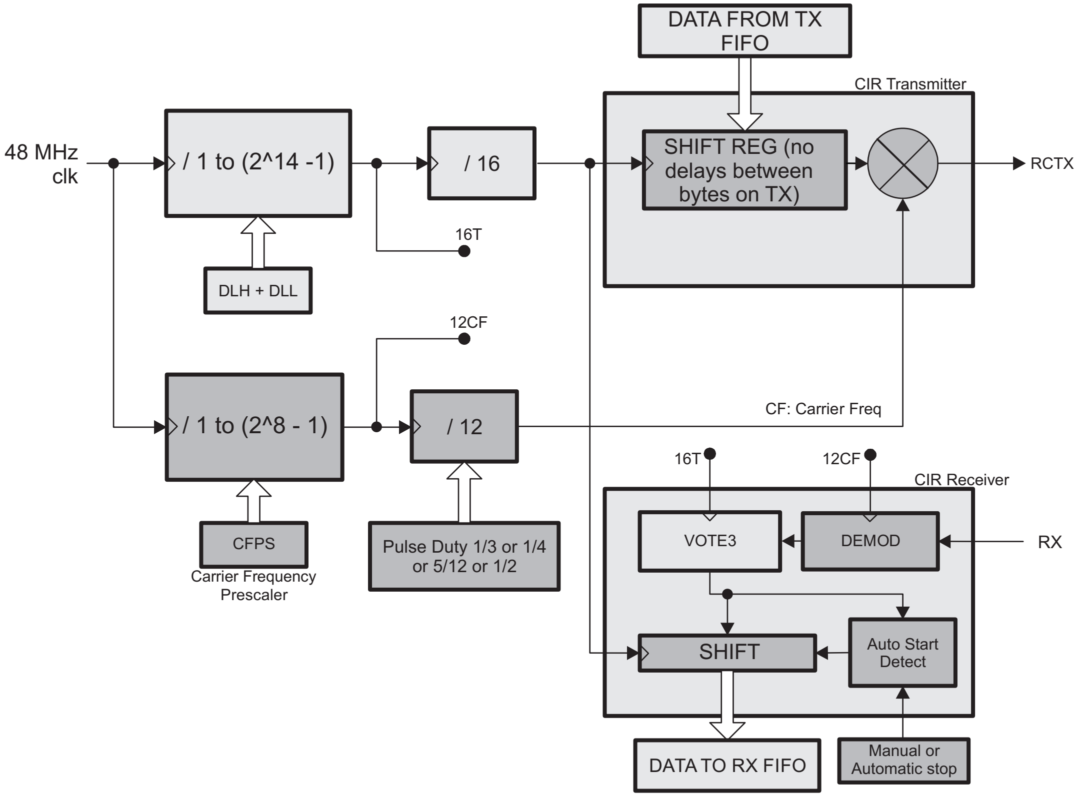


It is beyond the scope of this document to describe all encoding methods and techniques, the previous information was provided to illustrate the consideration required to employ different encoding methods for different industry standard protocols. The user should refer to industry standard documentation for specific methods of encoding and protocol usage.

21.3.8.3.2 CIR Mode Operation

Depending on the encoding method (variable pulse distance / bi-phase), the LH should develop a data structure that combines the 1 and 0 with a T period in order to encode the complete frame to transmit. This can then be transmitted to the infrared output with a method of modulation shown in the following diagram.

Figure 21-30. CIR Mode Block Components



In transmission, the LH software must exercise an element of real time control for transmitting data packets; they must each be emitted at a constant delay from the start bits of each of the individual packets which means when sending a series of packets, the packet to packet delay must respect a specific delay. To control this delay 2 methods can be used:

- By filling the TX FIFO with a number of zero bit which is transmitted with a T period.
- By using an external system timer which controls the delay either between each start of frame or between the end of a frame and the start of the next one. This can be performed:
  - By controlling the start of the frame through the configuration register MDR1[5] and ACREG[2] depending on the timer status (in case of control the delay between each start of frame).
  - By using the TX_STATUS interrupt IIR[5] to pre-load the next frame in the TX FIFO and to control the start of the timer (in case of control the delay between end of frame and start of next frame).

In reception, there are two ways to stop it :

- The LH can disable the reception by setting the ACREG[5] to 1 when it considers that the reception is finished because a large number of 0 has been received. To receive a new frame, the ACREG[5] must be set to 0.
- A specific mechanism, depending on the value set in the BOF length register (EBLR), allows for automatically stopping the reception. If the value set in the EBLR register is different than 0, this features is enabled and count a number of bit received at 0. When the counter achieves the value defined in the EBLR register, the reception is automatically stopped and RX_STOP_IT (IIR[2]) is set. When a 1 is detected on the RCRX pin, the reception is automatically enabled.

Note: There's a limitation when receiving data in UART CIR mode. The IrDA transceivers on the market have a common characteristic that shrinks the hold time of the received modulation pulse. The UART filtering schema on receiving is based on the same encoding mechanism used in transmission.

For the following scenario:

- Shift register period: 0.9 us
- Modulation frequency: 36 Khz
- Duty cycle: 1/4 of a modulation frequency period

The data sent in these conditions would look like 7us pulses within 28us period. The UART expects to receive similar incoming data on receive, but available transceiver timing characteristics typically send 2us modulated pulses. Those will be filtered out and RX FIFO won’t receive any data.

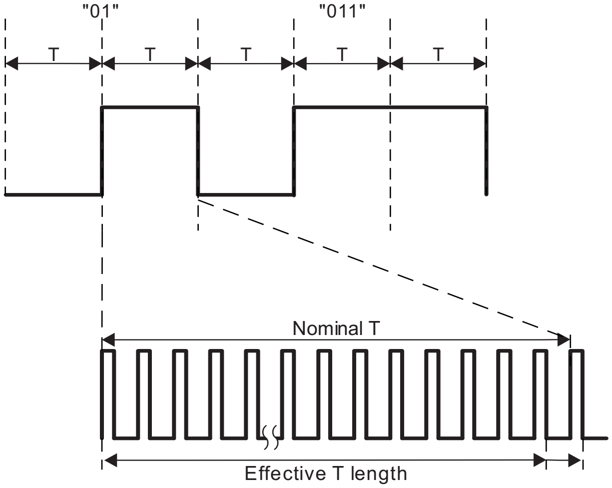
This does not affect UART CIR mode in transmission.

Note: The CIR RX demodulation can be bypassed by setting the MDR3[0] register bit.

21.3.8.3.3 Carrier Modulation

Looking closer at the actual modulation pulses of the infrared data stream, it should be noted that each modulated pulse that constitutes a digit is in fact a train of on/off pulses.

Figure 21-31. CIR Pulse Modulation



A minimum of 4 modulation pulses per bit is required by the module.

Based on the requested modulation frequency, the CFPS register must be set with the correct dividing value to provide the more accurate pulse frequency:

Dividing value = (FCLK/12)/MODfreq

Where FCLK = System clock frequency (48 MHz)

12 = real value of BAUD multiple

MODfreq = Effective frequency of the modulation (MHz)

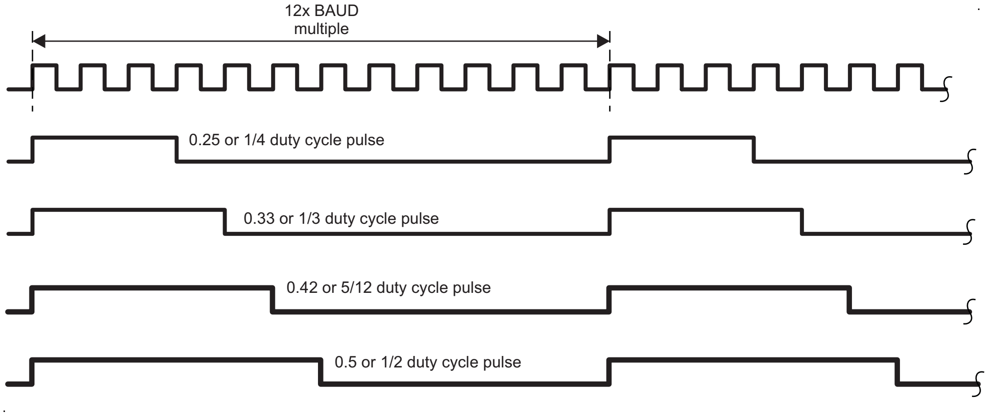
Example: For a targeted modulation frequency of 36 kHz, the CFPS value must be set to 111 in decimal which provide an modulation frequency of 36.04 kHz.

Note: The CFPS register is to start with a reset value of 105 (decimal) which translates to a frequency of 38.1 kHz.

The duty cycle of these pulses is user defined by the pulse duty register bits in the MDR2 configuration register.

MDR2[5:4]	Duty Cycle (High Level)
00	1/4
01	1/3
10	5/12
11	1/2

Figure 21-32. CIR Modulation Duty Cycle



The transmission logic ensures that all pulses are transmitted completely; i.e., there is no cut off of any pulses during its transmission. Furthermore, while transmitting continuous bytes back-to-back, no delay is inserted between two transmitted bytes. **Note:** The CIR RX demodulation can be bypassed by setting the MDR3[0] register bit. This bit will not affect the transmission modulation.

21.3.8.3.4 Frequency Divider Values

The data transferred is a succession of pulse with a T period. Depending on the standards used, the T period is defined through the DLL and DLH registers which defined the value to divide the functional clock (48 MHz):

Dividing value = (FCLK/16)/Tfreq

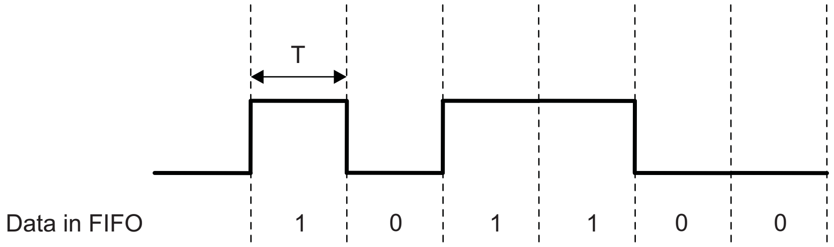
Where FCLK = System clock frequency (48 MHz)

16 = real value of BAUD multiple

Tfreq = Effective frequency of the T pulse (MHz)

In an example case using a variable pulse duration definitions:

Figure 21-33. Variable Pulse Duration Definitions



For a logical “1”, the pulse duration is equal to 2T and for a logical “0”, it’s equal to 4T.

If T =0.56 ms, the value coded into the DLH and DLL register must be 1680 in decimal.

21.4 UART/IrDA/CIR Basic Programming Model

21.4.1 UART Programming Model

21.4.1.1 Quick Start

This section describes the procedure for operating the UART with FIFO and DMA or interrupts. This three-part procedure ensures the quick start of the UART. It does not cover every UART feature.

The first programming model covers software reset of the UART. The second programming model describes FIFO and DMA configuration. The last programming model describes protocol, baud rate, and interrupt configuration.

NOTE: Each programming model can be used independently of the other two; for instance, reconfiguring the FIFOs and DMA settings only.

Each programming model can be executed starting from any UART register access mode (register modes, submodes, and other register dependencies). However, if the UART register access mode is known before executing the programming model, some steps that enable or restore register access are optional. For more information, see Section 21.3.7.1, *Register Access Modes*.

21.4.1.1.1 Software Reset

To clear the UART registers, perform the following steps:

1. Initiate a software reset:

   Set the UARTi.UART_SYSC[1] SOFTRESET bit to 1.
2. Wait for the end of the reset operation:

   Poll the UARTi.UART_SYSS[0] RESETDONE bit until it equals 1.

21.4.1.1.2 FIFOs and DMA Settings

To enable and configure the receive and transmit FIFOs and program the DMA mode, perform the following steps:

1. Switch to register configuration mode B to access the UARTi.UART_EFR register:
   a. Save the current UARTi.UART_LCR register value.
   b. Set the UARTi.UART_LCR register value to 0x00BF.
2. Enable register submode TCR_TLR to access the UARTi.UART_TLR register (part 1 of 2):
   a. Save the UARTi.UART_EFR[4] ENHANCED_EN value.
   b. Set the UARTi.UART_EFR[4] ENHANCED_EN bit to 1.
3. Switch to register configuration mode A to access the UARTi.UART_MCR register:

   Set the UARTi.UART_LCR register value to 0x0080.
4. Enable register submode TCR_TLR to access the UARTi.UART_TLR register (part 2 of 2):
   a. Save the UARTi.UART_MCR[6] TCR_TLR value.
   b. Set the UARTi.UART_MCR[6] TCR_TLR bit to 1.
5. Enable the FIFO; load the new FIFO triggers (part 1 of 3) and the new DMA mode (part 1 of 2):

   Set the following bits to the desired values:
   - UARTi.UART_FCR[7:6] RX_FIFO_TRIG
   - UARTi.UART_FCR[5:4] TX_FIFO_TRIG
   - UARTi.UART_FCR[3] DMA_MODE
   - UARTi.UART_FCR[0] FIFO_ENABLE (0: Disable the FIFO; 1: Enable the FIFO)

NOTE: The UARTi.UART_FCR register is not readable.

6. Switch to register configuration mode B to access the UARTi.UART_EFR register:
   Set the UARTi.UART_LCR register value to 0x00BF.
7. Load the new FIFO triggers (part 2 of 3):
   Set the following bits to the desired values:
   - UARTi.UART_TLR[7:4] RX_FIFO_TRIG_DMA
   - UARTi.UART_TLR[3:0] TX_FIFO_TRIG_DMA
8. Load the new FIFO triggers (part 3 of 3) and the new DMA mode (part 2 of 2):
   Set the following bits to the desired values:
   - UARTi.UART_SCR[7] RX_TRIG_GRANU1
   - UARTi.UART_SCR[6] TX_TRIG_GRANU1
   - UARTi.UART_SCR[2:1] DMA_MODE_2
   - UARTi.UART_SCR[0] DMA_MODE_CTL
9. Restore the UARTi.UART_EFR[4] ENHANCED_EN value saved in Step 2a.
10. Switch to register configuration mode A to access the UARTi.UART_MCR register:
    Set the UARTi.UART_LCR register value to 0x0080.
11. Restore the UARTi.UART_MCR[6] TCR_TLR value saved in Step 4a.
12. Restore the UARTi.UART_LCR value saved in Step 1a.

Triggers are used to generate interrupt and DMA requests. See Section 21.3.6.1.1, *Transmit FIFO Trigger,* to choose the following values:

- UARTi.UART_FCR[5:4] TX_FIFO_TRIG
- UARTi.UART_TLR[3:0] TX_FIFO_TRIG_DMA
- UARTi.UART_SCR[6] TX_TRIG_GRANU1

Triggers are used to generate interrupt and DMA requests. See Section 21.3.6.1.2, *Receive FIFO Trigger*, to choose the following values:

- UARTi.UART_FCR[7:6] RX_FIFO_TRIG
- UARTi.UART_TLR[7:4] RX_FIFO_TRIG_DMA
- UARTi.UART_SCR[7] RX_TRIG_GRANU1

DMA mode enables DMA requests. See Section 21.3.6.4, *FIFO DMA Mode Operation*, to choose the following values:

- UARTi.UART_FCR[3] DMA_MODE
- UARTi.UART_SCR[2:1] DMA_MODE_2
- UARTi.UART_SCR[0] DMA_MODE_CTL

21.4.1.1.3 Protocol, Baud Rate, and Interrupt Settings

To program the protocol, baud rate, and interrupt settings, perform the following steps:

1. Disable UART to access the UARTi.UART_DLL and UARTi.UART_DLH registers:
   Set the UARTi.UART_MDR1[2:0] MODE_SELECT bit field to 0x7.
2. Switch to register configuration mode B to access the UARTi.UART_EFR register:
   Set the UARTi.UART_LCR register value to 0x00BF.
3. Enable access to the UARTi.UART_IER[7:4] bit field:
   a. Save the UARTi.UART_EFR[4] ENHANCED_EN value.
   b. Set the UARTi.UART_EFR[4] ENHANCED_EN bit to 1.
4. Switch to register operational mode to access the UARTi.UART_IER register:
   Set the UARTi.UART_LCR register value to 0x0000.

5. Clear the UARTi.UART_IER register (set the UARTi.UART_IER[4] SLEEP_MODE bit to 0 to change the UARTi.UART_DLL and UARTi.UART_DLH registers). Set the UARTi.UART_IER register value to 0x0000.
6. Switch to register configuration mode B to access the UARTi.UART_DLL and UARTi.UART_DLH registers:

   Set the UARTi.UART_LCR register value to 0x00BF.
7. Load the new divisor value:

   Set the UARTi.UART_DLL[7:0] CLOCK_LSB and UARTi.UART_DLH[5:0] CLOCK_MSB bit fields to the desired values.
8. Switch to register operational mode to access the UARTi.UART_IER register:

   Set the UARTi.UART_LCR register value to 0x0000.
9. Load the new interrupt configuration (0: Disable the interrupt; 1: Enable the interrupt):

   Set the following bits to the desired values:
   - UARTi.UART_IER[7] CTS_IT
   - UARTi.UART_IER[6] RTS_IT
   - UARTi.UART_IER[5] XOFF_IT
   - UARTi.UART_IER[4] SLEEP_MODE
   - UARTi.UART_IER[3] MODEM_STS_IT
   - UARTi.UART_IER[2] LINE_STS_IT
   - UARTi.UART_IER[1] THR_IT
   - UARTi.UART_IER[0] RHR_IT
10. Switch to register configuration mode B to access the UARTi.UART_EFR register:

    Set the UARTi.UART_LCR register value to 0x00BF.
11. Restore the UARTi.UART_EFR[4] ENHANCED_EN value saved in Step 3a.
12. Load the new protocol formatting (parity, stop-bit, character length) and switch to register operational mode:

    Set the UARTi.UART_LCR[7] DIV_EN bit to 0.

    Set the UARTi.UART_LCR[6] BREAK_EN bit to 0.

    Set the following bits to the desired values:
    - UARTi.UART_LCR[5] PARITY_TYPE_2
    - UARTi.UART_LCR[4] PARITY_TYPE_1
    - UARTi.UART_LCR[3] PARITY_EN
    - UARTi.UART_LCR[2] NB_STOP
    - UARTi.UART_LCR[1:0] CHAR_LENGTH
13. Load the new UART mode:

    Set the UARTi.UART_MDR1[2:0] MODE_SELECT bit field to the desired value.

See Section 21.3.8.1.2, *Choosing the Appropriate Divisor Value*, to choose the following values:

- UARTi.UART_DLL[7:0] CLOCK_LSB
- UARTi.UART_DLH[5:0] CLOCK_MSB
- UARTi.UART_MDR1[2:0] MODE_SELECT

See Section 21.3.8.1.3.1, *Frame Formatting*, to choose the following values:

- UARTi.UART_LCR[5] PARITY_TYPE_2
- UARTi.UART_LCR[4] PARITY_TYPE_1
- UARTi.UART_LCR[3] PARITY_EN
- UARTi.UART_LCR[2] NB_STOP
- UARTi.UART_LCR[1:0] CHAR_LENGTH

21.4.1.2 Hardware and Software Flow Control Configuration

This section describes the programming steps to enable and configure hardware and software flow control. Hardware and software flow control cannot be used at the same time.

NOTE: Each programming model can be executed starting from any UART register access mode (register modes, submodes, and other register dependencies). However, if the UART register access mode is known before executing the programming model, some steps that enable or restore register access are optional. For more information, see Section 21.3.7.1, *Register Access Modes*.

21.4.1.2.1 Hardware Flow Control Configuration

To enable and configure hardware flow control, perform the following steps:

1. Switch to register configuration mode A to access the UARTi.UART_MCR register:
   a. Save the current UARTi.UART_LCR register value.
   b. Set the UARTi.UART_LCR register value to 0x0080.
2. Enable register submode TCR_TLR to access the UARTi.UART_TCR register (part 1 of 2):
   a. Save the UARTi.UART_MCR[6] TCR_TLR value.
   b. Set the UARTi.UART_MCR[6] TCR_TLR bit to 1.
3. Switch to register configuration mode B to access the UARTi.UART_EFR register:

   Set the UARTi.UART_LCR register value to 0x00BF.
4. Enable register submode TCR_TLR to access the UARTi.UART_TCR register (part 2 of 2):
   a. Save the UARTi.UART_EFR[4] ENHANCED_EN value.
   b. Set the UARTi.UART_EFR[4] ENHANCED_EN bit to 1.
5. Load the new start and halt trigger values for hardware flow control:

   Set the following bits to the desired values:
   - UARTi.UART_TCR[7:4] AUTO_RTS_START
   - UARTi.UART_TCR[3:0] AUTO_RTS_HALT
6. Enable or disable receive and transmit hardware flow control mode and restore the UARTi.UART_EFR[4] ENHANCED_EN value saved in Step 4a.

   Set the following bits to the desired values:
   - UARTi.UART_EFR[7] AUTO_CTS_EN (0: Disable; 1: Enable)
   - UARTi.UART_EFR[6] AUTO_RTS_EN (0: Disable; 1: Enable)

   Restore the UARTi.UART_EFR[4] ENHANCED_EN bit to the saved value.
7. Switch to register configuration mode A to access the UARTi.UART_MCR register:

   Set the UARTi.UART_LCR register value to 0x0080.
8. Restore the UARTi.UART_MCR[6] TCR_TLR value saved in Step 2a.
9. Restore the UARTi.UART_LCR value saved in Step 1a.

See Section 21.3.8.1.3.2, *Hardware Flow Control*, to choose the following values:

- UARTi.UART_EFR[7] AUTO_CTS_EN
- UARTi.UART_EFR[6] AUTO_RTS_EN
- UARTi.UART_TCR[7:4] AUTO_RTS_START
- UARTi.UART_TCR[3:0] AUTO_RTS_HALT

21.4.1.2.2 Software Flow Control Configuration

To enable and configure software flow control, perform the following steps:

1. Switch to register configuration mode B to access the UARTi.UART_EFR register.
    a. Save the current UARTi.UART_LCR register value.
    b. Set the UARTi.UART_LCR register value to 0x00BF.
2. Enable register submode XOFF to access the UARTi.UART_XOFF1 and UARTi.UART_XOFF2 registers:
    a. Save the UARTi.UART_EFR[4] ENHANCED_EN value.
    b. Set the UARTi.UART_EFR[4] ENHANCED_EN bit to 0.
3. Load the new software flow control characters:

    Set the following bits to the desired values:
    - UARTi.UART_XON1_ADDR1[7:0] XON_WORD1
    - UARTi.UART_XON2_ADDR2[7:0] XON_WORD2
    - UARTi.UART_XOFF1[7:0] XOFF_WORD1
    - UARTi.UART_XOFF2[7:0] XOFF_WORD2
4. Enable access to the UARTi.UART_MCR[7:5] bit field and enable register submode TCR_TLR to access the UARTi.UART_TCR register (part 1 of 2):

    Set the UARTi.UART_EFR[4] ENHANCED_EN bit to 1.
5. Switch to register configuration mode A to access the UARTi.UART_MCR register:

    Set the UARTi.UART_LCR register value to 0x0080.
6. Enable register submode TCR_TLR to access the UARTi.UART_TCR register (part 2 of 2) and enable or disable XON any function:
    a. Save the UARTi.UART_MCR[6] TCR_TLR value.
    b. Set the UARTi.UART_MCR[6] TCR_TLR bit to 1.
    c. Set the UARTi.UART_MCR[5] XON_EN bit to the desired value (0: Disable; 1: Enable).
7. Switch to register configuration mode B to access the UARTi.UART_EFR register:

    Set the UARTi.UART_LCR register value to 0x00BF.
8. Load the new start and halt trigger values for software flow control:

    Set the following bits to the desired values:
    - UARTi.UART_TCR[7:4] AUTO_RTS_START
    - UARTi.UART_TCR[3:0] AUTO_RTS_HALT
9. Enable or disable special character function and load the new software flow control mode and restore the UARTi.UART_EFR[4] ENHANCED_EN value saved in Step 2a:

    Set the following bits to the desired values:
    - UARTi.UART_EFR[5] SPEC_CHAR (0: Disable; 1: Enable)
    - UARTi.UART_EFR[3:0] SW_FLOW_CONTROL

    Restore the UARTi.UART_EFR[4] ENHANCED_EN bit to the saved value.
10. Switch to register configuration mode A to access the UARTi.UART_MCR register:

    Set the UARTi.UART_LCR register value to 0x0080.
11. Restore the UARTi.UART_MCR[6] TCR_TLR bit value saved in Step 6a.
12. Restore the UARTi.UART_LCR value saved in Step 1a.

See Section 21.3.8.1.3.3, *Software Flow Control*, to choose the following values:

- UARTi.UART_EFR[5] SPEC_CHAR
- UARTi.UART_EFR[3:0] SW_FLOW_CONTROL
- UARTi.UART_TCR[7:4] AUTO_RTS_START
- UARTi.UART_TCR[3:0] AUTO_RTS_HALT

- UARTi.UART_XON1_ADDR1[7:0] XON_WORD1
- UARTi.UART_XON2_ADDR2[7:0] XON_WORD2
- UARTi.UART_XOFF1[7:0] XOFF_WORD1
- UARTi.UART_XOFF2[7:0] XOFF_WORD2

21.4.2 IrDA Programming Model

21.4.2.1 SIR Mode

21.4.2.1.1 Receive

The following programming model explains how to program the module to receive an IrDA frame with parity forced to 1, baud rate = 112.5KB, FIFOs disabled, 2 stop-bits, and 8-bit word length:

1. Disable the UART before accessing the UARTi.UART_DLL and UARTi.UART_DLH registers: Set the UARTi.UART_MDR1[2:0] MODE_SELECT bit field to 0x7.
2. Grant access to the UART_DLL and UART_DLH registers (the UART_LCR[7] DIV_EN bit = 1): UARTi.UART_LCR = 0x80 (Data format is unaffected by the use and settings of the UARTi.UART_LCR register in IrDA mode.)
3. Load the new baud rate (115.2 kbps):

   UARTi.UART_DLL = 0x1A
   UARTi.UART_DLH = 0x00
4. Set SIR mode: UARTi.UART_MDR1[2:0] MODE_SELECT = 0x1
5. Disable access to the UART_DLL and UART_DLH registers and switch to register operational mode: UARTi.UART_LCR = 0x00.
6. Optional: Enable the RHR interrupt:

   UARTi.UART_IER[0] RHR_IT = 0x1

21.4.2.1.2 Transmit

The following programming model explains how to program the module to transmit an IrDA 6-byte frame with no parity, baud rate = 112.5 kbps, FIFOs disabled, 3/16 encoding, 2 stop-bits, and 7-bit word length:

1. Disable the UART before accessing the UARTi.UART_DLL and UARTi.UART_DLH registers: Set the UART_MDR1[2:0] MODE_SELECT bit field to 0x7.
2. Grant access to the UART_EFR register:

   UARTi.UART_LCR = 0xBF
3. Enable the enhanced features (the UART_EFR[4] ENAHNCED_EN bit = 1):

   Set the UARTi.UART_EFR register value to 0x10.
4. Grant access to the UART_DLL and UART_DLH registers (the UART_LCR[7] DIV_EN bit = 1):

   UARTi.UART_LCR = 0x80 (Data format is unaffected by the use and settings of the UARTi.UART_LCR register in IrDA mode.)
5. Load the new baud rate (115.2 kbps):

   UARTi.UART_DLL = 0x1A
   UARTi.UART_DLH = 0x00
6. Set SIR mode (the UART_MDR1[2:0] MODE_SELECT bit field = 0x1): UARTi.UART_MDR1 = 0x01
7. Disable access to the UART_DLL and UART_DLH registers and switch to register operational mode: UARTi.UART_LCR = 0x00.
8. Force DTR output to active:

   UARTi.UART_MCR[0] DTR = 1
9. Optional: Enable the THR interrupt:

   UARTi.UART_IER[1] THR_IT = 1
10. Set transmit frame length to 6 bytes:

    UARTi.UART_TXFLL = 0x06
11. Set 7 starts of frame transmission:

    UARTi.UART_EBLR = 0x08

12. Optional: Set SIR pulse width to be 1.6 us:
    UARTi.UART_ACREG[7] PULSE_TYPE = 1
13. Load the UART_THR register with the data to be transmitted.

21.4.2.2 MIR Mode

21.4.2.2.1 Receive

The following programming model explains how to program the module to receive an IrDA frame with no parity, baud rate = 1.152 Mpbs, and FIFOs disabled.

1. Disable the UART before accessing the UARTi.UART_DLL and UARTi.UART_DLH registers:
   Set the UARTi.UART_MDR1[2:0] MODE_SELECT bit field to 0x7.
2. Grant access to the UART_DLL and UART_DLH registers (UART_LCR[7] DIV_EN bit = 1):
   UARTi.UART_LCR = 0x80 (Data format is unaffected by the use and settings of the UARTi.UART_LCR register in IrDA mode.)
3. Load the new baud rate (1.152 Mpbs):
   UARTi.UART_DLL = 0x01
   UARTi.UART_DLH = 0x00
4. Set MIR mode:
   UARTi.UART_MDR1[2:0] MODE_SELECT = 0x4
5. Disable access to the UART_DLL and UART_DLH registers and switch to register operational mode:
   UARTi.UART_LCR = 0x00
6. Force DTR output to active (UART_MCR[0] DTR = 1):
   Force RTS output to active (UART_MCR[1] RTS = 1).
   UARTi.UART_MCR = 0x3
7. Optional: Enable the RHR interrupt:
   UARTi.UART_IER[0] RHR_IT = 1

21.4.2.2.2 Transmit

The following programming model explains how to program the module to transmit an IrDA 60-byte frame with no parity, baud rate = 1.152 Mpbs, and FIFOs disabled:

1. Disable the UART before accessing the UARTi.UART_DLL and UARTi.UART_DLH registers:
   Set the UARTi.UART_MDR1[2:0] MODE_SELECT bit field to 0x7.
2. Grant access to the UART_DLL and UART_DLH registers (UART_LCR[7] DIV_EN bit = 1):
   UARTi.UART_LCR = 0x80 (Data format is unaffected by the use and settings of the UARTi.UART_LCR register in IrDA mode.)
3. Load the new baud rate (1.152 Mpbs):
   UARTi.UART_DLL = 0x01
   UARTi.UART_DLH = 0x00
4. Set MIR mode:
   UARTi.UART_MDR1[2:0] MODE_SELECT = 0x4
5. Disable access to the UART_DLL and UART_DLH registers and switch to register operational mode:
   UARTi.UART_LCR = 0x00
6. Force DTR output to active:
   UARTi.UART_MCR[0] DTR = 1
7. Optional: Enable the THR interrupt:
   UARTi.UART_IER[1] THR_IT = 1
8. Set the frame length to 60 bytes:
   UARTi.UART_TXFLL = 0x3C
9. Optional: Transmit eight additional starts of frame (MIR mode requires two starts):

UARTi.UART_EBLR = 0x08

10. SIP is sent at the end of transmission:

    UARTi.UART_ACREG[3] = 1

11. Load the UART_THR register with the data to be transmitted.

21.4.2.3 FIR Mode

21.4.2.3.1 Receive

The following programming model explains how to program the module to receive the IrDA frame with no parity, baud rate = 4 Mbps, FIFOs enabled, 8-bit word length.

1. Disable the UART before accessing the UARTi.UART_DLL and UARTi.UART_DLH registers: Set the UARTi.UART_MDR1[2:0] MODE_SELECT bit field to 0x7.
2. Grant access to the UART_DLL and UART_DLH registers (UART_LCR[7] DIV_EN bit = 1):

   UARTi.UART_LCR = 0x80 (Data format is unaffected by the use and settings of the UARTi.UART_LCR register in IrDA mode.)
3. FIFO clear and enable:

   UARTi.UART_FCR = 0x7 (TX/RX FIFO trigger: UART_FCR[7:6] and UART_FCR[5:4])
   UARTi.UART_LCR[7] = 0
4. Set FIR mode:
   UARTi.UART_MDR1[2:0] MODE_SELECT = 0x5
5. Set frame length:

   UARTi.UART_RXFLL = 0xA (Data + CRC + STOP)
6. Disable access to the UARTi.UART_DLL registers and UARTi.UART_DLH and switch to register operational mode:

   UARTi.UART_LCR[7] DIV_EN = 0x0
7. Optional: Enable the RHR interrupt:

   UARTi.UART_IER[0] RHR_IT = 1

21.4.2.3.2 Transmit

The following programming model explains how to program the module to transmit an IrDA 4-byte frame with no parity, baud rate = 4 Mbps, FIFOs enabled, and 8-bit word length.

1. Disable the UART before accessing the UARTi.UART_DLL and UARTi.UART_DLH registers: Set the UARTi.UART_MDR1[2:0] MODE_SELECT bit field to 0x7.
2. Grant access to EFR_REG:

   UARTi.UART_LCR = 0xBF
3. Enable the enhanced features (EFR_REG[4] ENAHNCED_EN = 0x1):

   UARTi.UART_EFR = 0x10
4. FIFO clear and enable:
   UARTi.UART_FCR = 0x7 (TX/RX FIFO trigger: UART_FCR[7:6] and UART_FCR[5:4]).

   UARTi.UART_LCR[7] = 0
5. Set FIR mode and enable auto-SIP mode:

   UARTi.UART_MDR1 = 0x45
6. Set frame length:

   UARTi.UART_TXFLL = 0x4
   UARTi.UART_TXFLH = 0x0
   UARTi.UART_RXFLL = 0xA (Data + CRC + STOP)
   UARTi.UART_RXFLH = 0x0
7. Force DTR output to active:

   UARTi.UART_MCR[0] DTR = 0x1

8. Optional: Enable the THR interrupt:
   UARTi.UART_IER[1] THR_IT = 0x1
9. Optional: Transmit eight additional starts of frame (MIR mode requires two starts):
   UARTi.UART_EBLR = 0x08
10. SIP is sent at the end of transmission:
    UARTi.UART_ACREG[3] = 1
11. Load the UART_THR register with the data to be transmitted.

21.5 UART Registers

21.5.1 UART Registers

Table 21-29 lists the memory-mapped registers for the UART. All register offset addresses not listed in Table 21-29 should be considered as reserved locations and the register contents should not be modified.

Table 21-29. UART Registers

Offset	Acronym	Register Name	Section
0h	UART_THR	Transmit Holding Register	Section 21.5.1.1
0h	UART_RHR	Receiver Holding Register	Section 21.5.1.2
0h	UART_DLL	Divisor Latches Low Register	Section 21.5.1.3
4h	UART_IER_IRDA	Interrupt Enable Register (IrDA)	Section 21.5.1.4
4h	UART_IER_CIR	Interrupt Enable Register (CIR)	Section 21.5.1.5
4h	UART_IER	Interrupt Enable Register (UART)	Section 21.5.1.6
4h	UART_DLH	Divisor Latches High Register	Section 21.5.1.7
8h	UART_EFR	Enhanced Feature Register	Section 21.5.1.8
8h	UART_IIR	Interrupt Identification Register (UART)	Section 21.5.1.9
8h	UART_IIR_CIR	Interrupt Identification Register (CIR)	Section 21.5.1.10
8h	UART_FCR	FIFO Control Register	Section 21.5.1.11
Ch	UART_LCR	Line Control Register	Section 21.5.1.12
10h	UART_MCR	Modem Control Register	Section 21.5.1.13
10h	UART_XON1_ADDR1	XON1/ADDR1 Register	Section 21.5.1.14
14h	UART_XON2_ADDR2	XON2/ADDR2 Register	Section 21.5.1.15
14h	UART_LSR_CIR	Line Status Register (CIR)	Section 21.5.1.16
14h	UART_LSR_IRDA	Line Status Register (IrDA)	Section 21.5.1.17
14h	UART_LSR	Line Status Register (UART)	Section 21.5.1.18
18h	UART_TCR	Transmission Control Register	Section 21.5.1.19
18h	UART_MSR	Modem Status Register	Section 21.5.1.20
18h	UART_XOFF1	XOFF1 Register	Section 21.5.1.21
1Ch	UART_SPR	Scratchpad Register	Section 21.5.1.22
1Ch	UART_TLR	Trigger Level Register	Section 21.5.1.23
1Ch	UART_XOFF2	XOFF2 Register	Section 21.5.1.24
20h	UART_MDR1	Mode Definition Register 1	Section 21.5.1.25
24h	UART_MDR2	Mode Definition Register 2	Section 21.5.1.26
28h	UART_TXFLL	Transmit Frame Length Low Register	Section 21.5.1.27
28h	UART_SFLSR	Status FIFO Line Status Register	Section 21.5.1.28
2Ch	UART_RESUME	RESUME Register	Section 21.5.1.29
2Ch	UART_TXFLH	Transmit Frame Length High Register	Section 21.5.1.30
30h	UART_RXFLL	Received Frame Length Low Register	Section 21.5.1.31
30h	UART_SFREGL	Status FIFO Register Low	Section 21.5.1.32
34h	UART_SFREGH	Status FIFO Register High	Section 21.5.1.33

Table 21-29. UART Registers (continued)

Offset	Acronym	Register Name	Section
34h	UART_RXFLH	Received Frame Length High Register	Section 21.5.1.34
38h	UART_BLR	BOF Control Register	Section 21.5.1.35
38h	UART_UASR	UART Autobauding Status Register	Section 21.5.1.36
3Ch	UART_ACREG	Auxiliary Control Register	Section 21.5.1.37
40h	UART_SCR	Supplementary Control Register	Section 21.5.1.38
44h	UART_SSR	Supplementary Status Register	Section 21.5.1.39
48h	UART_EBLR	BOF Length Register	Section 21.5.1.40
50h	UART_MVR	Module Version Register	Section 21.5.1.41
54h	UART_SYSC	System Configuration Register	Section 21.5.1.42
58h	UART_SYSS	System Status Register	Section 21.5.1.43
5Ch	UART_WER	Wake-Up Enable Register	Section 21.5.1.44
60h	UART_CFPS	Carrier Frequency Prescaler Register	Section 21.5.1.45
64h	UART_RXFIFO_LVL	Received FIFO Level Register	Section 21.5.1.46
68h	UART_TXFIFO_LVL	Transmit FIFO Level Register	Section 21.5.1.47
6Ch	UART_IER2	IER2 Register	Section 21.5.1.48
70h	UART_ISR2	ISR2 Register	Section 21.5.1.49
74h	UART_FREQ_SEL	FREQ_SEL Register	Section 21.5.1.50
80h	UART_MDR3	Mode Definition Register 3	Section 21.5.1.51
84h	UART_TX_DMA_THR	TX DMA Threshold Register	Section 21.5.1.52

21.5.1.1 UART_THR Register (offset = 0h) [reset = 0h]

UART_THR is shown in Figure 21-34 and described in Table 21-30.

The transmit holding register (THR) is selected with the register bit setting of LCR[7] = 0. The transmitter section consists of the transmit holding register and the transmit shift register. The transmit holding register is a 64-byte FIFO. The MPU writes data to the THR. The data is placed in the transmit shift register where it is shifted out serially on the TX output. If the FIFO is disabled, location zero of the FIFO is used to store the data.

Figure 21-34. UART_THR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
THR							
W-0h							

Table 21-30. UART_THR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-0	THR	W	0h	Transmit holding register. Value 0 to FFh.

21.5.1.2 UART_RHR Register (offset = 0h) [reset = 0h]

UART_RHR is shown in Figure 21-35 and described in Table 21-31.

The receiver holding register (RHR) is selected with the register bit setting of LCR[7] = 0. The receiver section consists of the receiver holding register and the receiver shift register. The RHR is actually a 64-byte FIFO. The receiver shift register receives serial data from RX input. The data is converted to parallel data and moved to the RHR. If the FIFO is disabled, location zero of the FIFO is used to store the single data character. If an overflow occurs, the data in the RHR is not overwritten.

Figure 21-35. UART_RHR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RHR							
R-0h							

Table 21-31. UART_RHR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-0	RHR	R	0h	Receive holding register. Value 0 to FFh.

21.5.1.3 UART_DLL Register (offset = 0h) [reset = 0h]

UART_DLL is shown in Figure 21-36 and described in Table 21-32.

The divisor latches low register (DLL) is selected with a register bit setting of LCR[7] not equal to BFh or LCR[7] = BFh. The divisor latches low register (DLL) with the DLH register stores the 14-bit divisor for generation of the baud clock in the baud rate generator. DLH stores the most-significant part of the divisor, DLL stores the least-significant part of the divisor. DLL and DLH can be written to only before sleep mode is enabled (before IER[4] is set).

Figure 21-36. UART_DLL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CLOCK_LSB							
R/W-0h							

Table 21-32. UART_DLL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-0	CLOCK_LSB	R/W	0h	Divisor latches low. Stores the 8 LSB divisor value. Value 0 to FFh.

21.5.1.4 UART_IER_IRDA Register (offset = 4h) [reset = 0h]

UART_IER_IRDA is shown in Figure 21-37 and described in Table 21-33.

The following interrupt enable register (IER) description is for IrDA mode. The IrDA IER is selected with a register bit setting of LCR[7] = 0. In IrDA mode, EFR[4] has no impact on the access to IER[7:4]. The IrDA interrupt enable register (IER) can be programmed to enable/disable any interrupt. There are 8 types of interrupt in these modes, received EOF, LSR interrupt, TX status, status FIFO interrupt, RX overrun, last byte in RX FIFO, THR interrupt, and RHR interrupt. Each interrupt can be enabled/disabled individually. The TXSTATUSIT interrupt reflects two possible conditions. The MDR2[0] bit should be read to determine the status in the event of this interrupt.

Figure 21-37. UART_IER_IRDA Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
EOFIT	LINESTSIT	TXSTSIT	STSFIFOTRIGIT	RXOVERRUNIT	LASTRXBYTEIT	THRIT	RHRIT
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 21-33. UART_IER_IRDA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7	EOFIT	R/W	0h	EOFIT 0h (R/W) = Disables the received EOF interrupt. 1h (R/W) = Enables the received EOF interrupt.
6	LINESTSIT	R/W	0h	LINESTSIT 0h (R/W) = Disables the receiver line status interrupt. 1h (R/W) = Enables the receiver line status interrupt.
5	TXSTSIT	R/W	0h	TXSTATUSIT 0h (R/W) = Disables the TX status interrupt. 1h (R/W) = Enables the TX status interrupt.
4	STSFIFOTRIGIT	R/W	0h	STSFIFOTRIGIT 0h (R/W) = Disables status FIFO trigger level interrupt. 1h (R/W) = Enables status FIFO trigger level interrupt.
3	RXOVERRUNIT	R/W	0h	RXOVERRUNIT 0h (R/W) = Disables the RX overrun interrupt. 1h (R/W) = Enables the RX overrun interrupt.
2	LASTRXBYTEIT	R/W	0h	LASTRXBYTEIT 0h (R/W) = Disables the last byte of frame in RX FIFO interrupt. 1h (R/W) = Enables the last byte of frame in RX FIFO interrupt.
1	THRIT	R/W	0h	THRIT 0h (R/W) = Disables the THR interrupt. 1h (R/W) = Enables the THR interrupt.
0	RHRIT	R/W	0h	RHRIT 0h (R/W) = Disables the RHR interrupt. 1h (R/W) = Enables the RHR interrupt.

21.5.1.5 UART_IER_CIR Register (offset = 4h) [reset = 0h]

UART_IER_CIR is shown in Figure 21-38 and described in Table 21-34.

The following interrupt enable register (IER) description is for CIR mode. The CIR IER is selected with a register bit setting of LCR[7] = 0. In IrDA mode, EFR[4] has no impact on the access to IER[7:4]. The CIR interrupt enable register (IER) can be programmed to enable/disable any interrupt. There are 5 types of interrupt in these modes, TX status, RX overrun, RX stop interrupt, THR interrupt, and RHR interrupt. Each interrupt can be enabled/disabled individually. In CIR mode, the TXSTATUSIT bit has only one meaning corresponding to the case MDR2[0] = 0. The RXSTOPIT interrupt is generated based on the value set in the BOF Length register (EBLR).

Figure 21-38. UART_IER_CIR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED		TXSTSIT	RESERVED	RXOVERRUNIT	RXSTOPIT	THRIT	RHRIT
R-0h		R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 21-34. UART_IER_CIR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-6	RESERVED	R	0h	
5	TXSTSIT	R/W	0h	TXSTATUSIT. 0h (R/W) = Disables the TX status interrupt. 1h (R/W) = Enables the TX status interrupt.
4	RESERVED	R	0h	
3	RXOVERRUNIT	R/W	0h	RXOVERRUNIT. 0h (R/W) = Disables the RX overrun interrupt. 1h (R/W) = Enables the RX overrun interrupt.
2	RXSTOPIT	R/W	0h	RXSTOPIT. 0h (R/W) = Disables the RX stop interrupt. 1h (R/W) = Enables the RX stop interrupt.
1	THRIT	R/W	0h	THRIT. 0h (R/W) = Disables the THR interrupt. 1h (R/W) = Enables the THR interrupt.
0	RHRIT	R/W	0h	RHRIT. 0h (R/W) = Disables the RHR interrupt. 1h (R/W) = Enables the RHR interrupt.

21.5.1.6 UART_IER Register (offset = 4h) [reset = 0h]

UART_IER is shown in Figure 21-39 and described in Table 21-35.

The following interrupt enable register (IER) description is for UART mode. The UART IER is selected with a register bit setting of LCR[7] = 0. In UART mode, IER[7:4] can only be written when EFR[4] = 1 The interrupt enable register (IER) can be programmed to enable/disable any interrupt. There are seven types of interrupt in this mode: receiver error, RHR interrupt, THR interrupt, XOFF received and CTS (active-low)/RTS (active-low) change of state from low to high. Each interrupt can be enabled/disabled individually. There is also a sleep mode enable bit. The UART interrupt enable register (IER) is shown in and described in .

Figure 21-39. UART_IER Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CTSIT	RTSIT	XOFFIT	SLEEPMODE	MODEMSTSIT	LINESTSIT	THRIT	RHRIT
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 21-35. UART_IER Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7	CTSIT	R/W	0h	Can be written only when EFR[4] = 1. 0h (R/W) = Disables the CTS (active-low) interrupt. 1h (R/W) = Enables the CTS (active-low) interrupt.
6	RTSIT	R/W	0h	Can be written only when EFR[4] = 1. 0h (R/W) = Disables the RTS (active-low) interrupt. 1h (R/W) = Enables the RTS (active-low) interrupt.
5	XOFFIT	R/W	0h	Can be written only when EFR[4] = 1. 0h (R/W) = Disables the XOFF interrupt. 1h (R/W) = Enables the XOFF interrupt.
4	SLEEPMODE	R/W	0h	Can be only written when EFR[4] = 1. 0h (R/W) = Disables sleep mode. 1h (R/W) = Enables sleep mode (stop baud rate clock when the module is inactive).
3	MODEMSTSIT	R/W	0h	MODEMSTSIT. 0h (R/W) = Disables the modem status register interrupt. 1h (R/W) = Enables the modem status register interrupt
2	LINESTSIT	R/W	0h	LINESTSIT. 0h (R/W) = Disables the receiver line status interrupt. 1h (R/W) = Enables the receiver line status interrupt.
1	THRIT	R/W	0h	THRIT. 0h (R/W) = Disables the THR interrupt. 1h (R/W) = Enables the THR interrupt.
0	RHRIT	R/W	0h	RHRIT. 0h (R/W) = Disables the RHR interrupt and time out interrupt. 1h (R/W) = Enables the RHR interrupt and time out interrupt.

21.5.1.7 UART_DLH Register (offset = 4h) [reset = 0h]

UART_DLH is shown in Figure 21-40 and described in Table 21-36.

The divisor latches high register (DLH) is selected with a register bit setting of LCR[7] not equal to BFh or LCR[7] = BFh. The divisor latches high register (DLH) with the DLL register stores the 14-bit divisor for generation of the baud clock in the baud rate generator. DLH stores the most-significant part of the divisor, DLL stores the least-significant part of the divisor. DLL and DLH can be written to only before sleep mode is enabled (before IER[4] is set).

Figure 21-40. UART_DLH Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED		CLOCK_MSB					
R-0h		R/W-0h					

Table 21-36. UART_DLH Register Field Descriptions

Bit	Field	Type	Reset	Description
15-6	RESERVED	R	0h	
5-0	CLOCK_MSB	R/W	0h	Divisor latches high. Stores the 6 MSB divisor value. Value 0 to 3Fh.

21.5.1.8 UART_EFR Register (offset = 8h) [reset = 0h]

UART_EFR is shown in Figure 21-41 and described in Table 21-37.

The enhanced feature register (EFR) is selected with a register bit setting of LCR[7] = BFh. The enhanced feature register (EFR) enables or disables enhanced features. Most enhanced functions apply only to UART modes, but EFR[4] enables write accesses to FCR[5:4], the TX trigger level, which is also used in IrDA modes.

Figure 21-41. UART_EFR Register

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>AUTOCTSEN</td><td>AUTORTSEN</td><td>SPECIALCHAR DETECT</td><td>ENHANCEDEN</td><td colspan="4">SWFLOWCTRL</td></tr>
<tr><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 21-37. UART_EFR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7	AUTOCTSEN	R/W	0h	Auto-CTS enable bit (UART mode only). 0h (R/W) = Normal operation. 1h (R/W) = Auto-CTS flow control is enabled; transmission is halted when the CTS (active-low) pin is high (inactive).
6	AUTORTSEN	R/W	0h	Auto-RTS enable bit (UART mode only). 0h (R/W) = Normal operation. 1h (R/W) = Auto-RTS flow control is enabled; RTS (active-low) pin goes high (inactive) when the receiver FIFO HALT trigger level, TCR[3:0], is reached and goes low (active) when the receiver FIFO RESTORE transmission trigger level is reached.
5	SPECIALCHARDETECT	R/W	0h	Special character detect (UART mode only). 0h (R/W) = Normal operation. 1h (R/W) = Special character detect enable. Received data is compared with XOFF2 data. If a match occurs, the received data is transferred to RX FIFO and the IIR[4] bit is set to 1 to indicate that a special character was detected.
4	ENHANCEDEN	R/W	0h	Enhanced functions write enable bit. 0h (R/W) = Disables writing to IER[7:4], FCR[5:4], and MCR[7:5]. 1h (R/W) = Enables writing to IER[7:4], FCR[5:4], and MCR[7:5].

Table 21-37. UART_EFR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	SWFLOWCTRL	R/W	0h	Combinations of software flow control can be selected by programming this bit. XON1 and XON2 should be set to different values if the software flow control is enabled. The TX and RX software flow control options are as follows. EFR[3] = 0, EFR[2] = 0, EFR[1] = X, and EFR[0] = X, then: No transmit flow control. EFR[3] = 1, EFR[2] = 0, EFR[1] = X, and EFR[0] = X, then: Transmit XON1, XOFF1. EFR[3] = 0, EFR[2] = 1, EFR[1] = X, and EFR[0] = X, then: Transmit XON2, XOFF2. EFR[3] = 1, EFR[2] = 1, EFR[1] = X, and EFR[0] = X, then: Transmit XON1, XON2 or XOFF1, XOFF2. The XON1 and XON2 characters or the XOFF1 and XOFF2 characters must be transmitted/received sequentially with XON1/XOFF1 followed by XON2/XOFF2. EFR[3] = X, EFR[2] = X, EFR[1] = 0, and EFR[0] = 0, then: No receive flow control. EFR[3] = X, EFR[2] = X, EFR[1] = 1, and EFR[0] = 0, then: Receiver compares XON1, XOFF1. EFR[3] = X, EFR[2] = X, EFR[1] = 0, and EFR[0] = 1, then: Receiver compares XON2, XOFF2. EFR[3] = X, EFR[2] = X, EFR[1] = 1, and EFR[0] = 1, then: Receiver compares XON1, XON2 or XOFF1, XOFF2. The XON1 and XON2 characters or the XOFF1 and XOFF2 characters must be transmitted/received sequentially with XON1/XOFF1 followed by XON2/XOFF2. In IrDA mode, EFR[1] and EFR[0] select the IR address to check (see IR Address Checking).

21.5.1.9 UART_IIR Register (offset = 8h) [reset = 1h]

UART_IIR is shown in Figure 21-42 and described in Table 21-38.

The following interrupt identification register (IIR) description is for UART mode. The UART IIR is selected with a register bit setting of LCR[7] = 0 or LCR[7] not equal to BFh. The UART interrupt identification register (IIR) is a read-only register that provides the source of the interrupt. An interrupt source can be flagged only if enabled in the IER register.

Figure 21-42. UART_IIR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
FCR_MIRROR		IT_TYPE					IT_PENDING
R-0h		R-0h					R-1h

Table 21-38. UART_IIR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-6	FCR_MIRROR	R	0h	Mirror the contents of FCR[0] on both bits.
5-1	IT_TYPE	R	0h	Seven possible interrupts in UART mode. Other combinations never occur: 0h (R/W) = Modem interrupt. Priority = 4. 1h (R/W) = THR interrupt. Priority = 3. 2h (R/W) = RHR interrupt. Priority = 2. 3h (R/W) = Receiver line status error. Priority = 1. 4h (R/W) = Reserved 5h (R/W) = Reserved 6h (R/W) = Rx timeout. Priority = 2. 7h (R/W) = Reserved 8h (R/W) = Xoff/special character. Priority = 5. 9h (R/W) = Reserved, from 9h to Fh. 10h (R/W) = CTS (active-low), RTS (active-low), DSR (active-low) change state from active (low) to inactive (high). Priority = 6. 11h (R/W) = Reserved, from 11 to 1Fh.
0	IT_PENDING	R	1h	Interrupt pending. 0h (R/W) = An interrupt is pending. 1h (R/W) = No interrupt is pending.

21.5.1.10 UART_IIR_CIR Register (offset = 8h) [reset = 0h]

UART_IIR_CIR is shown in Figure 21-43 and described in Table 21-39.

The following interrupt identification register (IIR) description is for CIR mode. The CIR IIR is selected with a register bit setting of LCR[7] = 0 or LCR[7] not equal to BFh. The CIR interrupt identification register (IIR) is a read-only register that provides the source of the interrupt. An interrupt source can be flagged only if enabled in the IER register.

Figure 21-43. UART_IIR_CIR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED		TXSTSIT	RESERVED	RXOEIT	RXSTOPIT	THRIT	RHRIT
R-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 21-39. UART_IIR_CIR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-6	RESERVED	R	0h	
5	TXSTSIT	R	0h	TXSTATUSIT 0h (R/W) = TX status interrupt inactive 1h (R/W) = TX status interrupt active
4	RESERVED	R	0h	
3	RXOEIT	R	0h	RXOEIT 0h (R/W) = RX overrun interrupt inactive 1h (R/W) = RX overrun interrupt active
2	RXSTOPIT	R	0h	RXSTOPIT 0h (R/W) = Receive stop interrupt is inactive 1h (R/W) = Receive stop interrupt is active
1	THRIT	R	0h	THRIT 0h (R/W) = THR interrupt inactive 1h (R/W) = THR interrupt active
0	RHRIT	R	0h	RHRIT 0h (R/W) = RHR interrupt inactive 1h (R/W) = RHR interrupt active

21.5.1.11 UART_FCR Register (offset = 8h) [reset = 0h]

UART_FCR is shown in Figure 21-44 and described in Table 21-40.

The FIFO control register (FCR) is selected with a register bit setting of LCR[7] = 0 or LCR[7] not equal to BFh. FCR[5:4] can only be written when EFR[4] = 1.

Figure 21-44. UART_FCR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RX_FIFO_TRIG		TX_FIFO_TRIG		DMA_MODE	TX_FIFO_CLR	RX_FIFO_CLR	FIFO_EN
W-0h		W-0h		W-0h	W-0h	W-0h	W-0h

Table 21-40. UART_FCR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-6	RX_FIFO_TRIG	W	0h	Sets the trigger level for the RX FIFO: If SCR[7] = 0 and TLR[7] to TLR[4] not equal to 0000, RX_FIFO_TRIG is not considered. If SCR[7] = 1, RX_FIFO_TRIG is 2 LSB of the trigger level (1 to 63 on 6 bits) with the granularity 1. If SCR[7] = 0 and TLR[7] to TLR[4] = 0000, then: 0h (R/W) = 8 characters 1h (R/W) = 16 characters 2h (R/W) = 56 characters 3h (R/W) = 60 characters
5-4	TX_FIFO_TRIG	W	0h	Can be written only if EFR[4] = 1. Sets the trigger level for the TX FIFO: If SCR[6] = 0 and TLR[3] to TLR[0] not equal to 0000, TX_FIFO_TRIG is not considered. If SCR[6] = 1, TX_FIFO_TRIG is 2 LSB of the trigger level (1 to 63 on 6 bits) with a granularity of 1. If SCR[6] = 0 and TLR[3] to TLR[0] = 0000, then: 0h (R/W) = 8 characters 1h (R/W) = 16 characters 2h (R/W) = 32 characters 3h (R/W) = 56 characters
3	DMA_MODE	W	0h	Can be changed only when the baud clock is not running (DLL and DLH cleared to 0). If SCR[0] = 0, this register is considered. 0h (R/W) = DMA_MODE 0 (No DMA). 1h (R/W) = DMA_MODE 1 (UART_NDMA_REQ[0] in TX, UART_NDMA_REQ[1] in RX).
2	TX_FIFO_CLR	W	0h	TX_FIFO_CLEAR. 0h (R/W) = No change. 1h (R/W) = Clears the transmit FIFO and resets its counter logic to 0. Returns to 0 after clearing FIFO.
1	RX_FIFO_CLR	W	0h	RX_FIFO_CLEAR. 0h (R/W) = No change. 1h (R/W) = Clears the receive FIFO and resets its counter logic to 0. Returns to 0 after clearing FIFO.
0	FIFO_EN	W	0h	Can be changed only when the baud clock is not running (DLL and DLH cleared to 0). 0h (R/W) = Disables the transmit and receive FIFOs. The transmit and receive holding registers are 1-byte FIFOs. 1h (R/W) = Enables the transmit and receive FIFOs. The transmit and receive holding registers are 64-byte FIFOs.

21.5.1.12 UART_LCR Register (offset = Ch) [reset = 0h]

UART_LCR is shown in Figure 21-45 and described in Table 21-41.

The line control register (LCR) is selected with a bit register setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. As soon as LCR[6] is set to 1, the TX line is forced to 0 and remains in this state as long as LCR[6] = 1.

Figure 21-45. UART_LCR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
DIV_EN	BREAK_EN	PARITY_TYPE 2	PARITY_TYPE 1	PARITY_EN	NB_STOP	CHAR_LENGTH	
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

Table 21-41. UART_LCR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7	DIV_EN	R/W	0h	Divisor latch enable. 0h (R/W) = Normal operating condition. 1h (R/W) = Divisor latch enable. Allows access to DLL and DLH.
6	BREAK_EN	R/W	0h	Break control bit. Note: When LCR[6] is set to 1, the TX line is forced to 0 and remains in this state as long as LCR[6] = 1. 0h (R/W) = Normal operating condition. 1h (R/W) = Forces the transmitter output to go low to alert the communication terminal.
5	PARITY_TYPE2	R/W	0h	If LCR[3] = 1, then: 0h (R/W) = If LCR[5] = 0, LCR[4] selects the forced parity format. 1h (R/W) = If LCR[5] = 1 and LCR[4] = 0, the parity bit is forced to 1 in the transmitted and received data. If LCR[5] = 1 and LCR[4] = 1, the parity bit is forced to 0 in the transmitted and received data.
4	PARITY_TYPE1	R/W	0h	If LCR[3] = 1, then: 0h (R/W) = Odd parity is generated. 1h (R/W) = Even parity is generated.
3	PARITY_EN	R/W	0h	Parity bit. 0h (R/W) = No parity. 1h (R/W) = A parity bit is generated during transmission, and the receiver checks for received parity.
2	NB_STOP	R/W	0h	Specifies the number of stop bits. 0h (R/W) = 1 stop bit (word length = 5, 6, 7, 8). 1h (R/W) = 1.5 stop bits (word length = 5) or 2 stop bits (word length = 6, 7, 8).
1-0	CHAR_LENGTH	R/W	0h	Specifies the word length to be transmitted or received. 0h (R/W) = 5 bits 1h (R/W) = 6 bits 2h (R/W) = 7 bits 3h (R/W) = 8 bit

21.5.1.13 UART_MCR Register (offset = 10h) [reset = 0h]

UART_MCR is shown in Figure 21-46 and described in Table 21-42.

The modem control register (MCR) is selected with a register bit setting of LCR[7] = 0 or LCR[7] not equal to BFh. MCR[7:5] can only be written when EFR[4] = 1. Bits 3-0 control the interface with the modem, data set, or peripheral device that is emulating the modem.

Figure 21-46. UART_MCR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	TCRTLR	XONEN	LOOPBACKEN	CDSTSCH	RISTSCH	RTS	DTR
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 21-42. UART_MCR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	
6	TCRTLR	R/W	0h	Can be written only when EFR[4] = 1. 0h (R/W) = No action. 1h (R/W) = Enables access to the TCR and TLR registers.
5	XONEN	R/W	0h	Can be written only when EFR[4] = 1. 0h (R/W) = Disable XON any function. 1h (R/W) = Enable XON any function.
4	LOOPBACKEN	R/W	0h	Loopback mode enable. 0h (R/W) = Normal operating mode. 1h (R/W) = Enable local loopback mode (internal). In this mode, the MCR[3:0] signals are looped back into MSR[7:4]. The transmit output is looped back to the receive input internally.
3	CDSTSCH	R/W	0h	CDSTSCH. 0h (R/W) = In loopback mode, forces DCD (active-low) input high and IRQ outputs to INACTIVE state. 1h (R/W) = In loopback mode, forces DCD (active-low) input low and IRQ outputs to INACTIVE state.
2	RISTSCH	R/W	0h	RISTSCH. 0h (R/W) = In loopback mode, forces RI (active-low) input inactive (high). 1h (R/W) = In loopback mode, forces RI (active-low) input active (low).
1	RTS	R/W	0h	In loopback mode, controls MSR[4]. If auto-RTS is enabled, the RTS (active-low) output is controlled by hardware flow control. 0h (R/W) = Force RTS (active-low) output to inactive (high). 1h (R/W) = Force RTS (active-low) output to active (low).
0	DTR	R/W	0h	DTR. 0h (R/W) = Force DTR (active-low) output (used in loopback mode) to inactive (high). 1h (R/W) = Force DTR (active-low) output (used in loopback mode) to active (low).

21.5.1.14 UART_XON1_ADDR1 Register (offset = 10h) [reset = 0h]

UART_XON1_ADDR1 is shown in Figure 21-47 and described in Table 21-43.

The XON1/ADDR1 registers are selected with a register bit setting of LCR[7] = BFh. In UART mode, XON1 character; in IrDA mode, ADDR1 address 1.

Figure 21-47. UART_XON1_ADDR1 Register

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">XONWORD1</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 21-43. UART_XON1_ADDR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-0	XONWORD1	R/W	0h	Stores the 8 bit XON1 character in UART modes and ADDR1 address 1 in IrDA modes.

21.5.1.15 UART_XON2_ADDR2 Register (offset = 14h) [reset = 0h]

UART_XON2_ADDR2 is shown in Figure 21-48 and described in Table 21-44.

The XON2/ADDR2 registers are selected with a register bit setting of LCR[7] = BFh. In UART mode, XON2 character; in IrDA mode, ADDR2 address 2.

Figure 21-48. UART_XON2_ADDR2 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
XONWORD2							
R/W-0h							

Table 21-44. UART_XON2_ADDR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-0	XONWORD2	R/W	0h	Stores the 8 bit XON2 character in UART modes and ADDR2 address 2 in IrDA modes.

21.5.1.16 UART_LSR_CIR Register (offset = 14h) [reset = 81h]

UART_LSR_CIR is shown in Figure 21-49 and described in Table 21-45.

The following line status register (LSR) description is for CIR mode. The CIR LSR is selected with a register bit setting of LCR[7] = 0 or LCR[7] not equal to BFh.

Figure 21-49. UART_LSR_CIR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
THREMPTY	RESERVED	RXSTOP	RESERVED				RXFIFOE
R-1h	R-0h	R-0h	R-0h				R-1h

Table 21-45. UART_LSR_CIR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7	THREMPTY	R	1h	THREMPTY. 0h (R/W) = Transmit holding register (TX FIFO) is not empty. 1h (R/W) = Transmit hold register (TX FIFO) is empty. The transmission is not necessarily completed.
6	RESERVED	R	0h	
5	RXSTOP	R	0h	The RXSTOP is generated based on the value set in the BOF Length register (EBLR). 0h (R/W) = Reception is on going or waiting for a new frame. 1h (R/W) = Reception is completed. It is cleared on a single read of the LSR register.
4-1	RESERVED	R	0h	
0	RXFIFOE	R	1h	RXFIFOE. 0h (R/W) = At least one data character in the RX FIFO. 1h (R/W) = No data in the receive FIFO.

21.5.1.17 UART_LSR_IRDA Register (offset = 14h) [reset = A3h]

UART_LSR_IRDA is shown in Figure 21-50 and described in Table 21-46.

The following line status register (LSR) description is for IrDA mode. The IrDA LSR is selected with a register bit setting of LCR[7] = 0 or LCR[7] not equal to BFh. When the IrDA line status register (LSR) is read, LSR[4:2] reflect the error bits (FL, CRC, ABORT) of the frame at the top of the status FIFO (next frame status to be read).

Figure 21-50. UART_LSR_IRDA Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
THR_EMPTY	STS_FIFO_FULL	RX_LAST_BYTE	FRM_TOO_LONG	ABORT	CRC	STS_FIFO_E	RX_FIFO_E
R-1h	R-0h	R-1h	R-0h	R-0h	R-0h	R-1h	R-1h

Table 21-46. UART_LSR_IRDA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7	THR_EMPTY	R	1h	THR_EMPTY. 0h (R/W) = Transmit holding register (TX FIFO) is not empty. 1h (R/W) = Transmit hold register (TX FIFO) is empty. The transmission is not necessarily completed.
6	STS_FIFO_FULL	R	0h	STS_FIFO_FULL. 0h (R/W) = Status FIFO is not full. 1h (R/W) = Status FIFO is full.
5	RX_LAST_BYTE	R	1h	RX_LAST_BYTE. 0h (R/W) = The RX FIFO (RHR) does not contain the last byte of the frame to be read. 1h (R/W) = The RX FIFO (RHR) contains the last byte of the frame to be read. This bit is set to 1 only when the last byte of a frame is available to be read. It is used to determine the frame boundary. It is cleared on a single read of the LSR register.
4	FRM_TOO_LONG	R	0h	FRAME_TOO_LONG. 0h (R/W) = No frame-too-long error in frame. 1h (R/W) = Frame-too-long error in the frame at the top of the status FIFO (next character to be read). This bit is set to 1 when a frame exceeding the maximum length (set by RXFLH and RXFLL registers) is received. When this error is detected, current frame reception is terminated. Reception is stopped until the next START flag is detected.
3	ABORT	R	0h	ABORT. 0h (R/W) = No abort pattern error in frame. 1h (R/W) = Abort pattern received. SIR and MIR - abort pattern. FIR - Illegal symbol.
2	CRC	R	0h	CRC. 0h (R/W) = No CRC error in frame. 1h (R/W) = CRC error in the frame at the top of the status FIFO (next character to be read).
1	STS_FIFO_E	R	1h	STS_FIFO_E. 0h (R/W) = Status FIFO is not empty. 1h (R/W) = Status FIFO is empty.
0	RX_FIFO_E	R	1h	RX_FIFO_E. 0h (R/W) = At least one data character in the RX FIFO. 1h (R/W) = No data in the receive FIFO.

21.5.1.18 UART_LSR Register (offset = 14h) [reset = 60h]

UART_LSR is shown in Figure 21-51 and described in Table 21-47.

The following line status register (LSR) description is for UART mode. The UART LSR is selected with a register bit setting of LCR[7] = 0 or LCR[7] not equal to BFh. When the UART line status register (LSR) is read, LSR[4:2] reflect the error bits (BI, FE, PE) of the character at the top of the RX FIFO (next character to be read). Therefore, reading the LSR and then reading the RHR identifies errors in a character. Reading RHR updates BI, FE, and PE. LSR [7] is set when there is an error anywhere in the RX FIFO and is cleared only when there are no more errors remaining in the RX FIFO. Reading the LSR does not cause an increment of the RX FIFO read pointer. The RX FIFO read pointer is incremented by reading the RHR. Reading LSR clears OE if set.

Figure 21-51. UART_LSR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RXFIFOSTS	TXSRE	TXFIFOE	RXBI	RXFE	RXPE	RXOE	RXFIFOE
R-0h	R-1h	R-1h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 21-47. UART_LSR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7	RXFIFOSTS	R	0h	RXFIFOSTS. 0h (R/W) = Normal operation. 1h (R/W) = At least one parity error, framing error, or break indication in the RX FIFO. Bit 7 is cleared when no errors are present in the RX FIFO.
6	TXSRE	R	1h	TXSRE. 0h (R/W) = Transmitter hold (TX FIFO) and shift registers are not empty. 1h (R/W) = Transmitter hold (TX FIFO) and shift registers are empty.
5	TXFIFOE	R	1h	TXFIFOE. 0h (R/W) = Transmit hold register (TX FIFO) is not empty. 1h (R/W) = Transmit hold register (TX FIFO) is empty. The transmission is not necessarily completed.
4	RXBI	R	0h	RXBI. 0h (R/W) = No break condition. 1h (R/W) = A break was detected while the data being read from the RX FIFO was being received (RX input was low for one character + 1 bit time frame).
3	RXFE	R	0h	RXFE. 0h (R/W) = No framing error in data being read from RX FIFO. 1h (R/W) = Framing error occurred in data being read from RX FIFO (received data did not have a valid stop bit).
2	RXPE	R	0h	RXPE. 0h (R/W) = No parity error in data being read from RX FIFO. 1h (R/W) = Parity error in data being read from RX FIFO.
1	RXOE	R	0h	RXOE. 0h (R/W) = No overrun error. 1h (R/W) = Overrun error occurred. Set when the character held in the receive shift register is not transferred to the RX FIFO. This case occurs only when receive FIFO is full.

Table 21-47. UART_LSR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	RXFIFOE	R	0h	RXFIFOE. 0h (R/W) = No data in the receive FIFO. 1h (R/W) = At least one data character in the RX FIFO.

21.5.1.19 UART_TCR Register (offset = 18h) [reset = 0h]

UART_TCR is shown in Figure 21-52 and described in Table 21-48.

The transmission control register (TCR) is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. The TCR is accessible only when EFR[4] = 1 and MCR[6] = 1. The transmission control register (TCR) stores the receive FIFO threshold levels to start/stop transmission during hardware flow control. Trigger levels from 0-60 bytes are available with a granularity of 4. Trigger level = 4 x [4-bit register value]. You must ensure that TCR[3:0] > TCR[7:4], whenever auto-RTS or software flow control is enabled to avoid a misoperation of the device. In FIFO interrupt mode with flow control, you have to also ensure that the trigger level to HALT transmission is greater or equal to receive FIFO trigger level (either TLR[7:4] or FCR[7:6]); otherwise, FIFO operation stalls. In FIFO DMA mode with flow control, this concept does not exist because the DMA request is sent each time a byte is received.

Figure 21-52. UART_TCR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RXFIFOTRIGSTART				RXFIFOTRIGHALT			
R/W-0h				R/W-0h			

Table 21-48. UART_TCR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-4	RXFIFOTRIGSTART	R/W	0h	RX FIFO trigger level to RESTORE transmission (0 to 60).
3-0	RXFIFOTRIGHALT	R/W	0h	RX FIFO trigger level to HALT transmission (0 to 60).

21.5.1.20 UART_MSR Register (offset = 18h) [reset = 0h]

UART_MSR is shown in Figure 21-53 and described in Table 21-49.

The modem status register (MSR) is selected with a register bit setting of LCR[7] = 0 or LCR[7] not equal to BFh. The modem status register (MSR) provides information about the current state of the control lines from the modem, data set, or peripheral device to the Local Host. It also indicates when a control input from the modem changes state.

Figure 21-53. UART_MSR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
NCD_STS	NRI_STS	NDSR_STS	NCTS_STS	DCD_STS	RI_STS	DSR_STS	CTS_STS
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 21-49. UART_MSR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7	NCD_STS	R	0h	This bit is the complement of the DCD (active-low) input. In loopback mode, it is equivalent to MCR[3].
6	NRI_STS	R	0h	This bit is the complement of the RI (active-low) input. In loopback mode, it is equivalent to MCR[2].
5	NDSR_STS	R	0h	This bit is the complement of the DSR (active-low) input. In loopback mode, it is equivalent to MCR[0].
4	NCTS_STS	R	0h	This bit is the complement of the CTS (active-low) input. In loopback mode, it is equivalent to MCR[1].
3	DCD_STS	R	0h	DCD_STS. 0h (R/W) = No change. 1h (R/W) = Indicates that DCD (active-low) input (or MCR[3] in loopback mode) has changed. Cleared on a read.
2	RI_STS	R	0h	RI_STS. 0h (R/W) = No change. 1h (R/W) = Indicates that RI (active-low) input (or MCR[2] in loopback mode) changed state from low to high. Cleared on a read.
1	DSR_STS	R	0h	DSR_STS. 0h (R/W) = No change. 1h (R/W) = Indicates that DSR (active-low) input (or MCR[0] in loopback mode) changed state. Cleared on a read.
0	CTS_STS	R	0h	CTS_STS. 0h (R/W) = No change. 1h (R/W) = Indicates that CTS (active-low) input (or MCR[1] in loopback mode) changed state. Cleared on a read.

21.5.1.21 UART_XOFF1 Register (offset = 18h) [reset = 0h]

UART_XOFF1 is shown in Figure 21-54 and described in Table 21-50.

The XOFF1 register is selected with a register bit setting of LCR[7] = BFh. In UART mode, XOFF1 character.

Figure 21-54. UART_XOFF1 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
XOFFWORD1							
R/W-0h							

Table 21-50. UART_XOFF1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-0	XOFFWORD1	R/W	0h	Stores the 8 bit XOFF1 character in UART modes.

21.5.1.22 UART_SPR Register (offset = 1Ch) [reset = 0h]

UART_SPR is shown in Figure 21-55 and described in Table 21-51.

The scratchpad register (SPR) is selected with a register bit setting of LCR[7] = 0 or LCR[7] not equal to BFh. The scratchpad register (SPR) is a read/write register that does not control the module. It is a scratchpad register used to hold temporary data.

Figure 21-55. UART_SPR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
SPR_WORD							
R/W-0h							

Table 21-51. UART_SPR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-0	SPR_WORD	R/W	0h	Scratchpad register.

21.5.1.23 UART_TLR Register (offset = 1Ch) [reset = 0h]

UART_TLR is shown in Figure 21-56 and described in Table 21-52.

The trigger level register is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. The TLR is accessible only when EFR[4] = 1 and MCR[6] = 1. This register stores the programmable transmit and receive FIFO trigger levels used for DMA and IRQ generation.

Figure 21-56. UART_TLR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RX_FIFO_TRIG_DMA				TX_FIFO_TRIG_DMA			
R/W-0h				R/W-0h			

Table 21-52. UART_TLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-4	RX_FIFO_TRIG_DMA	R/W	0h	Receive FIFO trigger level. Following is a summary of settings for the RX FIFO trigger level. SCR[7] = 0, and TLR[7] to TLR[4]=0, then: Defined by FCR[7] and FCR[6] (either 8, 16, 56, 60 characters). SCR[7] = 0, and TLR[7] to TLR[4] not equal to 0000, then: Defined by TLR[7] to TLR[4] (from 4 to 60 characters with a granularity of 4 characters). SCR[7] = 1, and TLR[7] to TLR[4] = any value, then: Defined by the concatenated value of TLR[7] to TLR[4] and FCR[7] and FCR[6] (from 1 to 63 characters with a granularity of 1 character). Note: the combination of TLR[7] to TLR[4] = 0000 and FCR[7] and FCR[6] = 00 (all zeros) is not supported (minimum of 1 character is required). All zeros results in unpredictable behavior.
3-0	TX_FIFO_TRIG_DMA	R/W	0h	Transmit FIFO trigger level. Following is a summary of settings for the TX FIFO trigger level. SCR[6] = 0, and TLR[3] to TLR[0] = 0, then: Defined by FCR[5] and FCR[4] (either 8, 16, 32, 56 characters). SCR[6] = 0, and TLR[3] to TLR[0] not equal to 0000, then: Defined by TLR[3] to TLR[0] (from 4 to 60 characters with a granularity of 4 characters). SCR[6] = 1, and TLR[3] to TLR[0] = any value, then: Defined by the concatenated value of TLR[3] and TLR[0] and FCR[5] and FCR[4] (from 1 to 63 characters with a granularity of 1 character). Note: the combination of TLR[3] to TLR[0] = 0000 and FCR[5] and FCR[4] = 00 (all zeros) is not supported (minimum of 1 character is required). All zeros results in unpredictable behavior.

21.5.1.24 UART_XOFF2 Register (offset = 1Ch) [reset = 0h]

UART_XOFF2 is shown in Figure 21-57 and described in Table 21-53.

The XOFF2 register is selected with a register bit setting of LCR[7] = BFh. In UART mode, XOFF2 character.

Figure 21-57. UART_XOFF2 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
XOFFWORD2							
R/W-0h							

Table 21-53. UART_XOFF2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-0	XOFFWORD2	R/W	0h	Stores the 8 bit XOFF2 character in UART modes.

21.5.1.25 UART_MDR1 Register (offset = 20h) [reset = 7h]

UART_MDR1 is shown in Figure 21-58 and described in Table 21-54.

The mode definition register 1 (MDR1) is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. The mode of operation is programmed by writing to MDR1[2:0]; therefore, the mode definition register 1 (MDR1) must be programmed on startup after configuration of the configuration registers (DLL, DLH, and LCR). The value of MDR1[2:0] must not be changed again during normal operation. If the module is disabled by setting the MODESELECT field to 7h, interrupt requests can still be generated unless disabled through the interrupt enable register (IER). In this case, UART mode interrupts are visible. Reading the interrupt identification register (IIR) shows the UART mode interrupt flags.

Figure 21-58. UART_MDR1 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
FRMENDMODE	SIPMODE	SCT	SETTXIR	IRSLEEP	MODESELECT		
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-7h		

Table 21-54. UART_MDR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7	FRMENDMODE	R/W	0h	IrDA mode only. 0h (R/W) = Frame-length method. 1h (R/W) = Set EOT bit method.
6	SIPMODE	R/W	0h	MIR/FIR modes only. 0h (R/W) = Manual SIP mode: SIP is generated with the control of ACREG[3]. 1h (R/W) = Automatic SIP mode: SIP is generated after each transmission.
5	SCT	R/W	0h	Store and control the transmission. 0h (R/W) = Starts the infrared transmission when a value is written to the THR register. 1h (R/W) = Starts the infrared transmission with the control of ACREG[2]. Note: Before starting any transmission, there must be no reception ongoing.
4	SETTXIR	R/W	0h	Used to configure the infrared transceiver. 0h (R/W) = If MDR2[7] = 0, no action; if MDR2[7] = 1, TXIR pin output is forced low. 1h (R/W) = TXIR pin output is forced high (not dependant of MDR2[7] value).
3	IRSLEEP	R/W	0h	IrDA/CIR sleep mode. 0h (R/W) = IrDA/CIR sleep mode disabled. 1h (R/W) = IrDA/CIR sleep mode enabled.
2-0	MODESELECT	R/W	7h	UART/IrDA/CIR mode selection. 0h (R/W) = UART 16x mode. 1h (R/W) = SIR mode. 2h (R/W) = UART 16x auto-baud. 3h (R/W) = UART 13x mode. 4h (R/W) = MIR mode. 5h (R/W) = FIR mode. 6h (R/W) = CIR mode. 7h (R/W) = Disable (default state).

21.5.1.26 UART_MDR2 Register (offset = 24h) [reset = 0h]

UART_MDR2 is shown in Figure 21-59 and described in Table 21-55.

The mode definition register 2 (MDR2) is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. The MDR2[0] bit describes the status of the TX status interrupt in IIR[5]. The IRTXUNDERRUN bit must be read after a TX status interrupt occurs. The MDR2[2:1] bits set the trigger level for the frame status FIFO (8 entries) and must be programmed before the mode is programmed in MDR1[2:0]. The MDR2[6] bit gives the flexibility to invert the RX pin inside the UART module to ensure that the protocol at the input of the transceiver module has the same polarity at module level. By default, the RX pin is inverted because most of transceiver invert the IR receive pin.

Figure 21-59. UART_MDR2 Register

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>SETTXIRALT</td><td>IRRXINVERT</td><td colspan="2">CIRPULSEMODE</td><td>UARTPULSE</td><td colspan="2">STSFIFOTRIG</td><td>IRTXUNDERRUN</td></tr>
<tr><td>R/W-0h</td><td>R/W-0h</td><td colspan="2">R/W-0h</td><td>R/W-0h</td><td colspan="2">R/W-0h</td><td>R-0h</td></tr>
</table>

Table 21-55. UART_MDR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7	SETTXIRALT	R/W	0h	Provides alternate functionality for MDR1[4]. 0h (R/W) = Normal mode 1h (R/W) = Alternate mode for SETTXIR
6	IRRXINVERT	R/W	0h	Only for IR mode (IrDA and CIR). Invert RX pin in the module before the voting or sampling system logic of the infrared block. This does not affect the RX path in UART modem modes. 0h (R/W) = Inversion is performed. 1h (R/W) = No inversion is performed.
5-4	CIRPULSEMODE	R/W	0h	CIR pulse modulation definition. Defines high level of the pulse width associated with a digit: 0h (R/W) = Pulse width of 3 from 12 cycles. 1h (R/W) = Pulse width of 4 from 12 cycles. 2h (R/W) = Pulse width of 5 from 12 cycles. 3h (R/W) = Pulse width of 6 from 12 cycles.
3	UARTPULSE	R/W	0h	UART mode only. Used to allow pulse shaping in UART mode. 0h (R/W) = Normal UART mode. 1h (R/W) = UART mode with pulse shaping.
2-1	STSFIFOTRIG	R/W	0h	Only for IrDA mode. Frame status FIFO threshold select: 0h (R/W) = 1 entry 1h (R/W) = 4 entries 2h (R/W) = 7 entries 3h (R/W) = 8 entries
0	IRTXUNDERRUN	R	0h	IrDA transmission status interrupt. When the TX status interrupt (IIR[5]) occurs, the meaning of the interrupt is: 0h (R/W) = The last bit of the frame was transmitted successfully without error. 1h (R/W) = An underrun occurred. The last bit of the frame was transmitted but with an underrun error. The bit is reset to 0 when the RESUME register is read.

21.5.1.27 UART_TXFLL Register (offset = 28h) [reset = 0h]

UART_TXFLL is shown in Figure 21-60 and described in Table 21-56.

The transmit frame length low register (TXFLL) is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. The transmit frame length low register (TXFLL) and the TXFLH register hold the 13-bit transmit frame length (expressed in bytes). TXFLL holds the LSBs and TXFLH holds the MSBs. The frame length value is used if the frame length method of frame closing is used.

Figure 21-60. UART_TXFLL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXFLL							
W-0h							

Table 21-56. UART_TXFLL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-0	TXFLL	W	0h	LSB register used to specify the frame length.

21.5.1.28 UART_SFLSR Register (offset = 28h) [reset = 0h]

UART_SFLSR is shown in Figure 21-61 and described in Table 21-57.

The status FIFO line status register (SFLSR) is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. Reading the status FIFO line status register (SFLSR) effectively reads frame status information from the status FIFO. This register does not physically exist. Reading this register increments the status FIFO read pointer (SFREGL and SFREGH must be read first). Top of RX FIFO = Next frame to be read from RX FIFO.

Figure 21-61. UART_SFLSR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED			OE_ERROR	FRM_TOO_LONG_ERROR	ABORT_DETECT	CRC_ERROR	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h

Table 21-57. UART_SFLSR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-5	RESERVED	R	0h	
4	OE_ERROR	R	0h	OE_ERROR. 0h (R/W) = No error 1h (R/W) = Overrun error in RX FIFO when frame at top of RX FIFO was received.
3	FRM_TOO_LONG_ERROR	R	0h	FRAME_TOO_LONG_ERROR. 0h (R/W) = No error 1h (R/W) = Frame-length too long error in frame at top of RX FIFO.
2	ABORT_DETECT	R	0h	ABORT_DETECT. 0h (R/W) = No error 1h (R/W) = Abort pattern detected in frame at top of RX FIFO.
1	CRC_ERROR	R	0h	CRC_ERROR. 0h (R/W) = No error 1h (R/W) = CRC error in frame at top of RX FIFO.
0	RESERVED	R	0h	

21.5.1.29 UART_RESUME Register (offset = 2Ch) [reset = 0h]

UART_RESUME is shown in Figure 21-62 and described in Table 21-58.

The RESUME register is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. The RESUME register is used to clear internal flags, which halt transmission/reception when an underrun/overrun error occurs. Reading this register resumes the halted operation. This register does not physically exist and always reads as 00.

Figure 21-62. UART_RESUME Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESUME							
R-0h							

Table 21-58. UART_RESUME Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-0	RESUME	R	0h	Dummy read to restart the TX or RX, value 0 to FFh.

21.5.1.30 UART_TXFLH Register (offset = 2Ch) [reset = 0h]

UART_TXFLH is shown in Figure 21-63 and described in Table 21-59.

The transmit frame length high register (TXFLH) is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. The transmit frame length high register (TXFLH) and the TXFLL register hold the 13-bit transmit frame length (expressed in bytes). TXFLL holds the LSBs and TXFLH holds the MSBs. The frame length value is used if the frame length method of frame closing is used.

Figure 21-63. UART_TXFLH Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED			TXFLH				
R-0h			W-0h				

Table 21-59. UART_TXFLH Register Field Descriptions

Bit	Field	Type	Reset	Description
15-5	RESERVED	R	0h	
4-0	TXFLH	W	0h	MSB register used to specify the frame length, value 0 to 1Fh.

21.5.1.31 UART_RXFLL Register (offset = 30h) [reset = 0h]

UART_RXFLL is shown in Figure 21-64 and described in Table 21-60.

The received frame length low register (RXFLL) is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. The received frame length low register (RXFLL) and the RXFLH register hold the 12-bit receive maximum frame length. RXFLL holds the LSBs and RXFLH holds the MSBs. If the intended maximum receive frame length is n bytes, program RXFLL and RXFLH to be n + 3 in SIR or MIR modes and n + 6 in FIR mode (+3 and +6 are the result of frame format with CRC and stop flag; two bytes are associated with the FIR stop flag).

Figure 21-64. UART_RXFLL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RXFLL							
W-0h							

Table 21-60. UART_RXFLL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-0	RXFLL	W	0h	LSB register used to specify the frame length in reception, value 0 to FFh.

21.5.1.32 UART_SFREGL Register (offset = 30h) [reset = 0h]

UART_SFREGL is shown in Figure 21-65 and described in Table 21-61.

The status FIFO register low (SFREGL) is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. The frame lengths of received frames are written into the status FIFO. This information can be read by reading the status FIFO register low (SFREGL) and the status FIFO register high (SFREGH). These registers do not physically exist. The LSBs are read from SFREGL and the MSBs are read from SFREGH. Reading these registers does not alter the status FIFO read pointer. These registers must be read before the pointer is incremented by reading the SFLSR.

Figure 21-65. UART_SFREGL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
SFREGL							
R-0h							

Table 21-61. UART_SFREGL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-0	SFREGL	R	0h	LSB part of the frame length, value 0 to FFh.

21.5.1.33 UART_SFREGH Register (offset = 34h) [reset = 0h]

UART_SFREGH is shown in Figure 21-66 and described in Table 21-62.

The status FIFO register high (SFREGH) is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. The frame lengths of received frames are written into the status FIFO. This information can be read by reading the status FIFO register low (SFREGL) and the status FIFO register high (SFREGH). These registers do not physically exist. The LSBs are read from SFREGL and the MSBs are read from SFREGH. Reading these registers does not alter the status FIFO read pointer. These registers must be read before the pointer is incremented by reading the SFLSR.

Figure 21-66. UART_SFREGH Register

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">SFREGH</td></tr>
<tr><td colspan="4">R-0h</td><td colspan="4">R-0h</td></tr>
</table>

Table 21-62. UART_SFREGH Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	
3-0	SFREGH	R	0h	MSB part of the frame length, value 0 to Fh.

21.5.1.34 UART_RXFLH Register (offset = 34h) [reset = 0h]

UART_RXFLH is shown in Figure 21-67 and described in Table 21-63.

The received frame length high register (RXFLH) is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. The received frame length high register (RXFLH) and the RXFLL register hold the 12-bit receive maximum frame length. RXFLL holds the LSBs and RXFLH holds the MSBs. If the intended maximum receive frame length is n bytes, program RXFLL and RXFLH to be n + 3 in SIR or MIR modes and n + 6 in FIR mode (+3 and +6 are the result of frame format with CRC and stop flag; two bytes are associated with the FIR stop flag).

Figure 21-67. UART_RXFLH Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED				RXFLH			
R-0h				W-0h			

Table 21-63. UART_RXFLH Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	
3-0	RXFLH	W	0h	MSB register used to specify the frame length in reception, value 0 to Fh.

21.5.1.35 UART_BLR Register (offset = 38h) [reset = 40h]

UART_BLR is shown in Figure 21-68 and described in Table 21-64.

The BOF control register (BLR) is selected with a register bit setting of LCR[7] = 0. The BLR[6] bit is used to select whether C0h or FFh start patterns are to be used, when multiple start flags are required in SIR mode. If only one start flag is required, this is always C0h. If n start flags are required, either (n - 1) C0h or (n -1) FFh flags are sent, followed by a single C0h flag (immediately preceding the first data byte).

Figure 21-68. UART_BLR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STSFIFORESET	XBOFTYPE	RESERVED					
R/W-0h	R/W-1h	R-0h					

Table 21-64. UART_BLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7	STSFIFORESET	R/W	0h	Status FIFO reset. This bit is self-clearing.
6	XBOFTYPE	R/W	1h	SIR xBOF select. 0h (R/W) = FFh start pattern is used. 1h (R/W) = C0h start pattern is used.
5-0	RESERVED	R	0h	

21.5.1.36 UART_UASR Register (offset = 38h) [reset = 0h]

UART_UASR is shown in Figure 21-69 and described in Table 21-65.

The UART autobauding status register (UASR) is selected with a register bit setting of LCR[7] not equal to BFh or LCR[7] = BFh. The UART autobauding status register (UASR) returns the speed, the number of bits by characters, and the type of parity in UART autobauding mode. In autobauding mode, the input frequency of the UART modem must be fixed to 48 MHz. Any other module clock frequency results in incorrect baud rate recognition. This register is used to set up transmission according to characteristics of previous reception, instead of LCR, DLL, and DLH registers when UART is in autobauding mode. To reset the autobauding hardware (to start a new AT detection) or to set the UART in standard mode (no autobaud), MDR1[2:0] must be set to 7h (reset state), then set to 2h (UART in autobaud mode) or cleared to 0 (UART in standard mode). Usage limitation: Only 7 and 8 bits character (5 and 6 bits not supported). 7 bits character with space parity not supported. Baud rate between 1200 and 115 200 bp/s (10 possibilities).

Figure 21-69. UART_UASR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
PARITYTYPE		BITBYCHAR	SPEED				
R-0h		R-0h	R-0h				

Table 21-65. UART_UASR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-6	PARITYTYPE	R	0h	Type of the parity in UART autobauding mode. 0h (R/W) = No parity identified 1h (R/W) = Parity space 2h (R/W) = Even parity 3h (R/W) = Odd parity
5	BITBYCHAR	R	0h	Number of bits by characters. 0h (R/W) = 7-bit character identified. 1h (R/W) = 8-bit character identified.
4-0	SPEED	R	0h	Speed. 0h (R/W) = No speed identified. 1h (R/W) = 115 200 baud 2h (R/W) = 57 600 baud 3h (R/W) = 38 400 baud 4h (R/W) = 28 800 baud 5h (R/W) = 19 200 baud 6h (R/W) = 14 400 baud 7h (R/W) = 9600 baud 8h (R/W) = 4800 baud 9h (R/W) = 2400 baud Ah (R/W) = 1200 baud Bh (R/W) = Reserved from Bh to 1Fh.

21.5.1.37 UART_ACREG Register (offset = 3Ch) [reset = 0h]

UART_ACREG is shown in Figure 21-70 and described in Table 21-66.

The auxiliary control register (ACREG) is selected with a register bit setting of LCR[7] = 0. If transmit FIFO is not empty and MDR1[5] = 1, IrDA starts a new transfer with data of previous frame as soon as abort frame has been sent. Therefore, TX FIFO must be reset before sending an abort frame. It is recommended to disable TX FIFO underrun capability by masking corresponding underrun interrupt. When disabling underrun by setting ACREG[4] = 1, unknown data is sent over TX line.

Figure 21-70. UART_ACREG Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
PULSETYPE	SDMOD	DISIRRX	DISTXUNDERRUN	SENDSIP	SCTXEN	ABORTEN	EOTEN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 21-66. UART_ACREG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7	PULSETYPE	R/W	0h	SIR pulse-width select: 0h (R/W) = 3/16 of baud-rate pulse width 1h (R/W) = 1.6 microseconds
6	SDMOD	R/W	0h	Primary output used to configure transceivers. Connected to the SD/MODE input pin of IrDA transceivers. 0h (R/W) = SD pin is set to high. 1h (R/W) = SD pin is set to low.
5	DISIRRX	R/W	0h	Disable RX input. 0h (R/W) = Normal operation (RX input automatically disabled during transmit, but enabled outside of transmit operation). 1h (R/W) = Disables RX input (permanent state; independent of transmit).
4	DISTXUNDERRUN	R/W	0h	Disable TX underrun. 0h (R/W) = Long stop bits cannot be transmitted. TX underrun is enabled. 1h (R/W) = Long stop bits can be transmitted.
3	SENDSIP	R/W	0h	MIR/FIR modes only. Send serial infrared interaction pulse (SIP). If this bit is set during an MIR/FIR transmission, the SIP is sent at the end of it. This bit is automatically cleared at the end of the SIP transmission. 0h (R/W) = No action. 1h (R/W) = Send SIP pulse.
2	SCTXEN	R/W	0h	Store and control TX start. When MDR1[5] = 1 and the LH writes 1 to this bit, the TX state-machine starts frame transmission. This bit is self-clearing.
1	ABORTEN	R/W	0h	Frame abort. The LH can intentionally abort transmission of a frame by writing 1 to this bit. Neither the end flag nor the CRC bits are appended to the frame.
0	EOTEN	R/W	0h	EOT (end-of-transmission) bit. The LH writes 1 to this bit just before it writes the last byte to the TX FIFO in the set-EOT bit frame-closing method. This bit is automatically cleared when the LH writes to the THR (TX FIFO).

21.5.1.38 UART_SCR Register (offset = 40h) [reset = 0h]

UART_SCR is shown in Figure 21-71 and described in Table 21-67.

The supplementary control register (SCR) is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. Bit 4 enables the wake-up interrupt, but this interrupt is not mapped into the IIR register. Therefore, when an interrupt occurs and there is no interrupt pending in the IIR register, the SSR[1] bit must be checked. To clear the wake-up interrupt, bit SCR[4] must be reset to 0.

Figure 21-71. UART_SCR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RXTRIGGRANU1	TXTRIGGRANU1	DSRIT	RXCTSDSRWAKEUPEN	TXEMPTYCTLIT	DMAMODE2		DMAMODECTL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h

Table 21-67. UART_SCR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7	RXTRIGGRANU1	R/W	0h	RXTRIGGRANU1 0h (R/W) = Disables the granularity of 1 for trigger RX level. 1h (R/W) = Enables the granularity of 1 for trigger RX level.
6	TXTRIGGRANU1	R/W	0h	TXTRIGGRANU1 0h (R/W) = Disables the granularity of 1 for trigger TX level. 1h (R/W) = Enables the granularity of 1 for trigger TX level.
5	DSRIT	R/W	0h	DSRIT 0h (R/W) = Disables DSR (active-low) interrupt. 1h (R/W) = Enables DSR (active-low) interrupt.
4	RXCTSDSRWAKEUPEN	R/W	0h	RX CTS wake-up enable. 0h (R/W) = Disables the WAKE UP interrupt and clears SSR[1]. 1h (R/W) = Waits for a falling edge of RX, CTS (active-low), or DSR (active-low) pins to generate an interrupt.
3	TXEMPTYCTLIT	R/W	0h	TXEMPTYCTLIT 0h (R/W) = Normal mode for THR interrupt. 1h (R/W) = THR interrupt is generated when TX FIFO and TX shift register are empty.
2-1	DMAMODE2	R/W	0h	Specifies the DMA mode valid if SCR[0] = 1, then: 0h (R/W) = DMA mode 0 (no DMA). 1h (R/W) = DMA mode 1 (UARTnDMAREQ[0] in TX, UARTnDMAREQ[1] in RX) 2h (R/W) = DMA mode 2 (UARTnDMAREQ[0] in RX) 3h (R/W) = DMA mode 3 (UARTnDMAREQ[0] in TX)
0	DMAMODECTL	R/W	0h	DMAMODECT 0h (R/W) = The DMAMODE is set with FCR[3]. 1h (R/W) = The DMAMODE is set with SCR[2:1].

21.5.1.39 UART_SSR Register (offset = 44h) [reset = 4h]

UART_SSR is shown in Figure 21-72 and described in Table 21-68.

The supplementary status register (SSR) is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. Bit 1 is reset only when SCR[4] is reset to 0.

Figure 21-72. UART_SSR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED					DMACTRRST	RXCTSDSRWAKEUPSTS	TXFIFOFULL
R-0h					R/W-1h	R-0h	R-0h

Table 21-68. UART_SSR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-3	RESERVED	R	0h	
2	DMACTRRST	R/W	1h	DMACOUNTERRST. 0h (R/W) = The DMA counter will not be reset, if the corresponding FIFO is reset (via FCR[1] or FCR[2]). 1h (R/W) = The DMA counter will be reset, if the corresponding FIFO is reset (via FCR[1] or FCR[2]).
1	RXCTSDSRWAKEUPSTS	R	0h	Pin falling edge detection: Reset only when SCR[4] is reset to 0. 0h (R/W) = No falling-edge event on RX, CTS (active-low), and DSR (active-low). 1h (R/W) = A falling edge occurred on RX, CTS (active-low), or DSR (active-low).
0	TXFIFOFULL	R	0h	TXFIFOFULL. 0h (R/W) = TX FIFO is not full. 1h (R/W) = TX FIFO is full.

21.5.1.40 UART_EBLR Register (offset = 48h) [reset = 0h]

UART_EBLR is shown in Figure 21-73 and described in Table 21-69.

The BOF length register (EBLR) is selected with a register bit setting of LCR[7] = 0. In IrDA SIR operation, the BOF length register (EBLR) specifies the number of BOF + xBOFs to transmit. The value set into this register must consider the BOF character; therefore, to send only one BOF with no XBOF, this register must be set to 1. To send one BOF with n XBOFs, this register must be set to n + 1. Furthermore, the value 0 sends 1 BOF plus 255 XBOFs. In IrDA MIR mode, the BOF length register (EBLR) specifies the number of additional start flags (MIR protocol mandates a minimum of 2 start flags). In CIR mode, the BOF length register (EBLR) specifies the number of consecutive zeros to be received before generating the RXSTOP interrupt (IIR[2]). All the received zeros are stored in the RX FIFO. When the register is cleared to 0, this feature is deactivated and always in reception state, which is disabled by setting the ACREG[5] bit to 1. If the RX_STOP interrupt occurs before a byte boundary, the remaining bits of the last byte are filled with zeros and then passed into the RX FIFO.

Figure 21-73. UART_EBLR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
EBLR							
R/W-0h							

Table 21-69. UART_EBLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-0	EBLR	R/W	0h	IrDA mode: This register allows definition of up to 176 xBOFs, the maximum required by IrDA specification. CIR mode: This register specifies the number of consecutive zeros to be received before generating the RXSTOP interrupt (IIR[2]). 0h (R/W) = Feature disabled. 1h (R/W) = Generate RXSTOP interrupt after receiving 1 zero bit. FFh (R/W) = Generate RXSTOP interrupt after receiving 255 zero bits.

21.5.1.41 UART_MVR Register (offset = 50h) [reset = 0h]

UART_MVR is shown in Figure 21-74 and described in Table 21-70.

The module version register (MVR) is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. The reset value is fixed by hardware and corresponds to the RTL revision of this module. A reset has no effect on the value returned.

Figure 21-74. UART_MVR Register

15	14	13	12	11	10	9	8
RESERVED					MAJORREV		
R-0h					R-0h		

7	6	5	4	3	2	1	0
RESERVED		MINORREV					
R-0h		R-0h					

Table 21-70. UART_MVR Register Field Descriptions

Bit	Field	Type	Reset	Description
16-11	RESERVED	R	0h	
10-8	MAJORREV	R	0h	Major revision number of the module.
7-6	RESERVED	R	0h	
5-0	MINORREV	R	0h	Minor revision number of the module.

21.5.1.42 UART_SYSC Register (offset = 54h) [reset = 0h]

UART_SYSC is shown in Figure 21-75 and described in Table 21-71.

The system configuration register (SYSC) is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. The AUTOIDLE bit controls a power-saving technique to reduce the logic power consumption of the module interface; that is, when the feature is enabled, the interface clock is gated off until the module interface is accessed. When the SOFTRESET bit is set high, it causes a full device reset.

Figure 21-75. UART_SYSC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED			IDLEMODE		ENAWAKEUP	SOFTRESET	AUTOIDLE
R-0h			R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 21-71. UART_SYSC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-5	RESERVED	R	0h	
4-3	IDLEMODE	R/W	0h	Power management req/ack control. 0h (R/W) = Force idle: Idle request is acknowledged unconditionally. 1h (R/W) = No-idle: Idle request is never acknowledged. 2h (R/W) = Smart idle: Acknowledgement to an idle request is given based in the internal activity of the module. 3h (R/W) = Smart idle Wakeup: Acknowledgement to an idle request is given based in the internal activity of the module. The module is allowed to generate wakeup request. Only available on UART0.
2	ENAWAKEUP	R/W	0h	Wakeup control. 0h (R/W) = Wakeup is disabled. 1h (R/W) = Wakeup capability is enabled.
1	SOFTRESET	R/W	0h	Software reset. Set this bit to 1 to trigger a module reset. This bit is automatically reset by the hardware. Read returns 0. 0h (R/W) = Normal mode. 1h (R/W) = Module is reset.
0	AUTOIDLE	R/W	0h	Internal interface clock-gating strategy. 0h (R/W) = Clock is running. 1h (R/W) = Reserved.

21.5.1.43 UART_SYSS Register (offset = 58h) [reset = 0h]

UART_SYSS is shown in Figure 21-76 and described in Table 21-72.

The system status register (SYSS) is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh.

Figure 21-76. UART_SYSS Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							RESETDONE
R-0h							R-0h

Table 21-72. UART_SYSS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R	0h	
0	RESETDONE	R	0h	Internal reset monitoring. 0h (R/W) = Internal module reset is ongoing. 1h (R/W) = Reset complete.

21.5.1.44 UART_WER Register (offset = 5Ch) [reset = FFh]

UART_WER is shown in Figure 21-77 and described in Table 21-73.

The wake-up enable register (WER) is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. The wake-up enable register (WER) is used to mask and unmask a UART event that subsequently notifies the system. An event is any activity in the logic that can cause an interrupt and/or an activity that requires the system to wake up. Even if wakeup is disabled for certain events, if these events are also an interrupt to the UART, the UART still registers the interrupt as such.

Figure 21-77. UART_WER Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
TXWAKEUPEN	RLS__INTR	RHR__INTR	RX__ACTIVITY	DCD_ACTIVITY	RI__ACTIVITY	DSR_ACTIVITY	CTS__ACTIVITY
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 21-73. UART_WER Register Field Descriptions

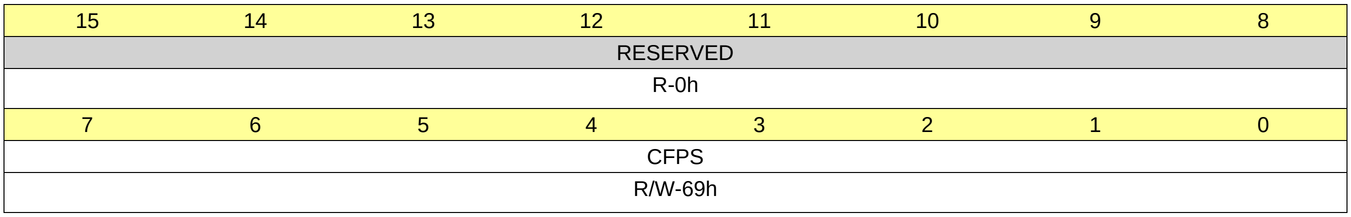
Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7	TXWAKEUPEN	R/W	1h	Wake-up interrupt. 0h (R/W) = Event is not allowed to wake up the system. 1h (R/W) = Event can wake up the system: Event can be: THRIT or TXDMA request and/or TXSATUSIT.
6	RLS__INTR	R/W	1h	Receiver line status interrupt. 0h (R/W) = Event is not allowed to wake up the system. 1h (R/W) = Event can wake up the system.
5	RHR__INTR	R/W	1h	RHR interrupt. 0h (R/W) = Event is not allowed to wake up the system. 1h (R/W) = Event can wake up the system.
4	RX__ACTIVITY	R/W	1h	RX_ACTIVITY. 0h (R/W) = Event is not allowed to wake up the system. 1h (R/W) = Event can wake up the system.
3	DCD_ACTIVITY	R/W	1h	DCD_ACTIVITY. 0h (R/W) = Event is not allowed to wake up the system. 1h (R/W) = Event can wake up the system.
2	RI__ACTIVITY	R/W	1h	RI_ACTIVITY. 0h (R/W) = Event is not allowed to wake up the system. 1h (R/W) = Event can wake up the system.
1	DSR_ACTIVITY	R/W	1h	DSR_ACTIVITY. 0h (R/W) = Event is not allowed to wake up the system. 1h (R/W) = Event can wake up the system.
0	CTS__ACTIVITY	R/W	1h	CTS_ACTIVITY. 0h (R/W) = Event is not allowed to wake up the system. 1h (R/W) = Event can wake up the system.

21.5.1.45 UART_CFPS Register (offset = 60h) [reset = 69h]

UART_CFPS is shown in Figure 21-78 and described in Table 21-74.

The carrier frequency prescaler register (CFPS) is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. Since the consumer IR (CIR) works at modulation rates of 30-56.8 kHz, the 48 MHz clock must be prescaled before the clock can drive the IR logic. The carrier frequency prescaler register (CFPS) sets the divisor rate to give a range to accommodate the remote control requirements in BAUD multiples of 12x. The value of the CFPS at reset is 105 decimal (69h), which equates to a 38.1 kHz output from starting conditions. The 48 MHz carrier is prescaled by the CFPS that is then divided by the 12x BAUD multiple.

Figure 21-78. UART_CFPS Register



15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
CFPS							
R/W-69h							

Table 21-74. UART_CFPS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-0	CFPS	R/W	69h	System clock frequency prescaler at (12x multiple). CFPS = 0 is not supported. Examples for CFPS values follow. Target Frequency (kHz) = 30, CFPS (decimal) = 133, Actual Frequency (kHz) = 30.08. Target Frequency (kHz) = 32.75, CFPS (decimal) = 122, Actual Frequency (kHz) = 32.79. Target Frequency (kHz) = 36, CFPS (decimal) = 111, Actual Frequency (kHz) = 36.04. Target Frequency (kHz) = 36.7, CFPS (decimal) = 109, Actual Frequency (kHz) = 36.69. Target Frequency (kHz) = 38, CFPS (decimal) = 105, Actual Frequency (kHz) = 38.1. Target Frequency (kHz) = 40, CFPS (decimal) = 100, Actual Frequency (kHz) = 40. Target Frequency (kHz) = 56.8, CFPS (decimal) = 70, Actual Frequency (kHz) = 57.14.

21.5.1.46 UART_RXFIFO_LVL Register (offset = 64h) [reset = 0h]

UART_RXFIFO_LVL is shown in Figure 21-79 and described in Table 21-75.

Figure 21-79. UART_RXFIFO_LVL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RXFIFO_LVL							
R-0h							

Table 21-75. UART_RXFIFO_LVL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-0	RXFIFO_LVL	R	0h	Level of the RX FIFO

21.5.1.47 UART_TXFIFO_LVL Register (offset = 68h) [reset = 0h]

UART_TXFIFO_LVL is shown in Figure 21-80 and described in Table 21-76.

Figure 21-80. UART_TXFIFO_LVL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXFIFO_LVL							
R-0h							

Table 21-76. UART_TXFIFO_LVL Register Field Descriptions

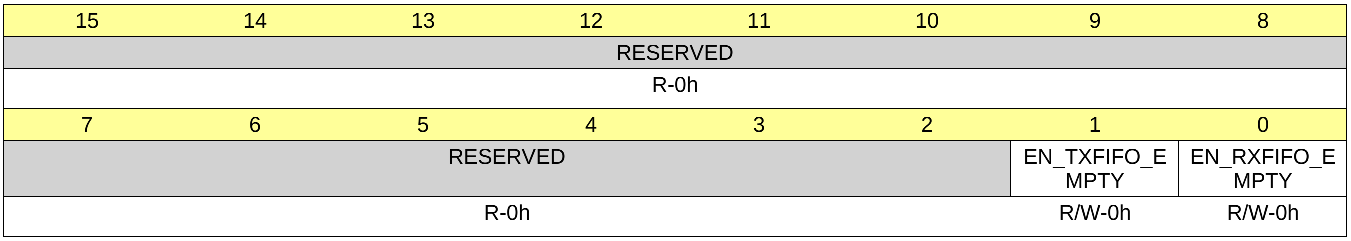
Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-0	TXFIFO_LVL	R	0h	Level of the TX FIFO

21.5.1.48 UART_IER2 Register (offset = 6Ch) [reset = 0h]

UART_IER2 is shown in Figure 21-81 and described in Table 21-77.

The IER2 enables RX/TX FIFOs empty corresponding interrupts.

Figure 21-81. UART_IER2 Register



15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						EN_TXFIFO_EMPTY	EN_RXFIFO_EMPTY
R-0h						R/W-0h	R/W-0h

Table 21-77. UART_IER2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-2	RESERVED	R	0h	
1	EN_TXFIFO_EMPTY	R/W	0h	EN_TXFIFO_EMPTY. 0h (R/W) = Disables EN_TXFIFO_EMPTY interrupt. 1h (R/W) = Enables EN_TXFIFO_EMPTY interrupt.
0	EN_RXFIFO_EMPTY	R/W	0h	Number of bits by characters. 0h (R/W) = Disables EN_RXFIFO_EMPTY interrupt. 1h (R/W) = Enables EN_RXFIFO_EMPTY interrupt.

21.5.1.49 UART_ISR2 Register (offset = 70h) [reset = 0h]

UART_ISR2 is shown in Figure 21-82 and described in Table 21-78.

The interrupt status register 2 (ISR2) is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. The ISR2 displays the status of RX/TX FIFOs empty corresponding interrupts.

Figure 21-82. UART_ISR2 Register

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>TXFIFO_EMPTY_STS</td><td>RXFIFO_EMPTY_STS</td></tr>
<tr><td colspan="6">R-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 21-78. UART_ISR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-2	RESERVED	R	0h	
1	TXFIFO_EMPTY_STS	R/W	0h	TXFIFO_EMPTY_STS. 0h (R/W) = TXFIFO_EMPTY interrupt not pending. 1h (R/W) = TXFIFO_EMPTY interrupt pending.
0	RXFIFO_EMPTY_STS	R/W	0h	RXFIFO_EMPTY_STS. 0h (R/W) = RXFIFO_EMPTY interrupt not pending. 1h (R/W) = RXFIFO_EMPTY interrupt pending.

21.5.1.50 UART_FREQ_SEL Register (offset = 74h) [reset = 0h]

UART_FREQ_SEL is shown in Figure 21-83 and described in Table 21-79.

Figure 21-83. UART_FREQ_SEL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FREQ_SEL							
R/W-0h							

Table 21-79. UART_FREQ_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	
7-0	FREQ_SEL	R/W	0h	Sets the sample per bit if non default frequency is used. MDR3[1] must be set to 1 after this value is set. Must be equal or higher then 6.

21.5.1.51 UART_MDR3 Register (offset = 80h) [reset = 0h]

UART_MDR3 is shown in Figure 21-84 and described in Table 21-80.

The mode definition register 3 (MDR3) is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh. The DISABLE_CIR_RX_DEMOD register bit will force the CIR receiver to bypass demodulation of received data if set. See the CIR Mode Block Components. The NONDEFAULT_FREQ register bit allows the user to set sample per bit by writing it into FREQ_SEL register. Set it if non-default (48 MHz) fclk frequency is used to achieve a less than 2% error rate. Changing this bit (to any value) will automatically disable the device by setting MDR[2:0] to 111 .

Figure 21-84. UART_MDR3 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					SET_DMA_TX_THR	NONDEFAULT_FREQ	DISABLE_CIR_RX_DEMOD
R-0h					R/W-0h	R/W-0h	R/W-0h

Table 21-80. UART_MDR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-3	RESERVED	R	0h	
2	SET_DMA_TX_THR	R/W	0h	SET_DMA_TX_THRESHOLD. 0h (R/W) = Disable use of TX DMA Threshold register. Use 64-TX trigger as DMA threshold. 1h (R/W) = Enable to set different TX DMA threshold in the TX DMA Threshold register.
1	NONDEFAULT_FREQ	R/W	0h	NONDEFAULT_FREQ. 0h (R/W) = Disables using NONDEFAULT fclk frequencies. 1h (R/W) = Enables using NONDEFAULT fclk frequencies (set FREQ_SEL and DLH/DLL).
0	DISABLE_CIR_RX_DEMOD	R/W	0h	DISABLE_CIR_RX_DEMOD. 0h (R/W) = Enables CIR RX demodulation. 1h (R/W) = Disables CIR RX demodulation.

21.5.1.52 UART_TX_DMA_THR Register (offset = 84h) [reset = 0h]

UART_TX_DMA_THR is shown in Figure 21-85 and described in Table 21-81.

The TX DMA threshold register is selected with a register bit setting of LCR[7] = 0, LCR[7] not equal to BFh, or LCR[7] = BFh.

Figure 21-85. UART_TX_DMA_THR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		TX_DMA_THR					
R-0h		R/W-0h					

Table 21-81. UART_TX_DMA_THR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-6	RESERVED	R	0h	
5-0	TX_DMA_THR	R/W	0h	Used to manually set the TX DMA threshold level. UART_MDR3[2] SET_TX_DMA_THRESHOLD must be 1 and must be value + tx_trigger_level = 64 (TX FIFO size). If not, 64_tx_trigger_level will be used without modifying the value of this register.

I2C

This chapter describes the I2C of the device.

Topic **Page**

22.1 Introduction

The multi-master I2C peripheral provides an interface between a CPU and any I2C-bus-compatible device that connects via the I2C serial bus. External components attached to the I2C bus can serially transmit/receive up to 8-bit data to/from the CPU device through the two-wire I2C interface.

The I2C bus is a multi-master bus. The I2C controller does support the multi-master mode that allows more than one device capable of controlling the bus to be connected to it. Each I2C device is recognized by a unique address and can operate as either transmitter or receiver, according to the function of the device. In addition to being a transmitter or receiver, a device connected to the I2C bus can also be considered as master or slave when performing data transfers. Note that a master device is the device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. During this transfer, any device addressed by this master is considered a slave.

22.1.1 I2C Features

The general features of the I2C controller are:

- Compliant with Philips I2C specification version 2.1
- Supports standard mode (up to 100K bits/s) and fast mode (up to 400K bits/s).
- Multimaster transmitter/slave receiver mode
- Multimaster receiver/slave transmitter mode
- Combined master transmit/receive and receive/transmit modes
- 7-bit and 10-bit device addressing modes
- Built-in 32-byte FIFO for buffered read or writes in each module
- Programmable clock generation
- Two DMA channels, one interrupt line

22.1.2 Unsupported I2C Features

The I2C module features not supported in this device are shown in Table 22-1.

Table 22-1. Unsupported I2C Features

Feature	Reason
SCCB Protocol	SCCB signal not pinned out
High Speed (3.4 MBPS) operation	Not supported

22.2 Integration

This device includes three instantiations of the I2C module. This peripheral implements the multi-master I2C bus which allows serial transfer of 8-bit data to/from other I2C master/slave devices through a two-wire interface. There are three I2C modules instantiations called I2C0, I2C1, and I2C2. The I2C0 module is located in the Wake-up power domain. Figure 22-1 and Figure 22-2 show examples of a system with multiple I2C-compatible devices.

Figure 22-1. I2C0 Integration and Bus Application

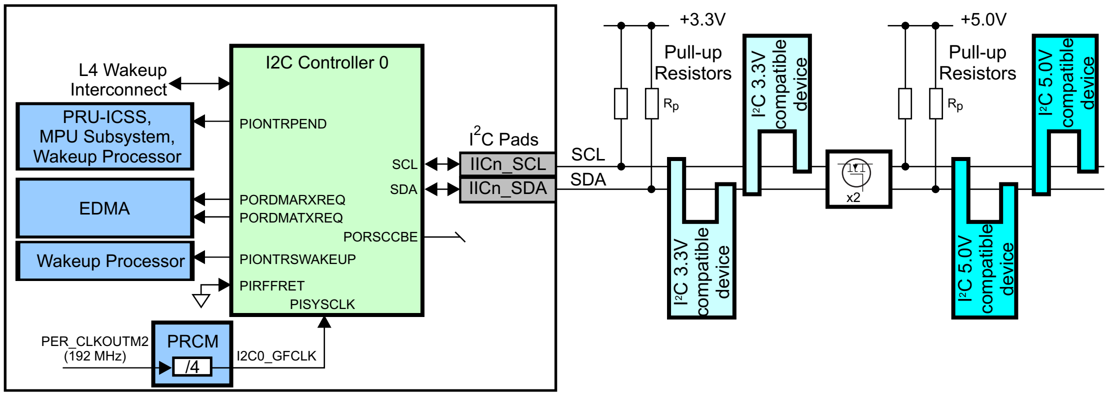
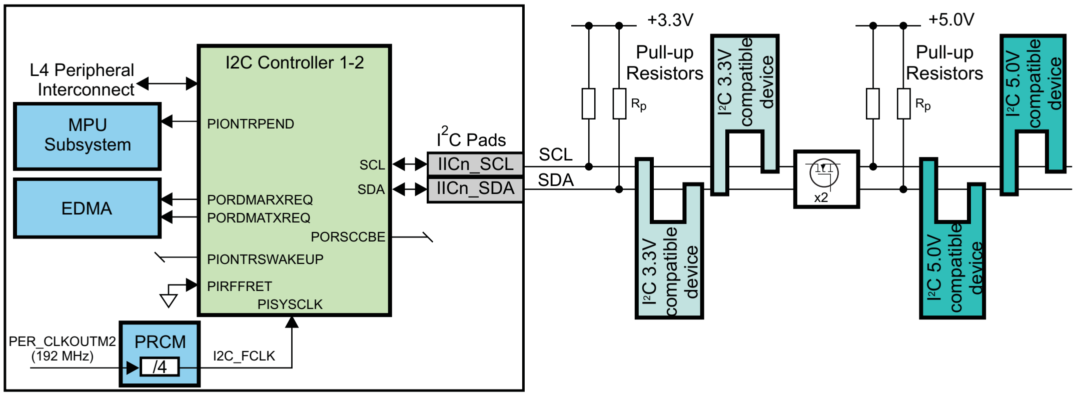


Figure 22-2. I2C(1–2) Integration and Bus Application



22.2.1 *I2C Connectivity Attributes*

The general connectivity attributes for the I2C module are shown in Table 22-2 and Table 22-3.

Table 22-2. I2C0 Connectivity Attributes

Attributes	Type
Power Domain	Wakeup Domain
Clock Domain	PD_WKUP_L4_WKUP_GCLK (Interface/OCP) PD_WKUP_I2C0_GFCLK (Func)
Reset Signals	WKUP_DOM_RST_N
Idle/Wakeup Signals	Smart Idle / Wakeup
Interrupt Requests	1 interrupt to MPU Subsystem (I2C0INT), PRU-ICSS, and Wakeup Processor
DMA Requests	2 DMA requests to EDMA (I2CTXEVT0, I2CRXEVT0)
Physical Address	L4 Wakeup slave port

Table 22-3. I2C(1–2) Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L4LS_GCLK (Interface/OCP) PD_PER_I2C_FCLK (Func)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	1 interrupt per instance to MPU Subsystem (I2C1INT, I2C2INT)
DMA Requests	2 DMA requests per instance to EDMA (I2CTXEVTx, I2CRXEVTx)
Physical Address	L4 Peripheral slave port

22.2.2 I2C Clock and Reset Management

The I2C controllers have separate bus interface and functional clocks. During power-down mode, the I2C*x*_SCL and I2C*x*_SDA are configured as inputs.

Table 22-4. I2C Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
I2C0 Clock Signals			
PIOCPCLK Interface clock	26 MHz	CLK_M_OSC	pd_wkup_l4_wkup_gclk From PRCM
PISYSCLK Functional clock	48 MHz	PER_CLKOUTM2 / 4	pd_wkup_i2c0_gfclk From PRCM
I2C(1-2) Clock Signals			
PIOCPCLK Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk From PRCM
PISYSCLK Functional clock	48 MHz	PER_CLKOUTM2 / 4	pd_per_ic2_fclk From PRCM

22.2.3 I2C Pin List

The external signals (I2C*x*_SDA, I2C*x*_SCL) on the device use standard LVCMOS I/Os and may not meet full compliance with the I2C specifications for Fast-mode devices for slope control and input filtering (spike suppression) to improve the EMC behavior.

Table 22-5. I2C Pin List

Pin	Type	Description
I2C*x*_SCL	I/OD(1)	I2C serial clock (open drain)
I2C*x*_SDA	I/OD	I2C serial data (open drain)

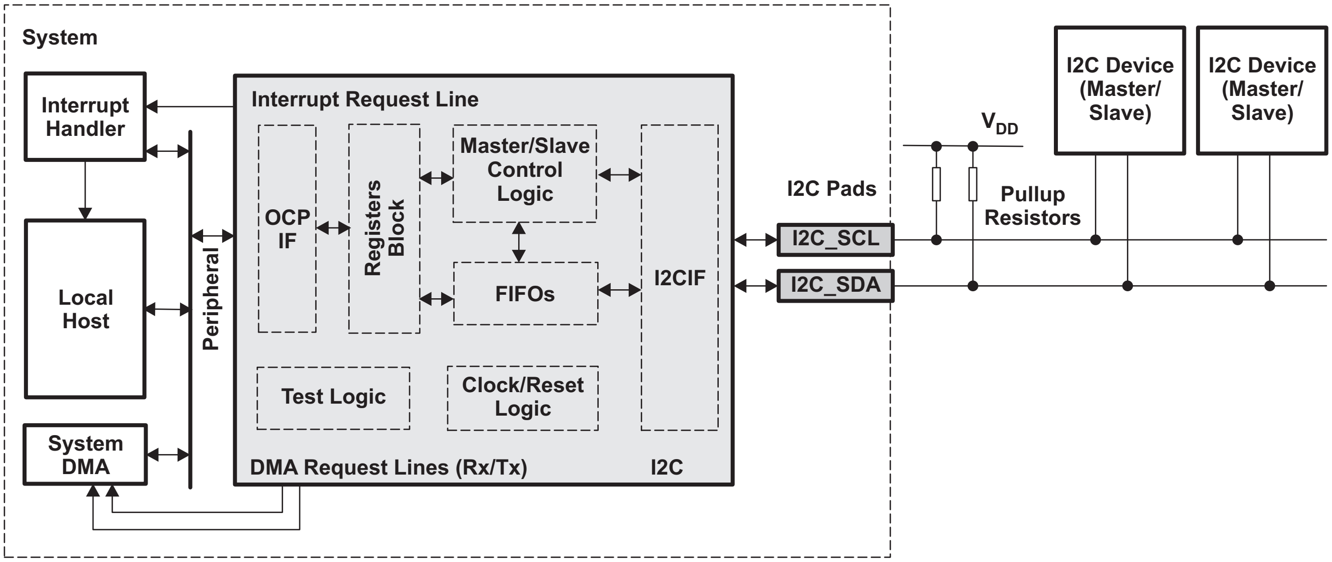
(1) These signals are also used as inputs to re-time or sync data. The associated CONF_<module>_pin_RXACTIVE bit for these signals must be set to 1 to enable the inputs back to the module. It is also recommended to place a 33-ohm resistor in series (close to the processor) on each of these signals to avoid signal reflections.

22.3 Functional Description

22.3.1 Functional Block Diagram

Figure 22-3 shows an example of a system with multiple I2C compatible devices in which the I2C serial ports are all connected together for a two-way transfer from one device to other devices.

Figure 22-3. I2C Functional Block Diagram



The I2C peripheral consists of the following primary blocks:

- A serial interface: one data pin (I2C_SDA) and one clock pin (I2C_SCL).
- Data registers to temporarily hold receive data and transmit data traveling between the I2C_SDA pin and the CPU or the DMA controller.
- Control and status registers
- A peripheral data bus interface to enable the CPU and the DMA controller to access the I2C peripheral registers.
- A clock synchronizer to synchronize the I2C input clock (from the processor clock generator) and the clock on the I2C_SCL pin, and to synchronize data transfers with masters of different clock speeds.
- A prescaler to divide down the input clock that is driven to the I2C peripheral
- A noise filter on each of the two pins, I2C_SDA and I2C_SCL
- An arbitrator to handle arbitration between the I2C peripheral (when it is a master) and another master
- Interrupt generation logic, so that an interrupt can be sent to the CPU
- DMA event generation logic to send an interrupt to the CPU upon reception and data transmission of data.

22.3.2 I2C Master/Slave Contoller Signals

Data is communicated to devices interfacing with the I2C via the serial data line (SDA) and the serial clock line (SCL). These two wires can carry information between a device and others connected to the I2C bus. Both SDA and SCL are bi-directional pins. They must be connected to a positive supply voltage via a pull-up resistor. When the bus is free, both pins are high. The driver of these two pins has an open drain to perform the required wired-AND function.

An example of multiple I2C modules that are connected for a two-way transfer from one device to other devices is shown in Figure 22-4.

Figure 22-4. Multiple I2C Modules Connected

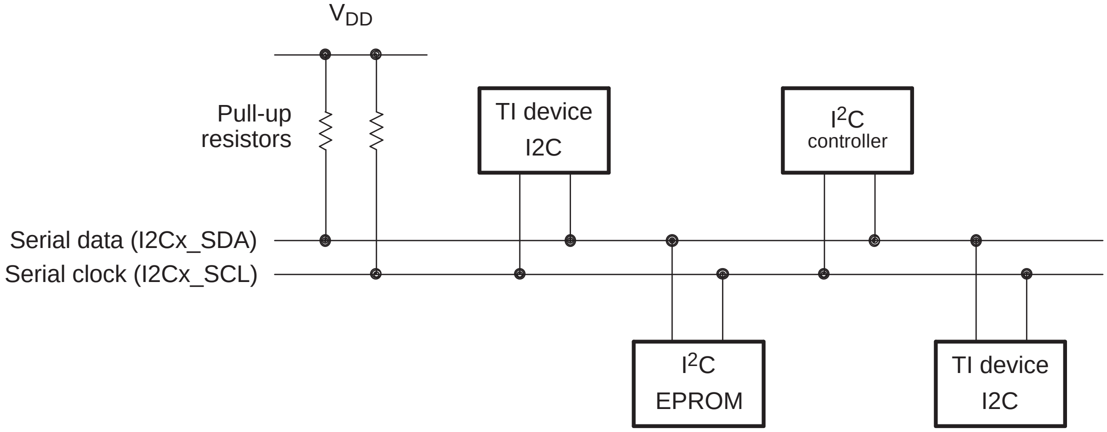


Table 22-6. Signal Pads

	I2C Mode	
Name	**Default Operating Mode**	**Description**
I2C_SCL	In/ Out	I2C serial CLK line Open-drain output buffer. Requires external pull-up resistor (Rp).
I2C_SDA	In/ Out	I2C serial data line Open-drain output buffer. Requires external pull-up resistor (Rp).

22.3.3 I2C Reset

The I2C module can be reset in the following three ways:

- A system reset (PIRSTNA = 0). A device reset causes the system reset. All registers are reset to power up reset values.
- A software reset by setting the SRST bit in the I2C_SYSC register. This bit has exactly the same action on the module logic as the system bus reset. All registers are reset to power up reset values.
- The I2C_EN bit in the I2C_CON register can be used to hold the I2C module in reset. When the system bus reset is removed (PIRSTNA = 1), I2C_EN = 0 keeps the functional part of I2C module in reset state and all configuration registers can be accessed. I2C_EN = 0 does not reset the registers to power up reset values.

Table 22-7. Reset State of I2C Signals

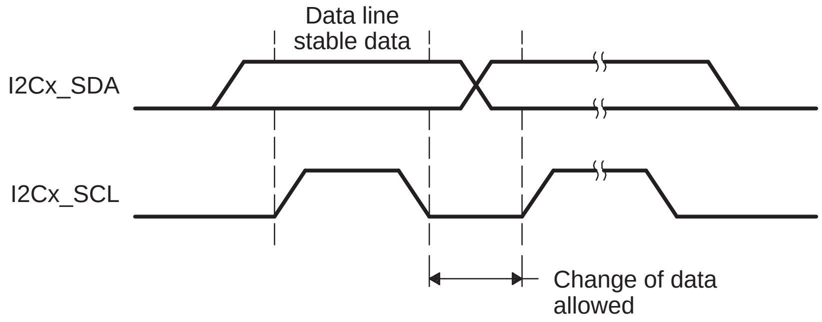
			I2C Reset
Pin	**I/O/Z** (1)	**System Reset**	**(I2C_EN = 0)**
SDA	I/O/Z	High impedance	High impedance
SCL	I/O/Z	High impedance	High impedance

(1) I = Input, O = Outpu, Z = High impedance

22.3.4 Data Validity

The data on the SDA line must be stable during the high period of the clock. The high and low states of the data line can only change when the clock signal on the SCL line is LOW.

Figure 22-5. Bit Transfer on the I2C Bus

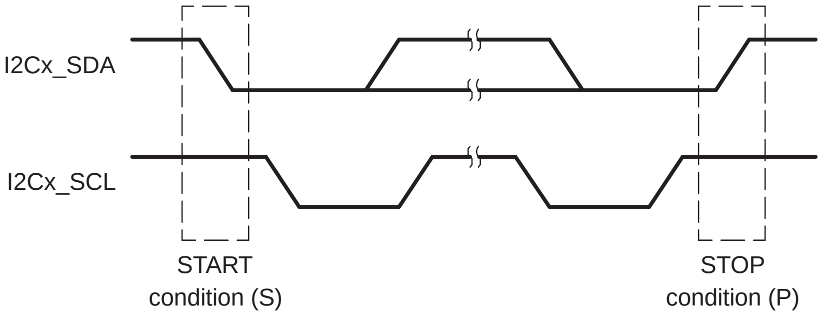


22.3.5 START & STOP Conditions

The I2C module generates START and STOP conditions when it is configured as a master.

- START condition is a high-to-low transition on the SDA line while SCL is high.
- STOP condition is a low-to-high transition on the SDA line while SCL is high.
- The bus is considered to be busy after the START condition (BB = 1) and free after the STOP condition (BB = 0).

Figure 22-6. Start and Stop Condition Events

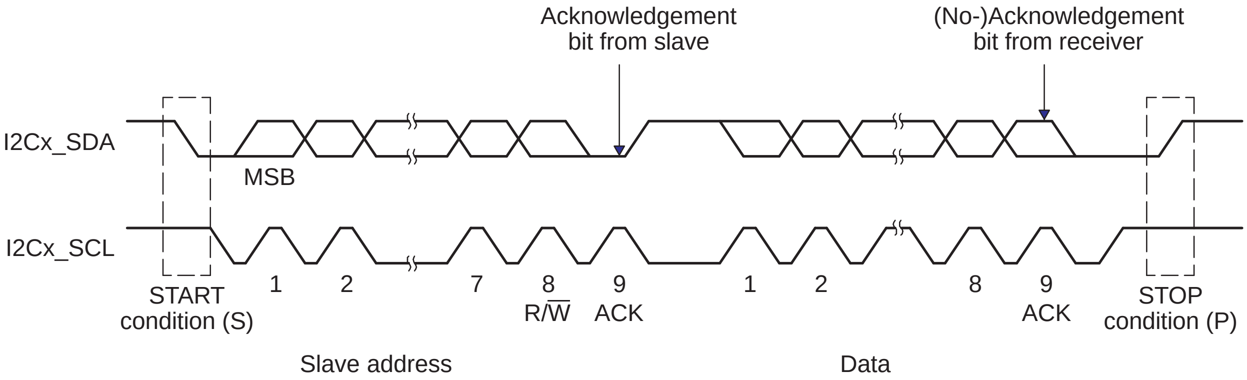


22.3.6 I2C Operation

22.3.6.1 Serial Data Formats

The I2C controller operates in 8-bit word data format (byte write access supported for the last access). Each byte put on the SDA line is 8 bits long. The number of bytes that can be transmitted or received is restricted by the value programmed in the DCOUNT register. The data is transferred with the most significant bit (MSB) first. Each byte is followed by an acknowledge bit from the module if it is in receiver mode.

Figure 22-7. I2C Data Transfer



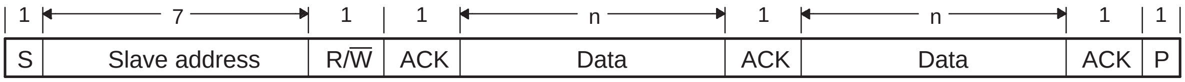
The I2C module supports two data formats, as shown in Figure 22-8:

- 7-bit/10-bit addressing format
- 7-bit/10-bit addressing format with repeated start condition

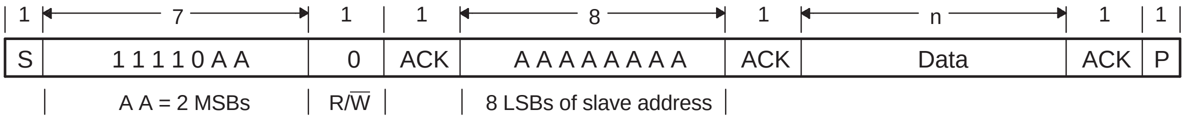
The first byte after a start condition (S) always consists of 8 bits. In the acknowledge mode, an extra bit dedicated for acknowledgment is inserted after each byte. In the addressing formats with 7-bit addresses, the first byte is composed of 7 MSB slave address bits and 1 LSB R/nW bit. In the addressing formats with 10-bit addresses, the first byte is composed of 7 MSB slave address bits, such as 11110XX, where XX is the two MSB of the 10-bit addresses, and 1 LSB R/nW bit, which is 0 in this case.

The least significant R/nW of the address byte indicates the direction of transmission of the following data bytes. If R/nW is 0, the master writes data into the selected slave; if it is 1, the master reads data out of the slave.

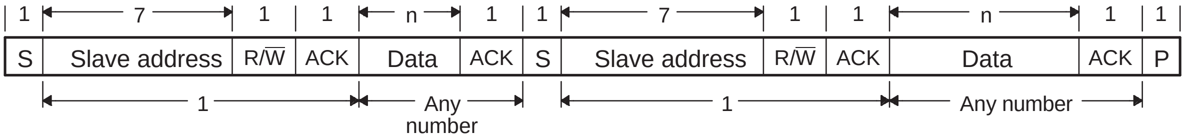
Figure 22-8. I2C Data Transfer Formats



7-Bit Addressing Format



10-Bit Addressing Format



7-Bit Addressing Format With Repeated START Condition

22.3.6.2 Master Transmitter

In this mode, data assembled in one of the previously described data formats is shifted out on the serial data line SDA in synch with the self-generated clock pulses on the serial clock line SCL. The clock pulses are inhibited and SCL held low when the intervention of the processor is required (XUDF) after a byte has been transmitted.

22.3.6.3 Master Receiver

This mode can only be entered from the master transmitter mode. With either of the address formats (Figure 22-8 (a), (b), and (c)), the master receiver is entered after the slave address byte and bit R/W_ has been transmitted, if R/W_ is high. Serial data bits received on bus line SDA are shifted in synch with the self-generated clock pulses on SCL. The clock pulses are inhibited and SCL held low when the intervention of the processor is required (ROVR) after a byte has been transmitted. At the end of a transfer, it generates the stop condition.

22.3.6.4 Slave Transmitter

This mode can only be entered from the slave receiver mode. With either of the address formats (Figure 22-8 (a), (b), and (c)), the slave transmitter is entered if the slave address byte is the same as its own address and bit R/W_ has been transmitted, if R/W_ is high. The slave transmitter shifts the serial data out on the data line SDA in synch with the clock pulses that are generated by the master device. It does not generate the clock but it can hold clock line SCL low while intervention of the CPU is required (XUDF).

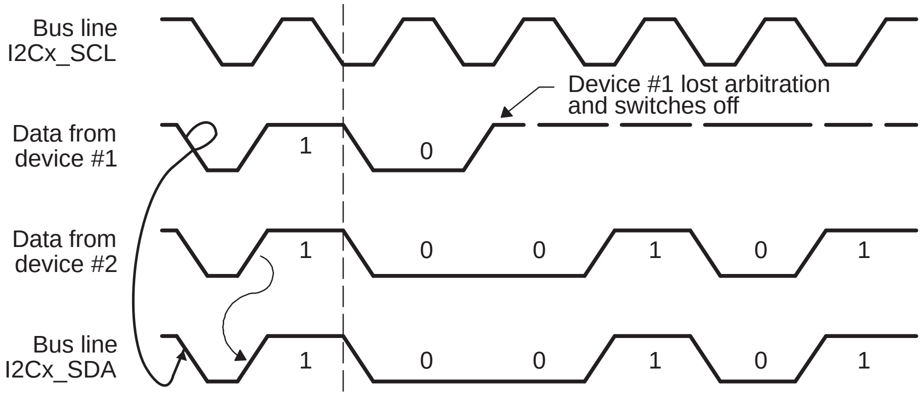
22.3.6.5 Slave Receiver

In this mode, serial data bits received on the bus line SDA are shifted-in in synch with the clock pulses on SCL that are generated by the master device. It does not generate the clock but it can hold clock line SCL low while intervention of the CPU is required (ROVR) following the reception of a byte.

22.3.7 Arbitration

If two or more master transmitters start a transmission on the same bus almost simultaneously, an arbitration procedure is invoked. The arbitration procedure uses the data presented on the serial bus by the competing transmitters. When a transmitter senses that a high signal it has presented on the bus has been overruled by a low signal, it switches to the slave receiver mode, sets the arbitration lost (AL) flag, and generates the arbitration lost interrupt. Figure 22-9 shows the arbitration procedure between two devices. The arbitration procedure gives priority to the device that transmits the serial data stream with the lowest binary value. Should two or more devices send identical first bytes, arbitration continues on the subsequent bytes.

Figure 22-9. Arbitration Procedure Between Two Master Transmitters



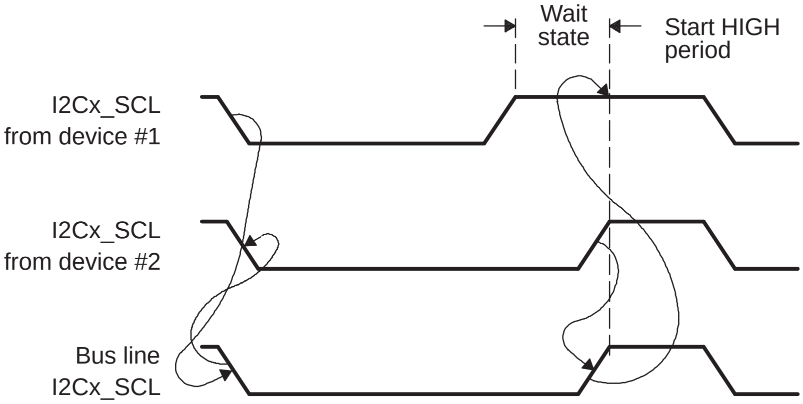
22.3.8 I2C Clock Generation and I2C Clock Synchronization

Under normal conditions, only one master device generates the clock signal, SCL. During the arbitration procedure, however, there are two or more master devices and the clock must be synchronized so that the data output can be compared. The wired-AND property of the clock line means that a device that first generates a low period of the clock line overrules the other devices. At this high/low transition, the clock generators of the other devices are forced to start generation of their own low period. The clock line is then held low by the device with the longest low period, while the other devices that finish their low periods must wait for the clock line to be released before starting their high periods. A synchronized signal on the clock line is thus obtained, where the slowest device determines the length of the low period and the fastest the length of the high period.

If a device pulls down the clock line for a longer time, the result is that all clock generators must enter the WAIT-state. In this way a slave can slow down a fast master and the slow device can create enough time to store a received byte or to prepare a byte to be transmitted (Clock Stretching). Figure 22-10 illustrates the clock synchronization.

Note: If the SCL or SDA lines are stuck low, the Bus Clear operation is supported. If the clock line (SCL) is stuck low, the preferred procedure is to reset the bus using the hardware reset signal if your I2C devices have hardware reset inputs. If the I2C devices do not have hardware reset inputs, cycle power to the devices to activate the mandatory internal power-on reset (POR) circuit. If the data line (SDA) is stuck low, the master should send nine clock pulses. The device that held the bus low should release it sometime within those nine clocks. If not, use the hardware reset or cycle power to clear the bus.

Figure 22-10. Synchronization of Two I2C Clock Generators



22.3.9 Prescaler (SCLK/ICLK)

The I2C module is operated with a functional clock (SCLK) frequency that can be in a range of 12-100 MHz, according to I2C mode that must be used (an internal ~24 MHz clock (ICLK) is recommended in case of F/S operation mode e). Note that the frequency of the functional clock influences directly the I2C bus performance and timings.

The internal clock used for I2C logic - ICLK - is generated via the I2C prescaler block. The prescaler consists of a 4-bit register - I2C _PSC, and is used to divide the system clock (SCLK) to obtain the internal required clock for the I2C module.

22.3.10 Noise Filter

The noise filter is used to suppress any noise that is 50 ns or less, in the case of F/S mode of operation. It is designed to suppress noise with one ICLK. The noise filter is always one ICLK cycle, regardless of the bus speed. For FS mode (prescaler = 4, ICLK = 24 MHz), the maximum width of the suppressed spikes is 41.6 ns. To ensure a correct filtering, the prescaler must be programmed accordingly.

22.3.11 I2C Interrupts

The I2C module generates 12 types of interrupt: addressed as slave, bus free (stop condition detected), access error, start condition, arbitration-lost, noacknowledge, general call, registers-ready-for-access, receive and transmit data, receive and transmit draining. These 12 interrupts are accompanied with 12 interrupt masks and flags defined in the I2C_IRQEN_SET and respectively I2C_IRQSTS_RAW registers. Note that all these 12 interrupt events are sharing the same hardware interrupt line.

- Addressed As Slave interrupt (AAS) is generated to inform the Local Host that an external master addressed the module as a slave. When this interrupt occurs, the CPU can check the I2C_ACTOA status register to check which of the 4 own addresses was used by the external master to access the module.
- Bus Free interrupt (BF) is generated to inform the Local Host that the I2C bus became free (when a Stop Condition is detected on the bus) and the module can initiate his own I2C transaction.
- Start Condition interrupt (STC) is generated after the module being in idle mode have detected (synchronously or asynchronously) a possible Start Condition on the bus (signalized with WakeUp).
- Access Error interrupt (AERR) is generated if a Data read access is performed while RX FIFO is empty or a Data write access is performed while TX FIFO is full.
- Arbitration lost interrupt (AL) is generated when the I2C arbitration procedure is lost.
- No-acknowledge interrupt (NACK) is generated when the master I2C does not receive acknowledge from the receiver.
- General call interrupt (GC) is generated when the device detects the address of all zeros (8 bits).
- Registers-ready-for-access interrupt (ARDY) is generated by the I2C when the previously programmed address, data, and command have been performed and the status bits have been updated. This

interrupt is used to let the CPU know that the I2C registers are ready for access.

- Receive interrupt/status (RRDY) is generated when there is received data ready to be read by the CPU from the I2C_DATA register (for a complete description of required conditions for interrupt generation, see Section 22.3.14, *FIFO Management*). The CPU can alternatively poll this bit to read the received data from the I2C_DATA register.
- Transmit interrupt/status (XRDY) is generated when the CPU needs to put more data in the I2C_DATA register after the transmitted data has been shifted out on the SDA pin (for a complete description of required conditions for interrupt generation, see Section 22.3.14, *FIFO Management*). The CPU can alternatively poll this bit to write the next transmitted data into the I2C_DATA register.
- Receive draining interrupt (RDR) is generated when the transfer length is not a multiple of threshold value, to inform the CPU that it can read the amount of data left to be transferred and to enable the draining mechanism (see Section 22.3.14.4, *Draining Feature*, for additional details).
- Transmit draining interrupt (XDR) is generated when the transfer length is not a multiple of threshold value, to inform the CPU that it can read the amount of data left to be written and to enable the draining mechanism (see Section 22.3.14.4, *Draining Feature*, for additional details).

When the interrupt signal is activated, the Local Host must read the I2C_IRQSTS_RAW register to define the type of the interrupt, process the request, and then write into these registers the correct value to clear the interrupt flag.

22.3.12 DMA Events

The I2C module can generate two DMA requests events, read (I2C_DMA_RX) and write (I2C_DMA_TX) that can be used by the DMA controller to synchronously read received data from the I2C_DATA or write transmitted data to the I2C_DATA register. The DMA read and write requests are generated in a similar manner as RRDY and XRDY, respectively.

The I2C DMA request signals (I2C_DMA_TX and I2C_DMA_RX) are activated according to Section 22.3.14, *FIFO Management*.

22.3.13 Interrupt and DMA Events

I2C has two DMA channels (Tx and Rx).

I2C has one interrupt line for all the interrupt requests.

For the event and interrupt numbers, see the device-specific datasheet.

22.3.14 FIFO Management

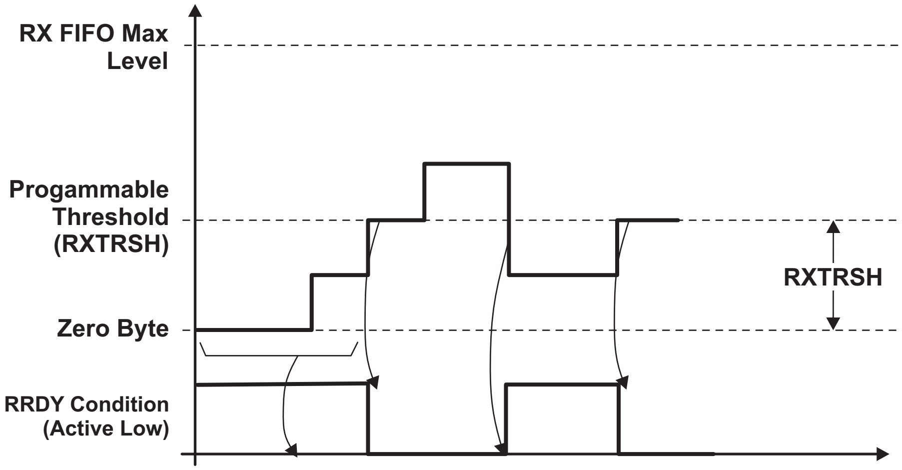
The I2C module implements two internal 32-bytes FIFOs with dual clock for RX and TX modes. The depth of the FIFOs can be configured at integration via a generic parameter which will also be reflected in I2C_IRQSTS_RAW.FIFODEPTH register.

22.3.14.1 FIFO Interrupt Mode Operation

In FIFO interrupt mode (relevant interrupts enabled via I2C_IRQEN_SET register), the processor is informed of the status of the receiver and transmitter by an interrupt signal. These interrupts are raised when receive/transmit FIFO threshold (defined by I2C_BUF.TXTRSH or I2C_BUF.RXTRSH) are reached; the interrupt signals instruct the Local Host to transfer data to the destination (from the I2C module in receive mode and/or from any source to the I2C FIFO in transmit mode).

Figure 22-11 and Figure 22-12, respectively, illustrate receive and transmit operations from FIFO management point of view.

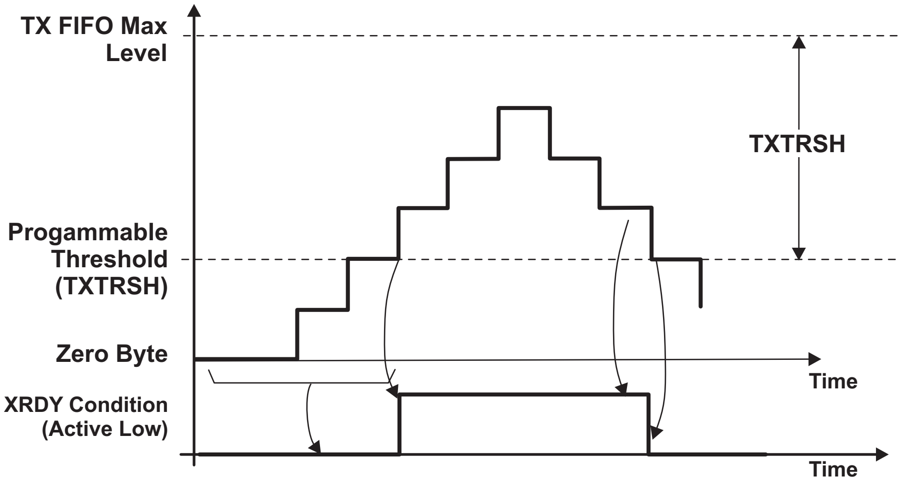
Figure 22-11. Receive FIFO Interrupt Request Generation



Note that in Figure 22-11, the RRDY Condition illustrates that the condition for generating a RRDY interrupt is achieved. The interrupt request is generated when this signal is active, and it can be cleared only by the CPU by writing a 1 in the corresponding interrupt flag. If the condition is still present after clearing the previous interrupt, another interrupt request will be generated.

In receive mode, RRDY interrupt is not generated until the FIFO reaches its receive threshold. Once low, the interrupt can only be de-asserted when the Local Host has handled enough bytes to make the FIFO level below threshold. For each interrupt, the Local Host can be configured to read an amount of bytes equal with the value of the RX FIFO threshold + 1.

Figure 22-12. Transmit FIFO Interrupt Request Generation



Note that in Figure 22-12, the XRDY Condition illustrates that the condition for generating a XRDY interrupt is achieved. The interrupt request is generated when this condition is achieved (when TX FIFO is empty, or the TX FIFO threshold is not reached and there are still data bytes to be transferred in the TX FIFO), and it can be cleared only by the CPU by writing a 1 in the corresponding interrupt flag after transmitting the configured number of bytes. If the condition is still present after clearing the previous interrupt, another interrupt request will be generated.

Note that in interrupt mode, the module offers two options for the CPU application to handle the interrupts:

- When detecting an interrupt request (XRDY or RRDY type), the CPU can write/read one data byte to/from the FIFO and then clear the interrupt. The module will not reassert the interrupt until the interrupt condition is not met.

- When detecting an interrupt request (XRDY or RRDY type), the CPU can be programmed to write/read the amount of data bytes specified by the corresponding FIFO threshold (I2C_BUF.TXTRSH + 1 or I2C_BUF.RXTRSH + 1). In this case, the interrupt condition will be cleared and the next interrupt will be asserted again when the XRDY or RRDY condition will be again met.

If the second interrupt serving approach is used, an additional mechanism (draining feature) is implemented for the case when the transfer length is not a multiple of FIFO threshold (see Section 22.3.14.4, *Draining Feature*).

In slave TX mode, the draining feature cannot be used, since the transfer length is not known at the configuration time, and the external master can end the transfer at any point by not acknowledging one data byte.

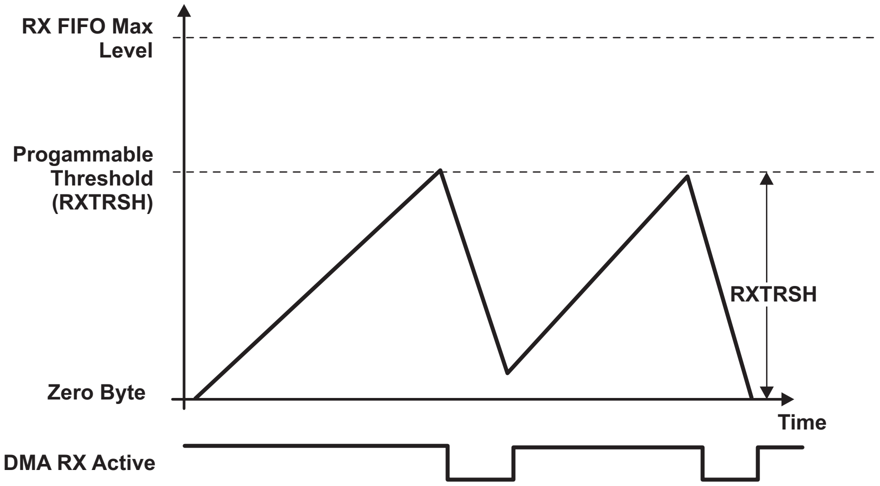
22.3.14.2 FIFO Polling Mode Operation

In FIFO polled mode (I2C_IRQEN_SET.XRDY_IE and I2C_IRQEN_SET.RRDY_IE disabled and DMA disabled), the status of the module (receiver or transmitter) can be checked by polling the XRDY and RRDY status registers (I2C_IRQSTS_RAW) (RDR and XDR can also be polled if draining feature must be used). The XRDY and RRDY flags are accurately reflecting the interrupt conditions mentioned in Interrupt Mode. This mode is an alternative to the FIFO interrupt mode of operation, where the status of the receiver and transmitter is automatically known by means of interrupts sent to the CPU.

22.3.14.3 FIFO DMA Mode Operation

In receive mode, a DMA request is generated as soon as the receive FIFO exceeds its threshold level defined in the threshold level register (I2C_BUF.RXTRSH +1). This request should be de-asserted when the number of bytes defined by the threshold level has been read by the DMA, by setting the I2C_DMARXEN_CLR.DMARX_EN_CLR field.

Figure 22-13. Receive FIFO DMA Request Generation



In transmit mode, a DMA request is automatically asserted when the transmit FIFO is empty. This request should be de-asserted when the number of bytes defined by the number in the threshold register (I2C_BUF.TXTHRS+1) has been written in the FIFO by the DMA, by setting the I2C_DMATXEN_CLR. DMATX_EN_CLR field. If an insufficient number of characters are written, then the DMA request will remain active. Figure 22-14 and Figure 22-15 illustrate DMA TX transfers with different values for TXTRSH.

Figure 22-14. Transmit FIFO DMA Request Generation (High Threshold)

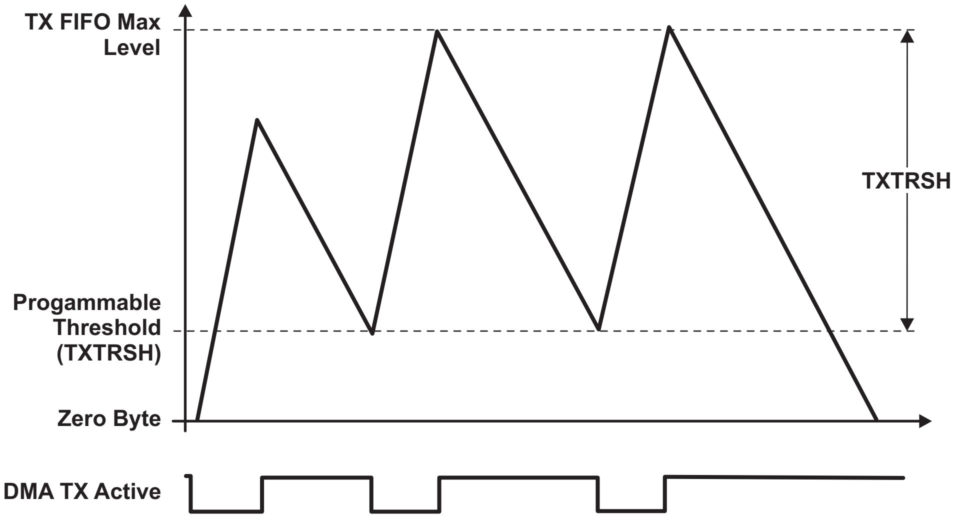
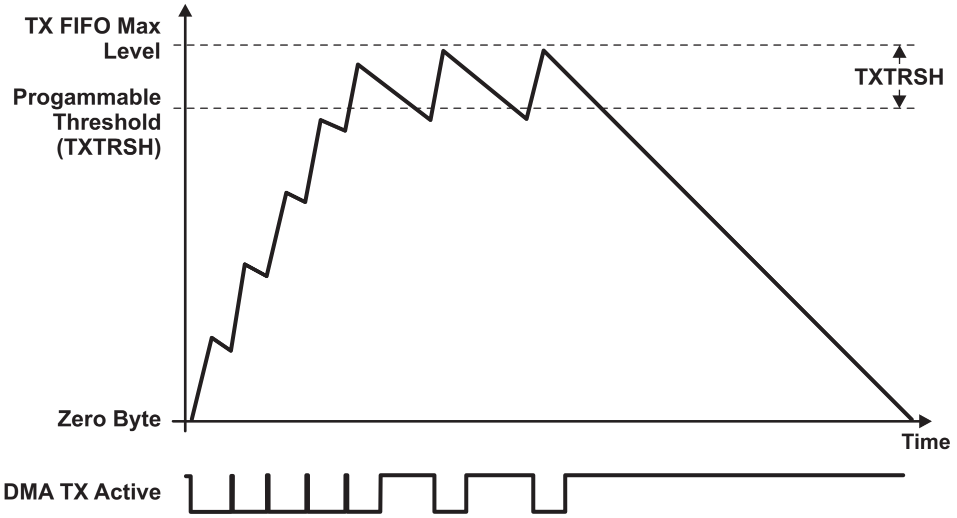


Figure 22-15. Transmit FIFO DMA Request Generation (Low Threshold)



Note that also in DMA mode it is possible to have a transfer whose length is not multiple of the configured FIFO threshold. In this case, the DMA draining feature is also used for transferring the additional bytes of the transfer (see Section 22.3.14.4, *Draining Feature*, for additional details).

According to the desired operation mode, the programmer must set the FIFO thresholds according to the following table (note that only the interface/OCP side thresholds can be programmed; the I2C side thresholds are default equals to 1). Note that the thresholds must be set consistent with the DMA channel length.

In I2C Slave RX Mode, the Local Host can program the RX threshold with the desired value, and use the FIFO draining feature at the end of the I2C transfer to extract from the FIFO the remaining bytes if the threshold is not reached (see Section 22.3.14.4, *Draining Feature*, for additional details).

Note that in I2C Slave TX Mode, the TX FIFO threshold should be set to 1 (I2C_BUF.TXTRSH=0, default value), since the length of the transfer may not be known at configuration time. In this way, the interrupt (or accordingly, DMA) requests will be generated for each byte requested by the remote I2C master to be transferred over the I2C bus. This configuration will prevent the I2C core to request additional data from the CPU or from the DMA controller (using IRQ or DMA), data that will eventually not be extracted from the FIFO by the external master (which can use not acknowledge at any time to end the transfer). If the TX threshold is not set to 1, the module will generate an interrupt or assert a DMA only when the external master requests a byte and the FIFO is empty. However, in this case the TX FIFO will require to be cleared at the end of the transfer.

The I2C module offers the possibility to the user to clear the RX or TX FIFO. This is achieved through I2C_BUF.RXFIFO_CLR and I2C_BUF.TXFIFO_CLR registers, which act like software reset for the FIFOs. In DMA mode, these bits will also reset the DMA state machines.

The FIFO clearing feature can be used when the module is configured as a transmitter, the external receiver responds with a NACK in the middle of the transfer, and there is still data in TX FIFO waiting to be transferred.

On the Functional (I2C) domain, the thresholds can always be considered equal to 1. This means that the I2C Core can start transferring data on the I2C bus whenever it has data in the FIFOs (FIFO is not empty).

22.3.14.4 Draining Feature

The Draining Feature is implemented by the I2C core for handling the end of the transfers whose length is not a multiple of FIFO threshold value, and offers the possibility to transfer the remaining amount of bytes (since the threshold is not reached).

Note that this feature prevents the CPU or the DMA controller to attempt more FIFO accesses than necessary (for example, to generate at the end of a transfer a DMA RX request having in the FIFO less bytes than the configured DMA transfer length). Otherwise, an Access Error interrupt will be generated (see I2C_IRQSTS_RAW.AERR interrupt).

The Draining mechanism will generate an interrupt (I2C_IRQSTS_RAW.RDR or I2C_IRQSTS_RAW.XDR) at the end of the transfer informing the CPU that it needs to check the amount of data left to be transferred (I2C_BUFSTAT.TXSTAT or RXSTAT) and to enable the Draining Feature of the DMA controller if DMA mode is enabled (by re-configuring the DMA transfer length according to this value), or perform only the required number of data accesses, if DMA mode is disabled.

In receiving mode (master or slave), if the RX FIFO threshold is not reached but the transfer was ended on the I2C bus and there is still data left in the FIFO (less than the threshold), the receive draining interrupt (I2C_IRQSTS_RAW.RDR) will be asserted to inform the local host that it can read the amount of data in the FIFO (I2C_BUFSTAT.RXSTAT). The CPU will perform a number of data read accesses equal with RXSTAT value (if interrupt or polling mode) or re-configure the DMA controller with the required value in order to drain the FIFO.

In master transmit mode, if the TX FIFO threshold is not reached but the amount of data remaining to be written in the FIFO is less than TXTRSH, the transmit draining interrupt (I2C_IRQSTS_RAW.XDR) will be asserted to inform the local host that it can read the amount of data remained to be written in the TX FIFO (I2C_BUFSTAT.TXSTAT). The CPU will need to write the required number of data bytes (specified by TXSTAT value) or re-configure the DMA controller with the required value in order to transfer the last bytes to the FIFO.

Note that in master mode, the CPU can alternatively skip the checking of TXSTAT and RXSTAT values since it can obtain internally this information (by computing DATACOUNT modulo TX/RXTHRSH).

The draining feature is disabled by default, and it can be enabled using I2C_IRQEN_SET.XDR_IE or I2C_IRQEN_SET.RDR_IE registers (default disabled) only for the transfers with length not equal with the threshold value.

22.3.15 How to Program I2C

22.3.15.1 Module Configuration Before Enabling the Module

1. Program the prescaler to obtain an approximately 12-MHz I2C module clock (I2C_PSC = x; this value is to be calculated and is dependent on the System clock frequency).
2. Program the I2C clock to obtain 100 Kbps or 400 Kbps (SCLL = x and SCLH = x; these values are to be calculated and are dependent on the System clock frequency).
3. Configure its own address (I2C_OA = x) - only in case of I2C operating mode (F/S mode).
4. Take the I2C module out of reset (I2C_CON:I2C_EN = 1).

22.3.15.2 Initialization Procedure

1. Configure the I2C mode register (I2C_CON) bits.
2. Enable interrupt masks (I2C_IRQEN_SET), if using interrupt for transmit/receive data.
3. Enable the DMA (I2C_BUF and I2C_DMA/RX/TX/EN_SET) and program the DMA controller) - only in case of I2C operating mode (F/S mode), if using DMA for transmit/receive data.

22.3.15.3 Configure Slave Address and DATA Counter Registers

In master mode, configure the slave address (I2C_SA = x) and the number of byte associated with the transfer (I2C_CNT = x).

22.3.15.4 Initiate a Transfer

Poll the bus busy (BB) bit in the I2C status register (I2C_IRQSTS_RAW). If it is cleared to 0 (bus not busy), configure START/STOP (I2C_CON: STT / I2C_CON: STP condition to initiate a transfer) - only in case of I2C operating mode (F/S mode).

22.3.15.5 Receive Data

Poll the receive data ready interrupt flag bit (RRDY) in the I2C status register (I2C_IRQSTS_RAW), use the RRDY interrupt (I2C_IRQEN_SET.RRDY_IE set) or use the DMA RX (I2C_BUF.RDMA_EN set together with I2C_DMARXEN_SET) to read the received data in the data receive register (I2C_DATA). Use draining feature (I2C_IRQSTS_RAW.RDR enabled by I2C_IRQEN_SET.RDR_IE)) if the transfer length is not equal with FIFO threshold.

22.3.15.6 Transmit Data

Poll the transmit data ready interrupt flag bit (XRDY) in the I2C status register (I2C_IRQSTS_RAW), use the XRDY interrupt (I2C_IRQEN_SET.XRDY_IE set) or use the DMA TX (I2C_BUF.XDMA_EN set together with I2C_DMATXEN_SET) to write data into the data transmit register (I2C_DATA). Use draining feature (I2C_IRQSTS_RAW.XDR enabled by I2C_IRQEN_SET.XDR_IE)) if the transfer length is not equal with FIFO threshold.

22.3.16 I2C Behavior During Emulation

To configure the I2C to stop during emulation suspend events (for example, debugger breakpoints), set up the I2C and the Debug Subsystem:

1. Set I2C_SYSTEST.FREE=0. This will allow the Suspend_Control signal from the Debug Subsystem (Chapter 31) to stop and start the I2C. Note that if FREE=1, the Suspend_Control signal is ignored and the I2C is free running regardless of any debug suspend event. This FREE bit gives local control from a module perspective to gate the suspend signal coming from the Debug Subsystem.
2. Set the appropriate xxx_Suspend_Control register = 0x9, as described in Section 31.1.1.1, *Debug Suspend Support for Peripherals*. Choose the register appropriate to the peripheral you want to suspend during a suspend event.

22.4 I2C Registers

22.4.1 I2C Registers

Table 22-8 lists the memory-mapped registers for the I2C. All register offset addresses not listed in Table 22-8 should be considered as reserved locations and the register contents should not be modified.

Table 22-8. I2C Registers

Offset	Acronym	Register Name	Section
0h	I2C_REVNB_LO	Module Revision Register (low bytes)	Section 22.4.1.1
4h	I2C_REVNB_HI	Module Revision Register (high bytes)	Section 22.4.1.2
10h	I2C_SYSC	System Configuration Register	Section 22.4.1.3
24h	I2C_IRQSTS_RAW	I2C Status Raw Register	Section 22.4.1.4
28h	I2C_IRQSTS	I2C Status Register	Section 22.4.1.5
2Ch	I2C_IRQEN_SET	I2C Interrupt Enable Set Register	Section 22.4.1.6
30h	I2C_IRQEN_CLR	I2C Interrupt Enable Clear Register	Section 22.4.1.7
34h	I2C_WE	I2C Wakeup Enable Register	Section 22.4.1.8
38h	I2C_DMARXEN_SET	Receive DMA Enable Set Register	Section 22.4.1.9
3Ch	I2C_DMATXEN_SET	Transmit DMA Enable Set Register	Section 22.4.1.10
40h	I2C_DMARXEN_CLR	Receive DMA Enable Clear Register	Section 22.4.1.11
44h	I2C_DMATXEN_CLR	Transmit DMA Enable Clear Register	Section 22.4.1.12
48h	I2C_DMARXWAKE_EN	Receive DMA Wakeup Register	Section 22.4.1.13

Table 22-8. I2C Registers (continued)

Offset	Acronym	Register Name	Section
4Ch	I2C_DMATXWAKE_EN	Transmit DMA Wakeup Register	Section 22.4.1.14
90h	I2C_SYSS	System Status Register	Section 22.4.1.15
94h	I2C_BUF	Buffer Configuration Register	Section 22.4.1.16
98h	I2C_CNT	Data Counter Register	Section 22.4.1.17
9Ch	I2C_DATA	Data Access Register	Section 22.4.1.18
A4h	I2C_CON	I2C Configuration Register	Section 22.4.1.19
A8h	I2C_OA	I2C Own Address Register	Section 22.4.1.20
ACh	I2C_SA	I2C Slave Address Register	Section 22.4.1.21
B0h	I2C_PSC	I2C Clock Prescaler Register	Section 22.4.1.22
B4h	I2C_SCLL	I2C SCL Low Time Register	Section 22.4.1.23
B8h	I2C_SCLH	I2C SCL High Time Register	Section 22.4.1.24
BCh	I2C_SYSTEST	System Test Register	Section 22.4.1.25
C0h	I2C_BUFSTAT	I2C Buffer Status Register	Section 22.4.1.26
C4h	I2C_OA1	I2C Own Address 1 Register	Section 22.4.1.27
C8h	I2C_OA2	I2C Own Address 2 Register	Section 22.4.1.28
CCh	I2C_OA3	I2C Own Address 3 Register	Section 22.4.1.29
D0h	I2C_ACTOA	Active Own Address Register	Section 22.4.1.30
D4h	I2C_SBLOCK	I2C Clock Blocking Enable Register	Section 22.4.1.31

22.4.1.1 I2C_REVNB_LO Register (Offset = 0h) [reset = 0h]

I2C_REVNB_LO is shown in Figure 22-16 and described in Table 22-9.

Return to Summary Table.

This read-only register contains the hard-coded revision number of the module. A write to this register has no effect. I2C controller with interrupt using interrupt vector register (I2C_IV) is revision 1.x. I2C controller with interrupt using status register bits (I2C_IRQSTATUS_RAW) is revision 2.x.

Figure 22-16. I2C_REVNB_LO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTL					MAJOR			CUSTOM		MINOR					
R-0h					R-0h			R-0h		R-0h					

Table 22-9. I2C_REVNB_LO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-11	RTL	R	0h	RTL version.
10-8	MAJOR	R	0h	Major Revision. This field changes when there is a major feature change. This field does not change due to bug fix, or minor feature change.
7-6	CUSTOM	R	0h	Indicates a special version for a particular device. Consequence of use may avoid use of standard Chip Support Library (CSL) / Drivers. 0 if non-custom.
5-0	MINOR	R	0h	Minor Revision This field changes when features are scaled up or down. This field does not change due to bug fix, or major feature change.

22.4.1.2 I2C_REVNB_HI Register (Offset = 4h) [reset = 0h]

I2C_REVNB_HI is shown in Figure 22-17 and described in Table 22-10.

Return to Summary Table.

A reset has no effect on the value returned.

Figure 22-17. I2C_REVNB_HI Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
SCHEME		RESERVED		FUNC			
R-0h		R-0h		R-0h			
7	6	5	4	3	2	1	0
FUNC							
R-0h							

Table 22-10. I2C_REVNB_HI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-14	SCHEME	R	0h	Used to distinguish between old Scheme and current. Spare bit to encode future schemes.
13-12	RESERVED	R	0h	
11-0	FUNC	R	0h	Function: Indicates a software compatible module family

22.4.1.3 I2C_SYSC Register (Offset = 10h) [reset = 0h]

I2C_SYSC is shown in Figure 22-18 and described in Table 22-11.

Return to Summary Table.

This register allows controlling various parameters of the peripheral interface.

Figure 22-18. I2C_SYSC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED						CLKACTIVITY	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
RESERVED			IDLEMODE		ENAWAKEUP	SRST	AUTOIDLE
R-0h			R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 22-11. I2C_SYSC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9-8	CLKACTIVITY	R/W	0h	Clock Activity selection bits. Those bits (one bit for each clock signal present on the boundary of the module) are set to 1 to disable external clock gating mechanism in Idle Mode. Values after reset are low (for both 2 bits). Note: If the System (functional) Clock is cut-off, the module will assert a WakeUp event when it asynchronously detects a Start Condition on the I2C Bus. Note that in this case the first transfer will not be taken into account by the module (NACK will be detected by the external master). 0h = Both clocks can be cut off 1h = Only Interface/OCP clock must be kept active; system clock can be cut off 2h = Only system clock must be kept active; Interface/OCP clock can be cut off 3h = Both clocks must be kept active
7-5	RESERVED	R	0h	
4-3	IDLEMODE	R/W	0h	Idle Mode selection bits. These two bits are used to select one of the idle mode operation mechanisms. Value after reset is 00 (Force Idle). 1h = NO_IDLE 2h = SMART_IDLE 3h = SMART_IDLE_WAKEUP
2	ENAWAKEUP	R/W	0h	Enable Wakeup control bit. When this bit is set to 1, the module enables its own wakeup mechanism. Value after reset is low. 0h = Wakeup mechanism is disabled 1h = Wakeup mechanism is enabled

Table 22-11. I2C_SYSC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	SRST	R/W	0h	SoftReset bit. When this bit is set to 1, entire module is reset as for the hardware reset. This bit is automatically cleared to 0 by the core and it is only reset by the hardware reset. During reads, it always returns 0. Value after reset is low. 0h = Normal mode 1h = The module is reset
0	AUTOIDLE	R/W	0h	Autoidle bit. When this bit is set to 1, the module activates its own idle mode mechanism. By evaluating its internal state, the module can decide to gate part of his internal clock tree in order to improve the overall power consumption. Value after reset is high. 0h = Auto Idle mechanism is disabled 1h = Auto Idle mechanism is enabled

22.4.1.4 I2C_IRQSTS_RAW Register (Offset = 24h) [reset = 0h]

I2C_IRQSTS_RAW is shown in Figure 22-19 and described in Table 22-12.

Return to Summary Table.

This register provides core status information for interrupt handling, showing all active events (enabled and not enabled). The fields are read-write. Writing a 1 to a bit will set it to 1, that is, trigger the IRQ (mostly for debug). Writing a 0 will have no effect, that is, the register value will not be modified. Only enabled, active events will trigger an actual interrupt request on the IRQ output line.

Figure 22-19. I2C_IRQSTS_RAW Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	XDR	RDR	BB	ROVR	XUDF	AAS	BF
R-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
AERR	STC	GC	XRDY	RRDY	ARDY	NACK	AL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-12. I2C_IRQSTS_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14	XDR	R/W	0h	Transmit draining IRQ status. I2C Master Transmit mode only. This read/clear only bit is set to 1 when the module is configured as a master transmitter, the TX FIFO level is below the configured threshold (TXTRSH) and the amount of data still to be transferred is less than TXTRSH. When this bit is set to 1 by the core, CPU must read the I2C_BUFSTAT.TXSTAT register in order to check the amount of data that need to be written in the TX FIFO. Then, according to the mode set (DMA or interrupt), the CPU can enable the DMA draining feature of the DMA controller with the number of data bytes to be transferred (I2C_BUFSTAT.TXSTAT), or generate write data accesses according to this value (IRQ mode). The interrupt needs to be cleared after the DMA controller was reconfigured (if DMA mode enabled), or before generating data accesses to the FIFO (if IRQ mode enabled). If the corresponding interrupt was enabled, an interrupt is signaled to the local host. The CPU can also poll this bit. For more details about TDR generation, refer to the FIFO Management subsection. The CPU can only clear this bit by writing a 1 into the I2C_IRQSTS register. A write 0 has no effect. Value after reset is low. 0h (R/W) = Transmit draining inactive 1h (R/W) = Transmit draining enabled

Table 22-12. I2C_IRQSTS_RAW Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	RDR	R/W	0h	Receive draining IRQ status. I2C Receive mode only. This read/clear only bit is set to 1 when the module is configured as a receiver, a stop condition was received on the bus and the RX FIFO level is below the configured threshold (RXTRSH). When this bit is set to 1 by the core, CPU must read the I2C_BUFSTAT.RXSTAT register in order to check the amount of data left to be transferred from the FIFO. Then, according to the mode set (DMA or interrupt), the CPU needs to enable the draining feature of the DMA controller with the number of data bytes to be transferred (I2C_BUFSTAT.RXSTAT), or generate read data accesses according to this value (IRQ mode). The interrupt needs to be cleared after the DMA controller was reconfigured (if DMA mode enabled), or before generating data accesses to the FIFO (if IRQ mode enabled). If the corresponding interrupt was enabled, an interrupt is signaled to the local host. The CPU can also poll this bit. For more details about RDR generation, refer to the FIFO Management subsection. The CPU can only clear this bit by writing a 1 into the I2C_IRQSTS register. A write 0 has no effect. Value after reset is low. 0h (R/W) = Receive draining inactive 1h (R/W) = Receive draining enabled
12	BB	R	0h	This read-only bit indicates the state of the serial bus. In slave mode, on reception of a start condition, the device sets BB to 1. BB is cleared to 0 after reception of a stop condition. In master mode, the software controls BB. To start a transmission with a start condition, MST, TRX, and STT must be set to 1 in the I2C_CON register. To end a transmission with a stop condition, STP must be set to 1 in the I2C_CON register. When BB = 1 and STT = 1, a restart condition is generated. Value after reset is low. 0h (R/W) = Bus is free 1h (R/W) = Bus is occupied
11	ROVR	R/W	0h	Receive overrun status. Writing into this bit has no effect. I2C receive mode only. This read-only bit indicates whether the receiver has experienced overrun. Overrun occurs when the shift register is full and the receive FIFO is full. An overrun condition does not result in a data loss the peripheral is just holding the bus (low on SCL) and prevents other bytes from being received. ROVR is set to 1 when the I2C has recognized an overrun. ROVR is clear when reading I2C_DATA register, or when resetting the I2C (I2C_CON:I2C_EN = 0). Value after reset is low. 0h (R/W) = Normal operation 1h (R/W) = Receiver overrun

Table 22-12. I2C_IRQSTS_RAW Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	XUDF	R/W	0h	Transmit underflow status. Writing into this bit has no effect. I2C transmit mode only. This read-only bit indicates whether the transmitter has experienced underflow. In master transmit mode, underflow occurs when the shift register is empty, the transmit FIFO is empty, and there are still some bytes to transmit (DCOUNT 0). In slave transmit mode, underflow occurs when the shift register is empty, the transmit FIFO is empty, and there are still some bytes to transmit (read request from external I2C master). XUDF is set to 1 when the I2C has recognized an underflow. The core holds the line till the underflow cause has disappeared. XUDF is clear when writing I2C_DATA register or resetting the I2C (I2C_CON:I2C_EN = 0). Value after reset is low. 0h (R/W) = Normal operation 1h (R/W) = Transmit underflow
9	AAS	R/W	0h	Address recognized as slave IRQ status. I2C mode only. This read only bit is set to 1 by the device when it has recognized its own slave address (or one of the alternative own addresses), or an address of all zeros (8 bits). When this bit is set to 1 by the core, an interrupt is signaled to the local host if the interrupt was enabled. This bit can be cleared in 2 ways: One way is if the interrupt was enabled, it will be cleared by writing 1 into the I2C_IRQSTS register (writing 0 has no effect). The other way is if the interrupt was not enabled, the AAS bit is reset to 0 by restart or stop. Value after reset is low. 0h = No action 1h (R/W) = Address recognized
8	BF	R/W	0h	I2C mode only. This read only bit is set to 1 by the device when the I2C bus became free (after a transfer is ended on the bus stop condition detected). This interrupt informs the Local Host that it can initiate its own I2C transfer on the bus. When this bit is set to 1 by the core, an interrupt is signaled to the local host if the interrupt was enabled. The CPU can only clear this bit by writing a 1 into the I2C_IRQSTS register. Writing 0 has no effect. Value after reset is low. 0h (R/W) = No action 1h (R/W) = Bus Free
7	AERR	R/W	0h	Access Error IRQ status. I2C mode only. This read/clear only bit is set to 1 by the device if an Interface/OCP write access is performed to I2C_DATA while the TX FIFO is full or if an Interface/OCP read access is performed to the I2C_DATA while the RX FIFO is empty. Note that, when the RX FIFO is empty, a read access will return to the previous read data value. When the TX FIFO is full, a write access is ignored. In both events, the FIFO pointers will not be updated. When this bit is set to 1 by the core, an interrupt is signaled to the local host if the interrupt was enabled. The CPU can only clear this bit by writing a 1 into the I2C_IRQSTS register. Writing 0 has no effect. Value after reset is low. 0h = No action 1h (R/W) = Access Error

Table 22-12. I2C_IRQSTS_RAW Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	STC	R/W	0h	Start Condition IRQ status. I2C mode only. This read/clear only bit is set to 1 by the device if previously the module was in idle mode and a start condition was asynchronously detected on the I2C Bus and signalized with an Wakeup (if the I2C_SYSC.ClockActivity allows the system clock to be cut-off). When the Active Mode will be restored and the interrupt generated, this bit will indicate the reason of the wakeup. Note 1: The corresponding interrupt for this bit should be enabled only if the module was configured to allow the possibility of cutting-off the system clock while in Idle State (I2C_SYSC.ClockActivity = 00 or 01). Note 2: The first transfer (corresponding to the detected start condition) will be lost (not taken into account by the module) and it will be used only for generating the WakeUp enable for restoring the Active Mode of the module. On the I2C line, the external master which generated the transfer will detect this behavior as a not acknowledge to the address phase and will possibly restart the transfer. The CPU can only clear this bit by writing a 1 into the I2C_IRQSTS register. Writing 0 has no effect. Value after reset is low. 0h (R/W) = No action 1h (R/W) = Start Condition detected
5	GC	R/W	0h	General call IRQ status. Set to '1' by core when General call address detected and interrupt signaled to MPUSS. Write '1' to clear. I2C mode only. This read/clear only bit is set to 1 by the device if it detects the address of all zeros (8 bits) (general call). When this bit is set to 1 by the core, an interrupt is signaled to the local host if the interrupt was enabled. The CPU can only clear this bit by writing a 1 into the I2C_IRQSTS register. Writing 0 has no effect. Note: When this bit is set to 1, AAS also reads as 1. Value after reset is low. 0h (R/W) = No general call detected 1h (R/W) = General call address detected

Table 22-12. I2C_IRQSTS_RAW Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	XRDY	R/W	0h	Transmit data ready IRQ status. Set to '1' by core when transmitter and when new data is requested. When set to '1' by core, an interrupt is signaled to MPUSS. Write '1' to clear. Transmit mode only (I2C mode). This read/clear only bit (XRDY) is set to 1 when the I2C peripheral is a master or slave transmitter, the CPU needs to send data through the I2C bus, and the module (transmitter) requires new data to be served. Note that a master transmitter requests new data if the FIFO TX level is below the threshold (TXTRSH) and the required amount of data remained to be transmitted (I2C_BUFSTAT.TXSTAT) is greater than the threshold. A slave transmitter requests new data when the FIFO TX level is below the threshold (if TXTRSH > 1), or anytime there is a read request from external master (for each acknowledge received from the master), if TXTRSH = 1. When this bit is set to 1 by the core, an interrupt is signaled to the local host if the interrupt was enabled. The CPU can also poll this bit (refer to the FIFO Management subsection for details about XRDY generation). The CPU can only clear this bit by writing a 1 into the I2C_IRQSTS register. Writing 0 has no effect. Note: If the DMA transmit mode is enabled (I2C_BUF.XDMA_EN is set, together with I2C_DMATXEN_SET), this bit is forced to 0 and no interrupt will be generated instead, a DMA TX request to the main DMA controller of the system is generated. Value after reset is low. 0h (R/W) = Transmission ongoing 1h (R/W) = Transmit data ready
3	RRDY	R/W	0h	Receive mode only (I2C mode). This read/clear only RRDY is set to 1 when the RX FIFO level is above the configured threshold (RXTRSH). When this bit is set to 1 by the core, CPU is able to read new data from the I2C_DATA register. If the corresponding interrupt was enabled, an interrupt is signaled to the local host. The CPU to read the received data in I2C_DATA register can also poll this bit (refer to the FIFO Management subsection for details about RRDY generation). The CPU can only clear this bit by writing a 1 into the I2C_IRQSTS register. A write 0 has no effect. If the DMA receive mode is enabled (I2C_BUF.RDMA_EN is set, together with I2C_DMARXENABLE_SET), this bit is forced to 0 and no interrupt will be generated instead a DMA RX request to the main DMA controller of the system is generated. Value after reset is low. 0h (R/W) = Receive FIFO threshold not reached 1h (R/W) = Receive data ready for read (RX FIFO threshold reached)

Table 22-12. I2C_IRQSTS_RAW Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	ARDY	R/W	0h	I2C mode only. This read/clear only bit, when set to 1, indicates that the previously programmed data and command (receive or transmit, master or slave) has been performed and status bit has been updated. The CPU uses this flag to let it know that the I2C registers are ready to be accessed again. The CPU can only clear this bit by writing a 1 into the I2C_IRQSTS register. A write 0 has no effect. Mode: I2C Master transmit, Others: STP = 1, ARDY Set Condition: DCOUNT = 0. Mode: I2C Master receive, Others: STP = 1, ARDY Set Condition: DCOUNT = 0 and receiver FIFO empty Mode: I2C Master transmit, Others: STP = 0, ARDY Set Condition: DCOUNT passed 0 Mode: I2C Master receive, Others: STP = 0, ARDY Set Condition: DCOUNT passed 0 and receiver FIFO empty Mode: I2C Master transmit, Others: n/a, ARDY Set Condition: Stop or restart condition received from master Mode: I2C Slave receive, Others: n/a, ARDY Set Condition: Stop or restart condition and receiver FIFO empty Value after reset is low. 0h (R/W) = No action 1h (R/W) = Access ready
1	NACK	R/W	0h	No acknowledgment IRQ status. Bit is set when No Acknowledge has been received, an interrupt is signaled to MPUSS. Write '1' to clear this bit. I2C mode only. The read/clear only No Acknowledge flag bit is set when the hardware detects No Acknowledge has been received. When a NACK event occurs on the bus, this bit is set to 1, the core automatically ends the transfer and clears the MST/STP bits in the I2C_CON register and the I2C becomes a slave. Clearing the FIFOs from remaining data might be required. The CPU can only clear this bit by writing a 1 into the I2C_IRQSTS register. Writing 0 has no effect. Value after reset is low. 0h (R/W) = Normal operation 1h (R/W) = Not Acknowledge detected
0	AL	R/W	0h	Arbitration lost IRQ status. This bit is automatically set by the hardware when it loses the Arbitration in master transmit mode, an interrupt is signaled to MPUSS. During reads, it always returns 0. I2C mode only. The read/clear only Arbitration Lost flag bit is set to 1 when the device (configured in master mode) detects it has lost an arbitration (in Address Phase). This happens when two or more masters initiate a transfer on the I2C bus almost simultaneously or when the I2C attempts to start a transfer while BB (bus busy) is 1. When this is set to 1 due to arbitration lost, the core automatically clears the MST/STP bits in the I2C_CON register and the I2C becomes a slave receiver. The CPU can only clear this bit by writing a 1 to the I2C_IRQSTS register. Writing 0 has no effect. Value after reset is low. 0h (R/W) = Normal operation 1h (R/W) = Arbitration lost detected

22.4.1.5 I2C_IRQSTS Register (Offset = 28h) [reset = 0h]

I2C_IRQSTS is shown in Figure 22-20 and described in Table 22-13.

Return to Summary Table.

This register provides core status information for interrupt handling, showing all active and enabled events and masking the others. The fields are read-write. Writing a 1 to a bit will clear it to 0, that is, clear the IRQ. Writing a 0 will have no effect, that is, the register value will not be modified. Only enabled, active events will trigger an actual interrupt request on the IRQ output line. For all the internal fields of the I2C_IRQSTATUS register, the descriptions given in the I2C_IRQSTATUS_RAW subsection are valid.

Figure 22-20. I2C_IRQSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	XDR	RDR	BB	ROVR	XUDF	AAS	BF
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
AERR	STC	GC	XRDY	RRDY	ARDY	NACK	AL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-13. I2C_IRQSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14	XDR	R/W	0h	Transmit draining IRQ enabled status. 0h = Transmit draining inactive 1h = Transmit draining enabled
13	RDR	R/W	0h	Receive draining IRQ enabled status. 0h = Receive draining inactive 1h = Receive draining enabled
12	BB	R/W	0h	Bus busy enabled status. Writing into this bit has no effect. 0h = Bus is free 1h = Bus is occupied
11	ROVR	R/W	0h	Receive overrun enabled status. Writing into this bit has no effect. 0h = Normal operation 1h = Receiver overrun
10	XUDF	R/W	0h	Transmit underflow enabled status. Writing into this bit has no effect. 0h = Normal operation 1h = Transmit underflow
9	AAS	R/W	0h	Address recognized as slave IRQ enabled status. 0h = No action 1h = Address recognized
8	BF	R/W	0h	Bus Free IRQ enabled status. 0h = No action 1h = Bus free

Table 22-13. I2C_IRQSTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	AERR	R/W	0h	Access Error IRQ enabled status. 0h = No action 1h = Access error
6	STC	R/W	0h	Start Condition IRQ enabled status. 0h = No action 1h = Start condition detected
5	GC	R/W	0h	General call IRQ enabled status. Set to '1' by core when General call address detected and interrupt signaled to MPUSS. Write '1' to clear. 0h = No general call detected 1h = General call address detected
4	XRDY	R/W	0h	Transmit data ready IRQ enabled status. Set to '1' by core when transmitter and when new data is requested. When set to '1' by core, an interrupt is signaled to MPUSS. Write '1' to clear. 0h = Transmission ongoing 1h = Transmit data ready
3	RRDY	R/W	0h	Receive data ready IRQ enabled status. Set to '1' by core when receiver mode, a new data is able to be read. When set to '1' by core, an interrupt is signaled to MPUSS. Write '1' to clear. 0h = No data available 1h = Receive data available
2	ARDY	R/W	0h	Register access ready IRQ enabled status. When set to '1' it indicates that previous access has been performed and registers are ready to be accessed again. An interrupt is signaled to MPUSS. Write '1' to clear. 0h = Module busy 1h = Access ready
1	NACK	R/W	0h	No acknowledgment IRQ enabled status. Bit is set when No Acknowledge has been received, an interrupt is signaled to MPUSS. Write '1' to clear this bit. 0h = Normal operation 1h = Not Acknowledge detected
0	AL	R/W	0h	Arbitration lost IRQ enabled status. This bit is automatically set by the hardware when it loses the Arbitration in master transmit mode, an interrupt is signaled to MPUSS. During reads, it always returns 0. 0h = Normal operation 1h = Arbitration lost detected

22.4.1.6 I2C_IRQEN_SET Register (Offset = 2Ch) [reset = 0h]

I2C_IRQEN_SET is shown in Figure 22-21 and described in Table 22-14.

Return to Summary Table.

All 1-bit fields enable a specific interrupt event to trigger an interrupt request. Writing a 1 to a bit will enable the field. Writing a 0 will have no effect, that is, the register value will not be modified. For all the internal fields of the I2C_IRQENABLE_SET register, the descriptions given in the I2C_IRQSTATUS_RAW subsection are valid.

Figure 22-21. I2C_IRQEN_SET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	XDR_IE	RDR_IE	RESERVED	ROVR	XUDF	AAS_IE	BF_IE
R-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
AERR_IE	STC_IE	GC_IE	XRDY_IE	RRDY_IE	ARDY_IE	NACK_IE	AL_IE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-14. I2C_IRQEN_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14	XDR_IE	R/W	0h	Transmit draining interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[XDR]. 0h = Transmit draining interrupt disabled 1h = Transmit draining interrupt enabled
13	RDR_IE	R/W	0h	Receive draining interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[RDR]. 0h = Receive draining interrupt disabled 1h = Receive draining interrupt enabled
12	RESERVED	R	0h	
11	ROVR	R/W	0h	Receive overrun enable set. 0h = Receive overrun interrupt disabled 1h = Receive draining interrupt enabled
10	XUDF	R/W	0h	Transmit underflow enable set. 0h = Transmit underflow interrupt disabled 1h = Transmit underflow interrupt enabled
9	AAS_IE	R/W	0h	Addressed as slave interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[AAS]. 0h = Addressed as slave interrupt disabled 1h = Addressed as slave interrupt enabled
8	BF_IE	R/W	0h	Bus free interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[BF]. 0h = Bus free interrupt disabled 1h = Bus free interrupt enabled
7	AERR_IE	R/W	0h	Access error interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[AERR]. 0h = Access error interrupt disabled 1h = Access error interrupt enabled

Table 22-14. I2C_IRQEN_SET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	STC_IE	R/W	0h	Start condition interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[STC]. 0h = Start condition interrupt disabled 1h = Start condition interrupt enabled
5	GC_IE	R/W	0h	General call interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[GC]. 0h = General call interrupt disabled 1h = General call interrupt enabled
4	XRDY_IE	R/W	0h	Transmit data ready interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[XRDY]. 0h = Transmit data ready interrupt disabled 1h = Transmit data ready interrupt enabled
3	RRDY_IE	R/W	0h	Receive data ready interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[RRDY]. 0h = Receive data ready interrupt disabled 1h = Receive data ready interrupt enabled
2	ARDY_IE	R/W	0h	Register access ready interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[ARDY]. 0h = Register access ready interrupt disabled 1h = Register access ready interrupt enabled
1	NACK_IE	R/W	0h	No acknowledgment interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[NACK]. 0h = Not Acknowledge interrupt disabled 1h = Not Acknowledge interrupt enabled
0	AL_IE	R/W	0h	Arbitration lost interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[AL]. 0h = Arbitration lost interrupt disabled 1h = Arbitration lost interrupt enabled

22.4.1.7 I2C_IRQEN_CLR Register (Offset = 30h) [reset = 0h]

I2C_IRQEN_CLR is shown in Figure 22-22 and described in Table 22-15.

Return to Summary Table.

All 1-bit fields clear a specific interrupt event. Writing a 1 to a bit will disable the interrupt field. Writing a 0 will have no effect, that is, the register value will not be modified. For all the internal fields of the I2C_IRQENABLE_CLR register, the descriptions given in the I2C_IRQSTATUS_RAW subsection are valid.

Figure 22-22. I2C_IRQEN_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	XDR_IE	RDR_IE	RESERVED	ROVR	XUDF	AAS_IE	BF_IE
R-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
AERR_IE	STC_IE	GC_IE	XRDY_IE	RRDY_IE	ARDY_IE	NACK_IE	AL_IE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-15. I2C_IRQEN_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14	XDR_IE	R/W	0h	Transmit draining interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[XDR]. 0h = Transmit draining interrupt disabled 1h = Transmit draining interrupt enabled
13	RDR_IE	R/W	0h	Receive draining interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[RDR]. 0h = Receive draining interrupt disabled 1h = Receive draining interrupt enabled
12	RESERVED	R	0h	
11	ROVR	R/W	0h	Receive overrun enable clear. 0h = Receive overrun interrupt disabled 1h = Receive draining interrupt enabled
10	XUDF	R/W	0h	Transmit underflow enable clear. 0h = Transmit underflow interrupt disabled 1h = Transmit underflow interrupt enabled
9	AAS_IE	R/W	0h	Addressed as slave interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[AAS]. 0h = Addressed as slave interrupt disabled 1h = Addressed as slave interrupt enabled
8	BF_IE	R/W	0h	Bus Free interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[BF]. 0h = Bus free interrupt disabled 1h = Bus free interrupt enabled
7	AERR_IE	R/W	0h	Access error interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[AERR]. 0h = Access error interrupt disabled 1h = Access error interrupt enabled

Table 22-15. I2C_IRQEN_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	STC_IE	R/W	0h	Start condition interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[STC]. 0h = Start condition interrupt disabled 1h = Start condition interrupt enabled
5	GC_IE	R/W	0h	General call interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[GC]. 0h = General call interrupt disabled 1h = General call interrupt enabled
4	XRDY_IE	R/W	0h	Transmit data ready interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[XRDY]. 0h = Transmit data ready interrupt disabled 1h = Transmit data ready interrupt enabled
3	RRDY_IE	R/W	0h	Receive data ready interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[RRDY] 0h = Receive data ready interrupt disabled 1h = Receive data ready interrupt enabled
2	ARDY_IE	R/W	0h	Register access ready interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[ARDY]. 0h = Register access ready interrupt disabled 1h = Register access ready interrupt enabled
1	NACK_IE	R/W	0h	No acknowledgment interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[NACK]. 0h = Not Acknowledge interrupt disabled 1h = Not Acknowledge interrupt enabled
0	AL_IE	R/W	0h	Arbitration lost interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[AL]. 0h = Arbitration lost interrupt disabled 1h = Arbitration lost interrupt enabled

22.4.1.8 I2C_WE Register (Offset = 34h) [reset = 0h]

I2C_WE is shown in Figure 22-23 and described in Table 22-16.

Return to Summary Table.

Every 1-bit field in the I2C_WE register enables a specific (synchronous) IRQ request source to generate an asynchronous wakeup (on the appropriate swakeup line). When a bit location is set to 1 by the local host, a wakeup is signaled to the local host if the corresponding event is captured by the core of the I2C controller. Value after reset is low (all bits). There is no need for an Access Error WakeUp event, since this event occurs only when the module is in Active Mode (for Interface/OCP accesses to FIFO) and is signaled by an interrupt. With the exception of Start Condition WakeUp, which is asynchronously detected when the Functional clock is turned-off, all the other WakeUp events require the Functional (System) clock to be enabled.

Figure 22-23. I2C_WE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED	XDR_WE	RDR_WE	RESERVED	ROVR_WE	XUDF_WE	AAS_WE	BF_WE
R-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	STC_WE	GC_WE	RESERVED	DRDY_WE	ARDY_WE	NACK_WE	AL_WE
R-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-16. I2C_WE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14	XDR_WE	R/W	0h	Transmit draining wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is in idle mode, the TX FIFO level is below the threshold and the amount of data left to be transferred is less than TXTRSH value. This allows for the module to inform the CPU that it can check the amount of data to be written to the FIFO. 0h = Transmit draining wakeup disabled 1h = Transmit draining wakeup enabled
13	RDR_WE	R/W	0h	Receive draining wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C is in idle mode, configured as a receiver, and it has detected a stop condition on the bus but the RX FIFO threshold is not reached (but the FIFO is not empty). This allows for the module to inform the CPU that it can check the amount of data to be transferred from the FIFO. 0h = Receive draining wakeup disabled 1h = Receive draining wakeup enabled
12	RESERVED	R	0h	
11	ROVR_WE	R/W	0h	Receive overrun wakeup enable 0h = Receive overrun wakeup disabled 1h = Receive overrun wakeup enabled
10	XUDF_WE	R/W	0h	Transmit underflow wakeup enable 0h = Transmit underflow wakeup disabled 1h = Transmit underflow wakeup enabled

Table 22-16. I2C_WE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	AAS_WE	R/W	0h	Address as slave IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is in idle mode, and external master addresses the I2C module as a slave. This allows for the module to inform the CPU that it can check which of the own addresses was used by the external master to access the I2C core. 0h = Addressed as slave wakeup disabled 1h = Addressed as slave wakeup enabled
8	BF_WE	R/W	0h	Bus free IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is in idle mode and the I2C bus became free. This allows for the module to inform the CPU that it can initiate its own transfer on the I2C line. 0h = Bus Free wakeup disabled 1h = Bus Free wakeup enabled
7	RESERVED	R	0h	
6	STC_WE	R/W	0h	Start condition IRQ wakeup set. This read/write bit is used to enable or disable wakeup signal generation when I2C module is in idle mode (with the functional clock inactive) and a possible start condition is detected on the I2C line. The STC WakeUp is generated only if the I2C_SYSC.ClockActivity field indicates that the functional clock can be disabled. Note that if the functional clock is not active, the start condition is asynchronously detected (no filtering and synchronization is used). For this reason, it is possible that the signalized start condition to be a glitch. If the functional clock cannot be disabled (I2C_SYSC.ClockActivity = 10 or 11), the programmer should not enable this wakeup, since the module has other synchronously detected WakeUp event that might be used to exit from idle mode, only if the detected transfer is accessing the I2C module. 0h = Start condition wakeup disabled 1h = Start condition wakeup enabled
5	GC_WE	R/W	0h	General call IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is in idle mode and a general call is received on I2C line. 0h = General call wakeup disabled 1h = General call wakeup enabled
4	RESERVED	R	0h	
3	DRDY_WE	R/W	0h	Receive/Transmit data ready IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is involved into a long transfer and no more registers accesses are performed on the interface (for example module are set in F/S I2C master transmitter mode and FIFO is full). If in the middle of such a transaction, the FIFO buffer needs more data to be transferred, CPU must be informed to write (in case of transmitter mode) or read (if receiver mode) in/from the FIFO. 0h = Transmit/receive data ready wakeup disabled 1h = Transmit/receive data ready wakeup enabled
2	ARDY_WE	R/W	0h	Register access ready IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is involved into a long transfer and no more registers accesses are performed on the interface (for example the module is set in F/S I2C master transmitter mode and FIFO is full). If the current transaction is finished, the module needs to inform CPU about transmission completion. 0h = Register access ready wakeup disabled 1h = Register access ready wakeup enabled

Table 22-16. I2C_WE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	NACK_WE	R/W	0h	No acknowledgment IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is involved into a long transfer and no more registers accesses are performed on the interface (for example the module is set in F/S I2C master transmitter mode and FIFO is full). If in the middle of such of a transaction a Not Acknowledgment event is raised, the module needs to inform CPU about transmission error. 0h = Not Acknowledge wakeup disabled 1h = Not Acknowledge wakeup enabled
0	AL_WE	R/W	0h	Arbitration lost IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is configured as a master and it loses the arbitration. This wake up is very useful when the module is configured as a master transmitter, all the necessary data is provided in the FIFO Tx, STT is enabled and the module enters in Idle Mode. If the module loses the arbitration, an Arbitration Lost event is raised and the module needs to inform CPU about transmission error. Note: The AL wakeup must be enabled only for multimaster communication. If the AL_WE is not enabled and the scenario described above occurs, the module will not be able to inform the CPU about the state of the transfer and it will be blocked in an undetermined state. 0h = Arbitration lost wakeup disabled 1h = Arbitration lost wakeup enabled

22.4.1.9 I2C_DMARXEN_SET Register (Offset = 38h) [reset = 0h]

I2C_DMARXEN_SET is shown in Figure 22-24 and described in Table 22-17.

Return to Summary Table.

The 1-bit field enables a receive DMA request. Writing a 1 to this field will set it to 1. Writing a 0 will have no effect, that is, the register value is not modified. Note that the I2C_BUF.RDMA_EN field is the global (slave) DMA enabler, and that it is disabled by default. The I2C_BUF.RDMA_EN field should also be set to 1 to enable a receive DMA request.

Figure 22-24. I2C_DMARXEN_SET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							DMARX_EN_SET
R-0h							R/W-0h

Table 22-17. I2C_DMARXEN_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	DMARX_EN_SET	R/W	0h	Receive DMA channel enable set.

22.4.1.10 I2C_DMATXEN_SET Register (Offset = 3Ch) [reset = 0h]

I2C_DMATXEN_SET is shown in Figure 22-25 and described in Table 22-18.

Return to Summary Table.

The 1-bit field enables a transmit DMA request. Writing a 1 to this field will set it to 1. Writing a 0 will have no effect, that is, the register value is not modified. Note that the I2C_BUF.XDMA_EN field is the global (slave) DMA enabler, and that it is disabled by default. The I2C_BUF.XDMA_EN field should also be set to 1 to enable a transmit DMA request.

Figure 22-25. I2C_DMATXEN_SET Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="7">RESERVED</td><td>DMATX_TRANSMIT_SET</td></tr>
<tr><td colspan="7">R-0h</td><td>R/W-0h</td></tr>
</table>

Table 22-18. I2C_DMATXEN_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	DMATX_TRANSMIT_SET	R/W	0h	Transmit DMA channel enable set.

22.4.1.11 I2C_DMARXEN_CLR Register (Offset = 40h) [reset = 0h]

I2C_DMARXEN_CLR is shown in Figure 22-26 and described in Table 22-19.

Return to Summary Table.

The 1-bit field disables a receive DMA request. Writing a 1 to a bit will clear it to 0. Another result of setting to 1 the DMARX_ENABLE_CLEAR field, is the reset of the DMA RX request and wakeup lines. Writing a 0 will have no effect, that is, the register value is not modified.

Figure 22-26. I2C_DMARXEN_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							DMARX_EN_CLR
R-0h							R/W-0h

Table 22-19. I2C_DMARXEN_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	DMARX_EN_CLR	R/W	0h	Receive DMA channel enable clear.

22.4.1.12 I2C_DMATXEN_CLR Register (Offset = 44h) [reset = 0h]

I2C_DMATXEN_CLR is shown in Figure 22-27 and described in Table 22-20.

Return to Summary Table.

The 1-bit field disables a transmit DMA request. Writing a 1 to a bit will clear it to 0. Another result of setting to 1 the DMATX_EN_CLEAR field, is the reset of the DMA TX request and wakeup lines. Writing a 0 will have no effect, that is, the register value is not modified.

Figure 22-27. I2C_DMATXEN_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							DMATX_EN_CLR
R-0h							R/W-0h

Table 22-20. I2C_DMATXEN_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	DMATX_EN_CLR	R/W	0h	Transmit DMA channel enable clear.

22.4.1.13 I2C_DMARXWAKE_EN Register (Offset = 48h) [reset = 0h]

I2C_DMARXWAKE_EN is shown in Figure 22-28 and described in Table 22-21.

Return to Summary Table.

All 1-bit fields enable a specific (synchronous) DMA request source to generate an asynchronous wakeup (on the appropriate swakeup line). Note that the I2C_SYSC.ENAWAKEUP field is the global (slave) wakeup enabler, and that it is disabled by default.

Figure 22-28. I2C_DMARXWAKE_EN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	XDR	RDR	RESERVED	ROVR	XUDF	AAS	BF
R-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	STC	GC	RESERVED	DRDY	ARDY	NACK	AL
R-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-21. I2C_DMARXWAKE_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14	XDR	R/W	0h	Transmit draining wakeup set. 0h = Transmit draining interrupt disabled 1h = Transmit draining interrupt enabled
13	RDR	R/W	0h	Receive draining wakeup set. 0h = Receive draining interrupt disabled 1h = Receive draining interrupt enabled
12	RESERVED	R	0h	
11	ROVR	R/W	0h	Receive overrun wakeup set. 0h = Receive overrun interrupt disabled 1h = Receive draining interrupt enabled
10	XUDF	R/W	0h	Transmit underflow wakeup set. 0h = Transmit underflow interrupt disabled 1h = Transmit underflow interrupt enabled
9	AAS	R/W	0h	Address as slave IRQ wakeup set. 0h = Addressed as slave interrupt disabled 1h = Addressed as slave interrupt enabled
8	BF	R/W	0h	Bus free IRQ wakeup set. 0h = Bus free wakeup disabled 1h = Bus free wakeup enabled
7	RESERVED	R	0h	
6	STC	R/W	0h	Start condition IRQ wakeup set. 0h = Start condition wakeup disabled 1h = Start condition wakeup enabled
5	GC	R/W	0h	General call IRQ wakeup set. 0h = General call wakeup disabled 1h = General call wakeup enabled

Table 22-21. I2C_DMARXWAKE_EN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	RESERVED	R	0h	
3	DRDY	R/W	0h	Receive/transmit data ready IRQ wakeup set. 0h = Transmit/receive data ready wakeup disabled 1h = Transmit/receive data ready wakeup enabled
2	ARDY	R/W	0h	Register access ready IRQ wakeup set. 0h = Register access ready wakeup disabled 1h = Register access ready wakeup enabled
1	NACK	R/W	0h	No acknowledgment IRQ wakeup set. 0h = Not Acknowledge wakeup disabled 1h = Not Acknowledge wakeup enabled
0	AL	R/W	0h	Arbitration lost IRQ wakeup set. 0h = Arbitration lost wakeup disabled 1h = Arbitration lost wakeup enabled

22.4.1.14 I2C_DMATXWAKE_EN Register (Offset = 4Ch) [reset = 0h]

I2C_DMATXWAKE_EN is shown in Figure 22-29 and described in Table 22-22.

Return to Summary Table.

All 1-bit fields enable a specific (synchronous) DMA request source to generate an asynchronous wakeup (on the appropriate swakeup line). Note that the I2C_SYSC.ENAWAKEUP field is the global (slave) wakeup enabler, and that it is disabled by default.

Figure 22-29. I2C_DMATXWAKE_EN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	XDR	RDR	RESERVED	ROVR	XUDF	AAS	BF
R-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	STC	GC	RESERVED	DRDY	ARDY	NACK	AL
R-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-22. I2C_DMATXWAKE_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14	XDR	R/W	0h	Transmit draining wakeup set. 0h = Transmit draining interrupt disabled 1h = Transmit draining interrupt enabled
13	RDR	R/W	0h	Receive draining wakeup set. 0h = Receive draining interrupt disabled 1h = Receive draining interrupt enabled
12	RESERVED	R	0h	
11	ROVR	R/W	0h	Receive overrun wakeup set. 0h = Receive overrun interrupt disabled 1h = Receive draining interrupt enabled
10	XUDF	R/W	0h	Transmit underflow wakeup set. 0h = Transmit underflow interrupt disabled 1h = Transmit underflow interrupt enabled
9	AAS	R/W	0h	Address as slave IRQ wakeup set. 0h = Addressed as slave interrupt disabled 1h = Addressed as slave interrupt enabled
8	BF	R/W	0h	Bus free IRQ wakeup set. 0h = Bus free wakeup disabled 1h = Bus free wakeup enabled
7	RESERVED	R	0h	
6	STC	R/W	0h	Start condition IRQ wakeup set. 0h = Start condition wakeup disabled 1h = Start condition wakeup enabled
5	GC	R/W	0h	General call IRQ wakeup set. 0h = General call wakeup disabled 1h = General call wakeup enabled

Table 22-22. I2C_DMATXWAKE_EN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	RESERVED	R	0h	
3	DRDY	R/W	0h	Receive/transmit data ready IRQ wakeup set. 0h = Transmit/receive data ready wakeup disabled 1h = Transmit/receive data ready wakeup enabled
2	ARDY	R/W	0h	Register access ready IRQ wakeup set. 0h = Register access ready wakeup disabled 1h = Register access ready wakeup enabled
1	NACK	R/W	0h	No acknowledgment IRQ wakeup set. 0h = Not Acknowledge wakeup disabled 1h = Not Acknowledge wakeup enabled
0	AL	R/W	0h	Arbitration lost IRQ wakeup set. 0h = Arbitration lost wakeup disabled 1h = Arbitration lost wakeup enabled

22.4.1.15 I2C_SYSS Register (Offset = 90h) [reset = 0h]

I2C_SYSS is shown in Figure 22-30 and described in Table 22-23.

Return to Summary Table.

Figure 22-30. I2C_SYSS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="7">RESERVED</td><td>RDONE</td></tr>
<tr><td colspan="7">R-0h</td><td>R/W-0h</td></tr>
</table>

Table 22-23. I2C_SYSS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	RDONE	R/W	0h	Reset done bit. This read-only bit indicates the state of the reset in case of hardware reset, global software reset (I2C_SYSC.SRST) or partial software reset (I2C_CON.I2C_EN). The module must receive all its clocks before it can grant a reset-completed status. Value after reset is low. 0h = Internal module reset in ongoing 1h = Reset completed

22.4.1.16 I2C_BUF Register (Offset = 94h) [reset = 0h]

I2C_BUF is shown in Figure 22-31 and described in Table 22-24.

Return to Summary Table.

This read/write register enables DMA transfers and allows the configuration of FIFO thresholds for the FIFO management (see the FIFO Management subsection).

Figure 22-31. I2C_BUF Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RDMA_EN	RXFIFO_CLR	RXTRSH					
R/W-0h	R/W-0h	R/W-0h					
7	6	5	4	3	2	1	0
XDMA_EN	TXFIFO_CLR	TXTRSH					
R/W-0h	R/W-0h	R/W-0h					

Table 22-24. I2C_BUF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	RDMA_EN	R/W	0h	Receive DMA channel enable. When this bit is set to 1, the receive DMA channel is enabled and the receive data ready status bit (I2C_IRQSTATUS_RAW: RRDY) is forced to 0 by the core. Value after reset is low. 0h = Receive DMA channel disabled 1h = Receive DMA channel enabled
14	RXFIFO_CLR	R/W	0h	Receive FIFO clear. When set, receive FIFO is cleared (hardware reset for RX FIFO generated). This bit is automatically reset by the hardware. During reads, it always returns 0. Value after reset is low. 0h = Normal mode 1h = Rx FIFO is reset
13-8	RXTRSH	R/W	0h	Threshold value for FIFO buffer in RX mode. The receive threshold value is used to specify the trigger level for data receive transfers. The value is specified from the Interface/OCP point of view. Value after reset is 00h. For the FIFO management description, see the FIFO Management subsection. Note 1: programmed threshold cannot exceed the actual depth of the FIFO. Note 2: the threshold must not be changed while a transfer is in progress (after STT was configured or after the module was addressed as a slave). 0h = Receive Threshold value = 64 0h = Receive Threshold value = 1 1h = Receive Threshold value = 2

Table 22-24. I2C_BUF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	XDMA_EN	R/W	0h	Transmit DMA channel enable. When this bit is set to 1, the transmit DMA channel is enabled and the transmit data ready status (I2C_IRQSTATUS_RAW: XRDY) bit is forced to 0 by the core. Value after reset is low. 0h = Transmit DMA channel disabled 1h = Transmit DMA channel enabled
6	TXFIFO_CLR	R/W	0h	Transmit FIFO clear. When set, transmit FIFO is cleared (hardware reset for TX FIFO). This bit is automatically reset by the hardware. During reads, it always returns 0. Value after reset is low. 0h = Normal mode 1h = Tx FIFO is reset
5-0	TXTRSH	R/W	0h	Threshold value for FIFO buffer in TX mode. The Transmit Threshold value is used to specify the trigger level for data transfers. The value is specified from the OCP point of view. Value after reset is 00h Note 1: programmed threshold cannot exceed the actual depth of the FIFO. Note 2: the threshold must not be changed while a transfer is in progress (after STT was configured or after the module was addressed as a slave). 0h = Transmit Threshold value = 64 0h = Transmit Threshold value = 1 1h = Transmit Threshold value = 2

22.4.1.17 I2C_CNT Register (Offset = 98h) [reset = 0h]

I2C_CNT is shown in Figure 22-32 and described in Table 22-25.

Return to Summary Table.

CAUTION: During an active transfer phase (between STT having been set to 1 and reception of ARDY), no modification must be done in this register. Changing it may result in an unpredictable behavior. This read/write register is used to control the numbers of bytes in the I2C data payload.

Figure 22-32. I2C_CNT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">DCOUNT</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">R/W-0h</td></tr>
</table>

Table 22-25. I2C_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	DCOUNT	R/W	0h	Data count. I2C Master Mode only (receive or transmit F/S). This 16-bit countdown counter decrements by 1 for every byte received or sent through the I2C interface. A write initializes DCOUNT to a saved initial value. A read returns the number of bytes that are yet to be received or sent. A read into DCOUNT returns the initial value only before a start condition and after a stop condition. When DCOUNT reaches 0, the core generates a stop condition if a stop condition was specified (I2C_CON.STP = 1) and the ARDY status flag is set to 1 in the I2C_IRQSTATUS_RAW register. Note that DCOUNT must not be reconfigured after I2C_CON.STT was enabled and before ARDY is received. Note 1: In case of I2C mode of operation, if I2C_CON.STP = 0, then the I2C asserts SCL = 0 when DCOUNT reaches 0. The CPU can then reprogram DCOUNT to a new value and resume sending or receiving data with a new start condition (restart). This process repeats until the CPU sets to 1 the I2C_CON.STP bit. The ARDY flag is set each time DCOUNT reaches 0 and DCOUNT is reloaded to its initial value. Values after reset are low (all 16 bits). Note 2: Since for DCOUNT = 0, the transfer length is 65536, the module does not allow the possibility to initiate zero data bytes transfers. 0h = Data counter = 65535 bytes (216 - 1) 0h = Data counter = 65536 bytes (216) 1h = Data counter = 1 bytes

22.4.1.18 I2C_DATA Register (Offset = 9Ch) [reset = 0h]

I2C_DATA is shown in Figure 22-33 and described in Table 22-26.

Return to Summary Table.

This register is the entry point for the local host to read data from or write data to the FIFO buffer.

Figure 22-33. I2C_DATA Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">RESERVED</td><td colspan="8">DATA</td></tr>
<tr><td colspan="24">R-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 22-26. I2C_DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	DATA	R/W	0h	Transmit/Receive data FIFO endpoint. When read, this register contains the received I2C data. When written, this register contains the byte value to transmit over the I2C data. In SYSTEST loop back mode (I2C_SYSTEST: TMODE = 11), this register is also the entry/receive point for the data. Values after reset are unknown (all 8-bits). Note: A read access, when the buffer is empty, returns the previous read data value. A write access, when the buffer is full, is ignored. In both events, the FIFO pointers are not updated and an Access Error (AERR) Interrupt is generated.

22.4.1.19 I2C_CON Register (Offset = A4h) [reset = 0h]

I2C_CON is shown in Figure 22-34 and described in Table 22-27.

Return to Summary Table.

During an active transfer phase (between STT having been set to 1 and reception of ARDY), no modification must be done in this register (except STP enable). Changing it may result in an unpredictable behavior.

Figure 22-34. I2C_CON Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
I2C_EN	RESERVED	OPMODE		STB	MST	TRX	XSA
R/W-0h	R-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
XOA0	XOA1	XOA2	XOA3	RESERVED		STP	STT
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h

Table 22-27. I2C_CON Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	I2C_EN	R/W	0h	I2C module enable. When this bit is cleared to 0, the I2C controller is not enabled and reset. When 0, receive and transmit FIFOs are cleared and all status bits are set to their default values. All configuration registers (I2C_IRQENABLE_SET, I2C_IRQWAKE_SET, I2C_BUF, I2C_CNT, I2C_CON, I2C_OA, I2C_SA, I2C_PSC, I2C_SCLL and I2C_SCLH) are not reset, they keep their initial values and can be accessed. The CPU must set this bit to 1 for normal operation. Value after reset is low. 0h = Controller in reset. FIFO are cleared and status bits are set to their default value. 1h = Module enabled
14	RESERVED	R	0h	
13-12	OPMODE	R/W	0h	Operation mode selection. These two bits select module operation mode. Value after reset is 00. 0h = I2C Fast/Standard mode 1h = Reserved 2h = Reserved 3h = Reserved
11	STB	R/W	0h	Start byte mode (I2C master mode only). The start byte mode bit is set to 1 by the CPU to configure the I2C in start byte mode (I2C_SA = 0000 0001). See the Philips I2C spec for more details. Value after reset is low. 0h = Normal mode 1h = Start byte mode

Table 22-27. I2C_CON Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	MST	R/W	0h	Master/slave mode (I2C mode only). When this bit is cleared, the I2C controller is in the slave mode and the serial clock (SCL) is received from the master device. When this bit is set, the I2C controller is in the master mode and generates the serial clock. Note: This bit is automatically cleared at the end of the transfer on a detected stop condition, in case of arbitration lost or when the module is configured as a master but addressed as a slave by an external master. Value after reset is low. 0h = Slave mode 1h = Master mode
9	TRX	R/W	0h	Transmitter/receiver mode (i2C master mode only). When this bit is cleared, the I2C controller is in the receiver mode and data on data line SDA is shifted into the receiver FIFO and can be read from I2C_DATA register. When this bit is set, the I2C controller is in the transmitter mode and the data written in the transmitter FIFO via I2C_DATA is shifted out on data line SDA. Value after reset is low. The operating modes are defined as follows: MST = 0, TRX = x, Operating Mode = Slave receiver. MST = 0, TRX = x, Operating Mode = Slave transmitter. MST = 1, TRX = 0, Operating Modes = Master receiver. MST = 1, TRX = 0, Operating Modes = Master transmitter. 0h = Receiver mode 1h = Transmitter mode
8	XSA	R/W	0h	Expand slave address. (I2C mode only). When set, this bit expands the slave address to 10-bit. Value after reset is low. 0h = 7-bit address mode 1h = 10-bit address mode
7	XOA0	R/W	0h	Expand own address 0. (I2C mode only). When set, this bit expands the base own address (OA0) to 10-bit. Value after reset is low. 0h = 7-bit address mode 1h = 10-bit address mode
6	XOA1	R/W	0h	Expand own address 1. (I2C mode only). When set, this bit expands the first alternative own address (OA1) to 10-bit. Value after reset is low. 0h = 7-bit address mode 1h = 10-bit address mode
5	XOA2	R/W	0h	Expand own address 2. (I2C mode only). When set, this bit expands the second alternative own address (OA2) to 10-bit. Value after reset is low. 0h = 7-bit address mode. (I2C mode only). 1h = 10-bit address mode
4	XOA3	R/W	0h	Expand own address 3. When set, this bit expands the third alternative own address (OA3) to 10-bit. Value after reset is low. 0h = 7-bit address mode 1h = 10-bit address mode

Table 22-27. I2C_CON Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	RESERVED	R	0h	
1	STP	R/W	0h	Stop condition (I2C master mode only). This bit can be set to a 1 by the CPU to generate a stop condition. It is reset to 0 by the hardware after the stop condition has been generated. The stop condition is generated when DCOUNT passes 0. When this bit is not set to 1 before the end of the transfer (DCOUNT = 0), the stop condition is not generated and the SCL line is hold to 0 by the master, which can re-start a new transfer by setting the STT bit to 1. Value after reset is low 0h = No action or stop condition detected 1h = Stop condition queried
0	STT	R/W	0h	Start condition (I2C master mode only). This bit can be set to a 1 by the CPU to generate a start condition. It is reset to 0 by the hardware after the start condition has been generated. The start/stop bits can be configured to generate different transfer formats. Value after reset is low. Note: DCOUNT is data count value in I2C_CNT register. STT = 1, STP = 0, Conditions = Start, Bus Activities = S-A-D. STT = 0, STP = 1, Conditions = Stop, Bus Activities = P. STT = 1, STP = 1, Conditions = Start-Stop (DCOUNT=n), Bus Activities = S-A-D..(n)..D-P. STT = 1, STP = 0, Conditions = Start (DCOUNT=n), Bus Activities = S-A-D..(n)..D. 0h = No action or start condition detected 1h = Start condition queried

22.4.1.20 I2C_OA Register (Offset = A8h) [reset = 0h]

I2C_OA is shown in Figure 22-35 and described in Table 22-28.

Return to Summary Table.

CAUTION: During an active transfer phase (between STT having been set to 1 and reception of ARDY), no modification must be done in this register. Changing it may result in an unpredictable behavior. This register is used to specify the module s base I2C 7-bit or 10-bit address (base own address).

Figure 22-35. I2C_OA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						OA									
R-0h																						R/W-0h									

Table 22-28. I2C_OA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9-0	OA	R/W	0h	Own address. This field specifies either: A 10-bit address coded on OA [9:0] when XOA (Expand Own Address, I2C_CON[7]) is set to 1. or A 7-bit address coded on OA [6:0] when XOA (Expand Own Address, I2C_CON[7]) is cleared to 0. In this case, OA [9:7] bits must be cleared to 000 by application software. Value after reset is low (all 10 bits).

22.4.1.21 I2C_SA Register (Offset = ACh) [reset = 0h]

I2C_SA is shown in Figure 22-36 and described in Table 22-29.

Return to Summary Table.

CAUTION: During an active transfer phase (between STT having been set to 1 and reception of ARDY), no modification must be done in this register. Changing it may result in an unpredictable behavior. This register is used to specify the addressed I2C module 7-bit or 10-bit address (slave address).

Figure 22-36. I2C_SA Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="22">RESERVED</td><td colspan="10">SA</td></tr>
<tr><td colspan="22">R-0h</td><td colspan="10">R/W-0h</td></tr>
</table>

Table 22-29. I2C_SA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9-0	SA	R/W	0h	Slave address. This field specifies either: A 10-bit address coded on SA [9:0] when XSA (Expand Slave Address, I2C_CON[8]) is set to 1. or A 7-bit address coded on SA [6:0] when XSA (Expand Slave Address, I2C_CON[8]) is cleared to 0. In this case, SA [9:7] bits must be cleared to 000 by application software. Value after reset is low (all 10 bits).

22.4.1.22 I2C_PSC Register (Offset = B0h) [reset = 0h]

I2C_PSC is shown in Figure 22-37 and described in Table 22-30.

Return to Summary Table.

CAUTION: During an active mode (I2C_EN bit in I2C_CON register is set to 1), no modification must be done in this register. Changing it may result in an unpredictable behavior. This register is used to specify the internal clocking of the I2C peripheral core.

Figure 22-37. I2C_PSC Register

31-8	7-0
RESERVED	PSC
R-0h	R/W-0h

Table 22-30. I2C_PSC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	PSC	R/W	0h	Fast/Standard mode prescale sampling clock divider value. The core uses this 8-bit value to divide the system clock (SCLK) and generates its own internal sampling clock (ICLK) for Fast and Standard operation modes. The core logic is sampled at the clock rate of the system clock for the module divided by (PSC + 1). Value after reset is low (all 8 bits). 0h (R/W) = Divide by 1 1h (R/W) = Divide by 2 FFh (R/W) = Divide by 256

22.4.1.23 I2C_SCLL Register (Offset = B4h) [reset = 0h]

I2C_SCLL is shown in Figure 22-38 and described in Table 22-31.

Return to Summary Table.

CAUTION: During an active mode (I2C_EN bit in I2C_CON register is set to 1), no modification must be done in this register. Changing it may result in an unpredictable behavior. This register is used to determine the SCL low time value when master.

Figure 22-38. I2C_SCLL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SCLL							
R-0h																								R/W-0h							

Table 22-31. I2C_SCLL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	SCLL	R/W	0h	Fast/Standard mode SCL low time. I2C master mode only, (FS). This 8-bit value is used to generate the SCL low time value (tLOW) when the peripheral is operated in master mode. tLOW = (SCLL + 7) * ICLK time period, Value after reset is low (all 8 bits).

22.4.1.24 I2C_SCLH Register (Offset = B8h) [reset = 0h]

I2C_SCLH is shown in Figure 22-39 and described in Table 22-32.

Return to Summary Table.

CAUTION: During an active mode (I2C_EN bit in I2C_CON register is set to 1), no modification must be done in this register. Changing it may result in an unpredictable behavior. This register is used to determine the SCL high time value when master.

Figure 22-39. I2C_SCLH Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">RESERVED</td><td colspan="8">SCLH</td></tr>
<tr><td colspan="24">R-0h</td><td colspan="8">R/W-0h</td></tr>
</table>

Table 22-32. I2C_SCLH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	SCLH	R/W	0h	Fast/Standard mode SCL low time. I2C master mode only, (FS). This 8-bit value is used to generate the SCL high time value (tHIGH) when the peripheral is operated in master mode. - tHIGH = (SCLH + 5) * ICLK time period. Value after reset is low (all 8 bits).

22.4.1.25 I2C_SYSTEST Register (Offset = BCh) [reset = 0h]

I2C_SYSTEST is shown in Figure 22-40 and described in Table 22-33.

Return to Summary Table.

CAUTION: Never enable this register for normal I2C operation This register is used to facilitate system-level tests by overriding some of the standard functional features of the peripheral. It allows testing of SCL counters, controlling the signals that connect to I/O pins, or creating digital loop-back for self-test when the module is configured in system test (SYSTEST) mode. It also provides stop/non-stop functionality in the debug mode.

Figure 22-40. I2C_SYSTEST Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
ST_EN	FREE	TMODE		SSB	RESERVED		SCL_I_FUNC
R/W-0h	R/W-0h	R/W-0h		R/W-0h	R-0h		R-0h
7	6	5	4	3	2	1	0
SCL_O_FUNC	SDA_I_FUNC	SDA_O_FUNC	RESERVED	SCL_I	SCL_O	SDA_I	SDA_O
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 22-33. I2C_SYSTEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	ST_EN	R/W	0h	System test enable. This bit must be set to 1 to permit other system test registers bits to be set. Value after reset is low. 0h (R/W) = Normal mode. All others bits in register are read only. 1h (R/W) = System test enabled. Permit other system test registers bits to be set.

Table 22-33. I2C_SYSTEST Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	FREE	R/W	0h	Free running mode (on breakpoint). This bit is used to determine the state of the I2C controller when a breakpoint is encountered in the HLL debugger. Note: This bit can be set independently of ST_EN value. FREE = 0: the I2C controller stops immediately after completion of the on-going bit transfer. Stopping the transfer is achieved by forcing the SCL line low. Note that in this case there will be no status register updates. FREE = 1: the I2C interface runs free. When Suspend indication will be asserted, there will be no accesses on the OCP Interface (the CPU is in debug mode) and consequently the FIFOs will reach full/empty state (according to RX or TX modes) and the I2C SDA line will be kept low. Note that the status registers will be updated, but no DMA, IRQ or WakeUp will be generated. The status registers likely to be updated in this mode are: I2C_IRQSTATUS_RAW.XRDY, I2C_IRQSTATUS_RAW.RRDY, I2C_IRQSTATUS_RAW.XUDF, I2C_IRQSTATUS_RAW.ROVR, I2C_IRQSTATUS_RAW.ARDY and I2C_IRQSTATUS_RAW.NACK. Value after reset is low. 0h (R/W) = Stop mode (on breakpoint condition). If Master mode, it stops after completion of the on-going bit transfer. In slave mode, it stops during the phase transfer when 1 byte is completely transmitted/received. 1h (R/W) = Free running mode
13-12	TMODE	R/W	0h	Test mode select. In normal functional mode (ST_EN = 0), these bits are don't care. They are always read as 00 and a write is ignored. In system test mode (ST_EN = 1), these bits can be set according to the following table to permit various system tests. Values after reset are low (2 bits). SCL counter test mode: in this mode, the SCL pin is driven with a permanent clock as if mastered with the parameters set in the I2C_PSC, I2C_SCLL, and I2C_SCLH registers. Loop back mode: in the master transmit mode only, data transmitted out of the I2C_DATA register (write action) is received in the same I2C_DATA register via an internal path through the FIFO buffer. The DMA and interrupt requests are normally generated if enabled. SDA/SCL IO mode: in this mode, the SCL IO and SDA IO are controlled via the I2C_SYSTEST [5:0] register bits. 0h (R/W) = Functional mode (default) 1h = Reserved 2h (R/W) = Test of SCL counters (SCLL, SCLH, PSC). SCL provides a permanent clock with master mode. 3h (R/W) = Loop back mode select + SDA/SCL IO mode select
11	SSB	R/W	0h	Set status bits. Writing 1 into this bit also sets the 6 read/clear-only status bits contained in I2C_IRQSTATUS_RAW register (bits 5:0) to 1. Writing 0 into this bit doesn't clear status bits that are already set only writing 1 into a set status bit can clear it (see I2C_IRQSTATUS_RAW operation). This bit must be cleared prior attempting to clear a status bit. Value after reset is low. 0h (R/W) = No action. 1h (R/W) = Set all interrupt status bits to 1.
10-9	RESERVED	R	0h	

Table 22-33. I2C_SYSTEST Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	SCL_I_FUNC	R	0h	SCL line input value (functional mode). This read-only bit returns the logical state taken by the SCL line (either 1 or 0). It is active both in functional and test mode. Value after reset is low. 0h (R) = Read 0 from SCL line 1h (R) = Read 1 from SCL line
7	SCL_O_FUNC	R	0h	SCL line output value (functional mode). This read-only bit returns the value driven by the module on the SCL line (either 1 or 0). It is active both in functional and test mode. Value after reset is low. 0h (R) = Driven 0 on SCL line 1h (R) = Driven 1 on SCL line
6	SDA_I_FUNC	R	0h	SDA line input value (functional mode). This read-only bit returns the logical state taken by the SDA line (either 1 or 0). It is active both in functional and test mode. Value after reset is low. 0h (R) = Read 0 from SDA line 1h (R) = Read 1 from SDA line
5	SDA_O_FUNC	R	0h	SDA line output value (functional mode). This read-only bit returns the value driven by the module on the SDA line (either 1 or 0). It is active both in functional and test mode. Value after reset is low. 0h (R) = Driven 0 to SDA line 1h (R) = Driven 1 to SDA line
4	RESERVED	R	0h	
3	SCL_I	R	0h	SCL line sense input value. In normal functional mode (ST_EN = 0), this read-only bit always reads 0. In system test mode (ST_EN = 1 and TMODE = 11), this read-only bit returns the logical state taken by the SCL line (either 1 or 0). Value after reset is low. 0h (R) = Read 0 from SCL line 1h (R) = Read 1 from SCL line
2	SCL_O	R	0h	SCL line drive output value. In normal functional mode (ST_EN = 0), this bit is don't care. It always reads 0 and a write is ignored. In system test mode (ST_EN = 1 and TMODE = 11), a 0 forces a low level on the SCL line and a 1 puts the I2C output driver to a high-impedance state. Value after reset is low. 0h = Forces 0 on the SCL data line 1h (R) = SCL output driver in high-impedance state
1	SDA_I	R	0h	SDA line sense input value. In normal functional mode (ST_EN = 0), this read-only bit always reads 0. In system test mode (ST_EN = 1 and TMODE = 11), this read-only bit returns the logical state taken by the SDA line (either 1 or 0). Value after reset is low. 0h (R) = Read 0 from SDA line 1h (R) = Read 1 from SDA line

Table 22-33. I2C_SYSTEST Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SDA_O	R	0h	SDA line drive output value. In normal functional mode (ST_EN = 0), this bit is don't care. It reads as 0 and a write is ignored. In system test mode (ST_EN = 1 and TMODE = 11), a 0 forces a low level on the SDA line and a 1 puts the I2C output driver to a high-impedance state. Value after reset is low. 0h (W) = Write 0 to SDA line 1h (W) = Write 1 to SDA line

22.4.1.26 I2C_BUFSTAT Register (Offset = C0h) [reset = 0h]

I2C_BUFSTAT is shown in Figure 22-41 and described in Table 22-34.

Return to Summary Table.

This read-only register reflects the status of the internal buffers for the FIFO management (see the FIFO Management subsection).

Figure 22-41. I2C_BUFSTAT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="2">FIFODEPTH</td><td colspan="6">RXSTAT</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="6">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">RESERVED</td><td colspan="6">TXSTAT</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="6">R-0h</td></tr>
</table>

Table 22-34. I2C_BUFSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-14	FIFODEPTH	R	0h	Internal FIFO buffers depth. This read-only bit indicates the internal FIFO buffer depth. Value after reset is given by the boundary module generic parameter. 0h = 8-bytes FIFO 1h = 16-bytes FIFO 2h = 32-bytes FIFO 3h = 64-bytes FIFO
13-8	RXSTAT	R	0h	RX buffer status. This read-only field indicates the number of bytes to be transferred from the FIFO at the end of the I2C transfer (when RDR is asserted). It corresponds to the level indication of the RX FIFO (number of written locations). Value after reset is 0.
7-6	RESERVED	R	0h	
5-0	TXSTAT	R	0h	TX buffer status. This read-only field indicates the number of data bytes still left to be written in the TX FIFO (it s equal with the initial value of I2C_CNT.DCOUNT minus the number of data bytes already written in the TX FIFO through the OCP Interface). Value after reset is equal with 0.

22.4.1.27 I2C_OA1 Register (Offset = C4h) [reset = 0h]

I2C_OA1 is shown in Figure 22-42 and described in Table 22-35.

Return to Summary Table.

CAUTION: During an active transfer phase (between STT has been set to 1 and receiving of ARDY), no modification must be done in this register. Changing it may result in an unpredictable behavior. This register is used to specify the first alternative I2C 7-bit or 10-bit address (own address 1 - OA1).

Figure 22-42. I2C_OA1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						OA1									
R-0h																						R/W-0h									

Table 22-35. I2C_OA1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9-0	OA1	R/W	0h	Own address 1. This field specifies either: A 10-bit address coded on OA1 [9:0] when XOA1 (Expand Own Address 1 - XOA1, I2C_CON[6]) is set to 1. A 7-bit address coded on OA1 [6:0] when XOA1 (Expand Own Address 1 XOA1, I2C_CON[6]) is cleared to 0. In this case, OA1 [9:7] bits must be cleared to 000 by application software. Value after reset is low (all 10 bits).

22.4.1.28 I2C_OA2 Register (Offset = C8h) [reset = 0h]

I2C_OA2 is shown in Figure 22-43 and described in Table 22-36.

Return to Summary Table.

CAUTION: During an active transfer phase (between STT has been set to 1 and receiving of ARDY), no modification must be done in this register. Changing it may result in an unpredictable behavior. This register is used to specify the first alternative I2C 7-bit or 10-bit address (own address 2 - OA2).

Figure 22-43. I2C_OA2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						OA2									
R-0h																						R/W-0h									

Table 22-36. I2C_OA2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9-0	OA2	R/W	0h	Own address 2. This field specifies either: A 10-bit address coded on OA2 [9:0] when XOA1 (Expand Own Address 2 - XOA2, I2C_CON[5]) is set to 1. A 7-bit address coded on OA2 [6:0] when XOA2 (Expand Own Address 2 XOA2, I2C_CON[5]) is cleared to 0. In this case, OA2 [9:7] bits must be cleared to 000 by application software. Value after reset is low (all 10 bits).

22.4.1.29 I2C_OA3 Register (Offset = CCh) [reset = 0h]

I2C_OA3 is shown in Figure 22-44 and described in Table 22-37.

Return to Summary Table.

CAUTION: During an active transfer phase (between STT has been set to 1 and receiving of ARDY), no modification must be done in this register. Changing it may result in an unpredictable behavior. This register is used to specify the first alternative I2C 7-bit or 10-bit address (own address 3 - OA3).

Figure 22-44. I2C_OA3 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="22">RESERVED</td><td colspan="10">OA3</td></tr>
<tr><td colspan="22">R-0h</td><td colspan="10">R/W-0h</td></tr>
</table>

Table 22-37. I2C_OA3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9-0	OA3	R/W	0h	Own address 2. This field specifies either: A 10-bit address coded on OA3 [9:0] when XOA3 (Expand Own Address 3 - XOA3, I2C_CON[4]) is set to 1. A 7-bit address coded on OA3 [6:0] when XOA1 (Expand Own Address 3 XOA3, I2C_CON[4]) is cleared to 0. In this case, OA3 [9:7] bits must be cleared to 000 by application software. Value after reset is low (all 10 bits).

22.4.1.30 I2C_ACTOA Register (Offset = D0h) [reset = 0h]

I2C_ACTOA is shown in Figure 22-45 and described in Table 22-38.

Return to Summary Table.

This read-only register is used to indicate which one of the module s four own addresses the external master used when addressing the module. The CPU can read this register when the AAS indication was activated. The indication is cleared at the end of the transfer.

Figure 22-45. I2C_ACTOA Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				OA3_ACT	OA2_ACT	OA1_ACT	OA0_ACT
R-0h				R-0h	R-0h	R-0h	R-0h

Table 22-38. I2C_ACTOA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	OA3_ACT	R	0h	Own address 3 active. When a bit location is set to 1 by the core, it signalizes to the Local Host that an external master using the corresponding own address addressed the module. Value after reset is low. 0h = Own address inactive 1h = Own address active
2	OA2_ACT	R	0h	Own address 2 active. When a bit location is set to 1 by the core, it signalizes to the Local Host that an external master using the corresponding own address addressed the module. Value after reset is low. 0h = Own address inactive 1h = Own address active
1	OA1_ACT	R	0h	Own address 1 active. When a bit location is set to 1 by the core, it signalizes to the Local Host that an external master using the corresponding own address addressed the module. Value after reset is low. 0h = Own address inactive 1h = Own address active
0	OA0_ACT	R	0h	Own address 0 active. When a bit location is set to 1 by the core, it signalizes to the Local Host that an external master using the corresponding own address addressed the module. Value after reset is low. 0h = Own address inactive 1h = Own address active

22.4.1.31 I2C_SBLOCK Register (Offset = D4h) [reset = 0h]

I2C_SBLOCK is shown in Figure 22-46 and described in Table 22-39.

Return to Summary Table.

This read/write register controls the automatic blocking of I2C clock feature in slave mode. It is used for the Local Host to configure for which of the 4 own addresses, the core must block the I2C clock (keep SCL line low) right after the Address Phase, when it is addressed as a slave.

Figure 22-46. I2C_SBLOCK Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td>OA3_EN</td><td>OA2_EN</td><td>OA1_EN</td><td>OA0_EN</td></tr>
<tr><td colspan="4">R-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 22-39. I2C_SBLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	OA3_EN	R/W	0h	Enable I2C clock blocking for own address 3. When the CPU sets a bit location to 1, if an external master using the corresponding own address addresses the core, the core will block the I2C clock right after the address phase. For releasing the I2C clock the CPU must write 0 in the corresponding field. Value after reset is low. 0h = I2C clock released 1h = I2C clock blocked
2	OA2_EN	R/W	0h	Enable I2C clock blocking for own address 2. When the CPU sets a bit location to 1, if an external master using the corresponding own address addresses the core, the core will block the I2C clock right after the address phase. For releasing the I2C clock the CPU must write 0 in the corresponding field. Value after reset is low. 0h = I2C clock released 1h = I2C clock blocked
1	OA1_EN	R/W	0h	Enable I2C clock blocking for own address 1. When the CPU sets a bit location to 1, if an external master using the corresponding own address addresses the core, the core will block the I2C clock right after the address phase. For releasing the I2C clock the CPU must write 0 in the corresponding field. Value after reset is low. 0h = I2C clock released 1h = I2C clock blocked

Table 22-39. I2C_SBLOCK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	OA0_EN	R/W	0h	Enable I2C clock blocking for own address 0. When the CPU sets a bit location to 1, if an external master using the corresponding own address addresses the core, the core will block the I2C clock right after the address phase. For releasing the I2C clock the CPU must write 0 in the corresponding field. Value after reset is low. 0h = I2C clock released 1h = I2C clock blocked

HDQ/1-Wire Interface

This chapter describes the HDQ/1-wire interface of the device.

Topic **Page**

23.1 Introduction

23.1.1 HDQ1W Features

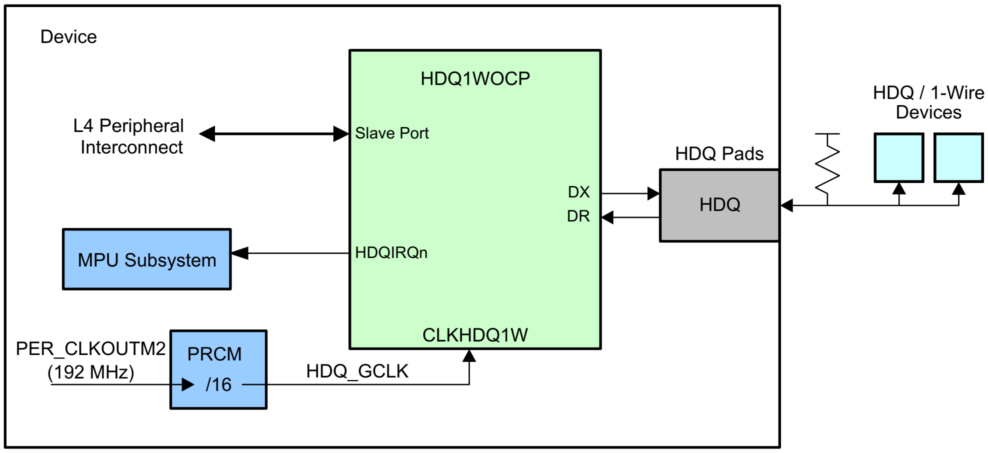
The main features of the HDQ1W include:

- Software selectable HDQ or 1-Wire mode
- 1-Wire single-bit mode
- Return-to-one accomplished via external pull-up
- Timeout monitor
- Interrupt to indicate Tx/Rx completion or timeout

23.2 Integration

The HDQ / 1-Wire module (HDQ1W) implements the hardware protocol of a Benchmarq HDQ and Dallas Semiconductor (Maxim) 1-Wire® master function.

Figure 23-1. HDQ1W Integration



23.2.1 *HDQ1W Connectivity Attributes*

The general connectivity attributes for the HDQ1W module are shown in Table 23-1.

Table 23-1. HDQ1W Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	L4PER_L4LS_GCLK
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart idle
Interrupt Requests	1 interrupt to MPU Subsystem (HDQINT)
DMA Requests	None
Physical Address	L4 peripheral slave port

23.2.2 *HDQ1W Clock and Reset Management*

The HDQ1W has an OCP clock and a separate functional clock.

Table 23-2. HDQ1W Clock Signals

Clock Signal	Maximum Frequency	Reference Source	Comments
CLK Interface / functional clock	100 MHz	CORE_CLKOUTM4 / 2	l4per_l4ls_gclk from PRCM
CLKHDQ1S Functional clock	12 MHz	PER_CLKOUTM2 / 16	pd_per_hdq_gclk from PRCM

23.2.3 *HDQ1W Pin List*

The HDQ1W interface uses a single wire for command and data as summarized in Table 23-3. The signal is implemented as oped-drain and requires an external pulllup for return-to-one when not being driven by the master or slave device.

Table 23-3. HDQ1W Pin List

Pin	Type	Description
HDQ	I/O	HDQ / 1-Wire command/data wire

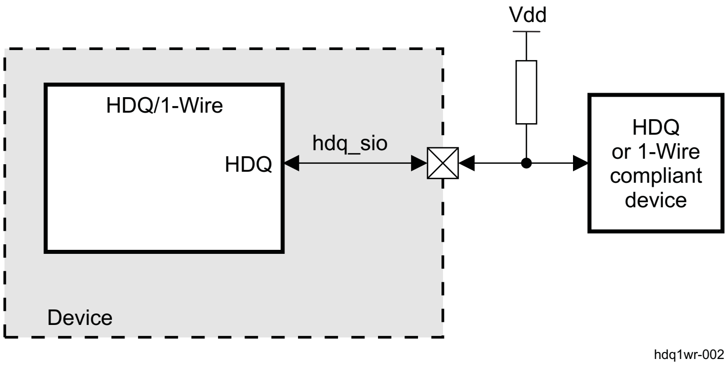
23.3 Functional Description

The HDQ/1-Wire module works with both the HDQ and 1-Wire protocols. The protocols use a single wire to establish communication between the master and the slave. Both protocols use a return-to-1 mechanism; that is, after any command is driven, the line is pulled to a high level. This mechanism requires an external pullup.

23.3.1 HDQ/1-Wire Functional Interface

Figure 23-2 shows a typical application using the HDQ/1-Wire connection.

Figure 23-2. HDQ/1-Wire Typical Application System Overview



An external pullup is required, because the two protocols use a return-to-1 mechanism (that is, logical '1' level is applied by a pullup after any drive-to-zero by a HDQ/1-Wire master or slave). Pullup is typically implemented as a resistor connected between HDQ/1-Wire line and the I/O power supply.

The HDQ/1-Wire module operates according to a command structure that is programmed into transmit command registers (as described in Section 23.4).

The 1-Wire mode runs at slower speeds than the capabilities of the mode.

Table 23-4 describes the external signal for connection with HDQ or 1-Wire compliant devices.

Table 23-4. I/O Description

Signal Name	I/O	Description	Value at Reset
hdq_sio	Bidir	Serial data input/output. Output is open drain type.	0

23.3.2 HDQ and 1-Wire (SDQ) Protocols

23.3.2.1 HDQ Protocol Initialization (Default)

In HDQ mode, the firmware does not require the host to create an initialization pulse to the slave. However, the slave can be reset by using an initialization pulse (also referred to as a break pulse). The initialization pulse is generated by setting the INITIALIZATION bit (HDQ1W_CTRL_STS[2]). The slave does not respond with a presence pulse as it does in the 1-Wire protocol.

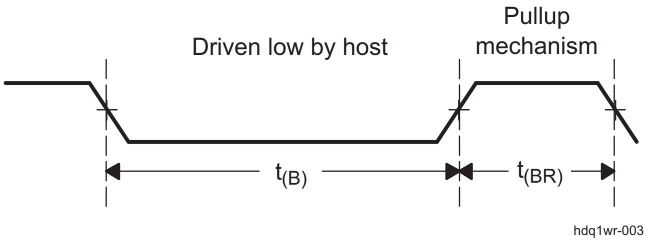
The HDQ is a command-based protocol in which the host sends a command byte to the slave. The command directs the slave either to store the next eight bits of data received to a register specified by the command byte (write operation) or to output the eight bits of data from a register specified by the command byte (read operation). The master implementation is a simple byte engine. The sending of the ID, command/address, and data is controlled by firmware. The master engine provides only a single HDQ1W_TX_DATA register.

The command and data bytes consist of a stream of eight bits with a maximum transmission rate of 5K bits/s. The least significant bit (LSB) of a command or data byte is transmitted first. If a communication time-out occurs between the host and the slave (for example, if the host waits longer than the specified time for the slave to respond, or if this is the first access command), then the host must send an initialization pulse (BREAK pulse) before sending the command again.

The slave detects a break when the HDQ pin is driven to a logic-low state for a specified break time t(B) or greater. The HDQ pin then returns to its normal ready-high logic state for a specified break-recovery time t(BR). The slave is then ready for a command from the host processor. Figure 23-3 shows this behavior.

An interrupt condition indicates either a TX-complete, an RX-complete, or a time-out condition. Reading the interrupt status register clears all interrupt conditions. Only one interrupt signal is sent to the microcontroller, and only one overall mask bit can enable or disable the interrupt. The interrupt conditions cannot be individually masked.

Figure 23-3. HDQ Break-Pulse Timing Diagram

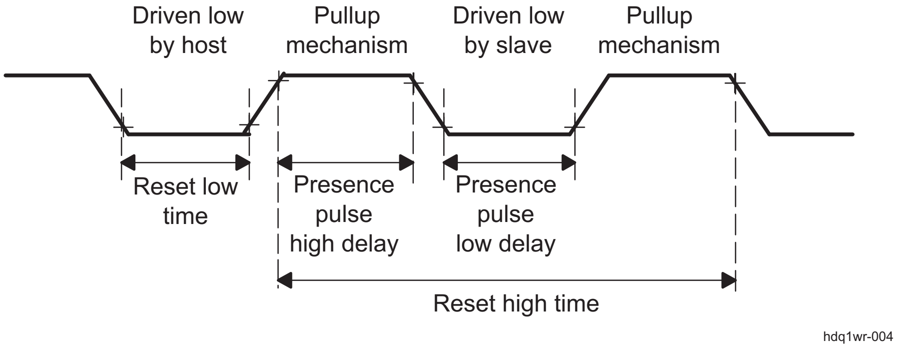


23.3.2.2 1-Wire (SDQ) Protocol Initialization

In 1-Wire (SDQ) protocol, the host first sends an initialization pulse (by pulling the line to a logic-low state) and then waits for the slave to respond with a presence pulse before enabling any communication sequence.

As for the initialization pulse, the presence pulse is a low-going edge on the line initiated by the slave. The timing diagram in Figure 23-4 shows the 1-Wire (SDQ) reset sequence.

Figure 23-4. 1-Wire (SDQ) Reset Timing Diagram



The host drives the line to a logic-low state for a minimum of reset low time. Once the slave detects this pulse, it must drive the line to a logic-low state within the presence pulse high delay for a minimum period of presence pulse low time.

If the slave does not respond within this interval of time, a time-out event occurs and no transaction can be initiated. The host must initiate the reset sequence again before sending any command to the slave.

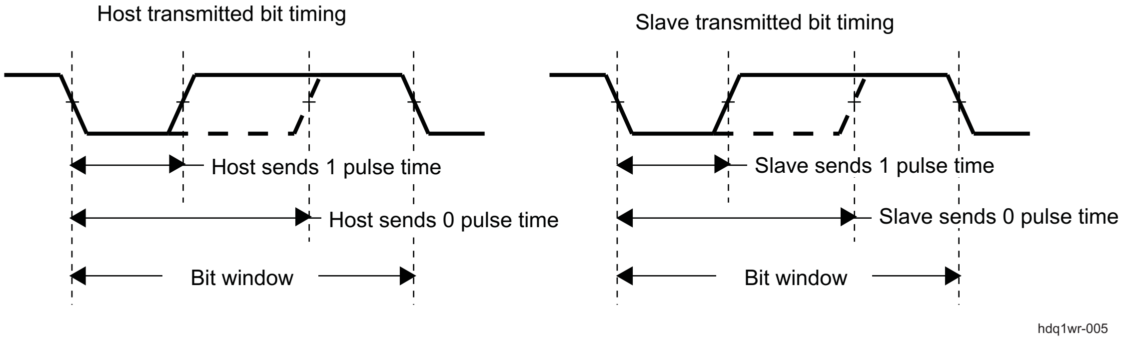
On the other hand, if the slave sends back its presence pulse within the specified interval of time, the communication can be enabled after the reset high time.

23.3.2.3 Communication Sequence (HDQ and 1-Wire Protocols)

This section describes the part common to both protocols.

After a successful break pulse (HDQ mode) or initialization sequence (1-Wire protocol), the host and slave are ready for bit transmission. Each bit to transmit (either from the host to the slave or from the slave to the host) is preceded by a low-going edge on the line, as shown in Figure 23-5.

Figure 23-5. HDQ/1-Wire Transmitted Bit Timing



The return-to-1 data-bit frame consists of three distinct sections. The first section starts the transmission when either the slave or the host takes the line to a logic-low state. The next section is the actual data transmission in which the data must be valid during a specified period of time after the negative edge that starts the communication. The final section stops the transmission by returning the HDQ/1-Wire line to a logic-high state. Communication with an HDQ/1-Wire slave always occurs with the LSB being transmitted first.

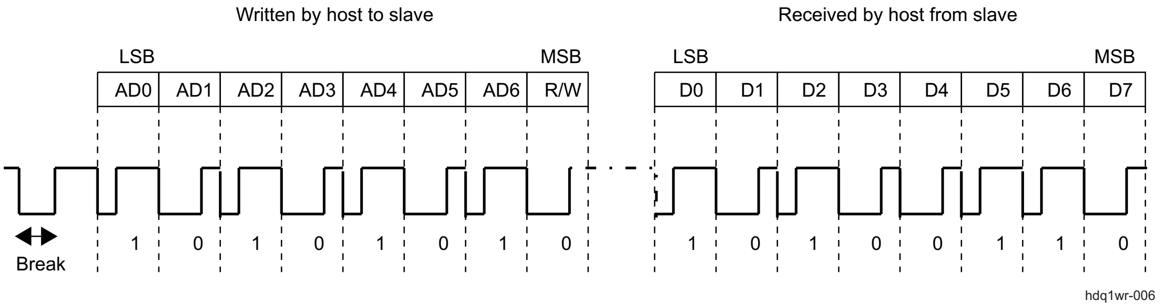
The command byte of the HDQ/1-Wire protocols consists of eight contiguous valid command bits. The command byte contains two fields: R/W command and address. The R/W bit of the command byte determines whether the command is a read or a write, and the address field containing bits AD6-AD0 indicates the address to be read or written. Table 23-5 lists the command byte values.

Table 23-5. HDQ/1-Wire Command Byte

7	6	5	4	3	2	1	0
R/W	AD6	AD5	AD4	AD3	AD2	AD1	AD0

R/W	Indicates whether the command byte is a read or a write. A 1 indicates a write command; the following eight bits must be written to the register specified by the address field of the command byte. A 0 indicates that the command is a read. On a read command, the slave outputs the requested register contents.
AD6-AD0	Represent the seven bits labeled AD6-AD0 containing the address portion of the register to be accessed. The communication sequence example in Figure 23-6 shows a read command at address 0x55; the received data is 0x65.

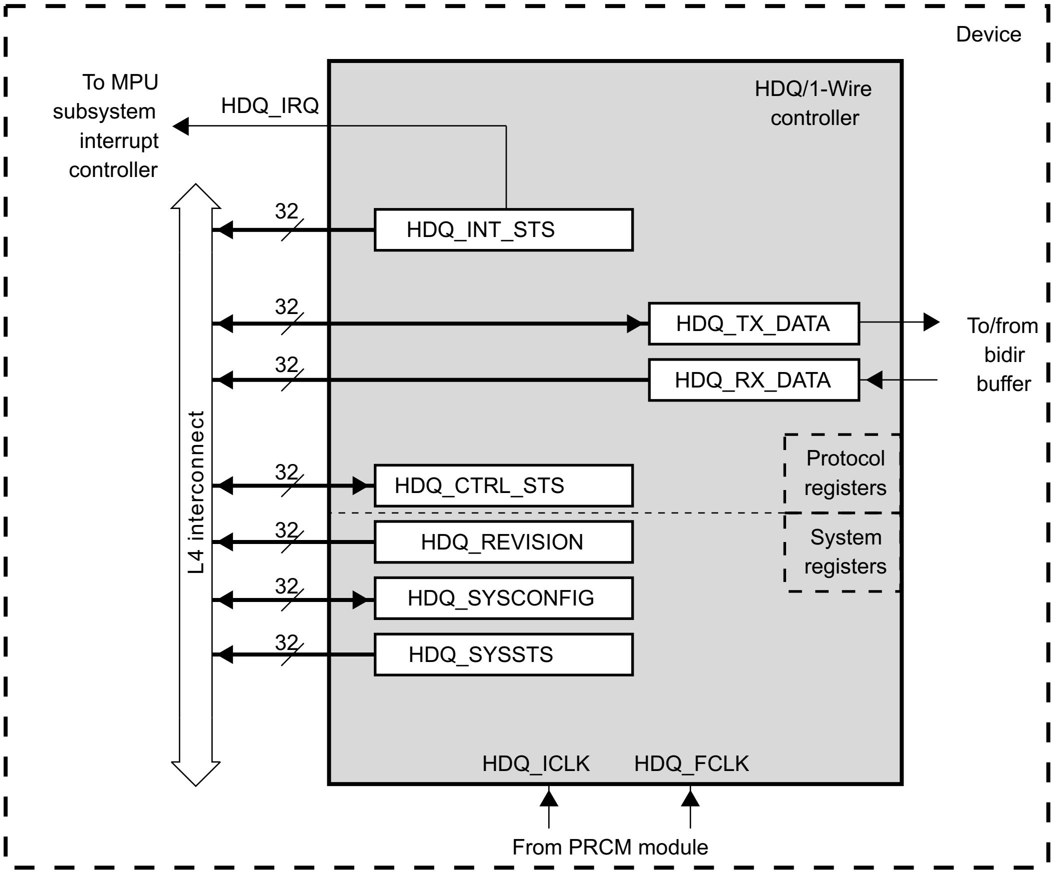
Figure 23-6. HDQ/1-Wire Communication Sequence



23.3.3 HDQ/1-Wire Block Diagram

Figure 23-7 shows the HDQ/1-Wire block diagram.

Figure 23-7. HDQ/1-Wire Block Diagram

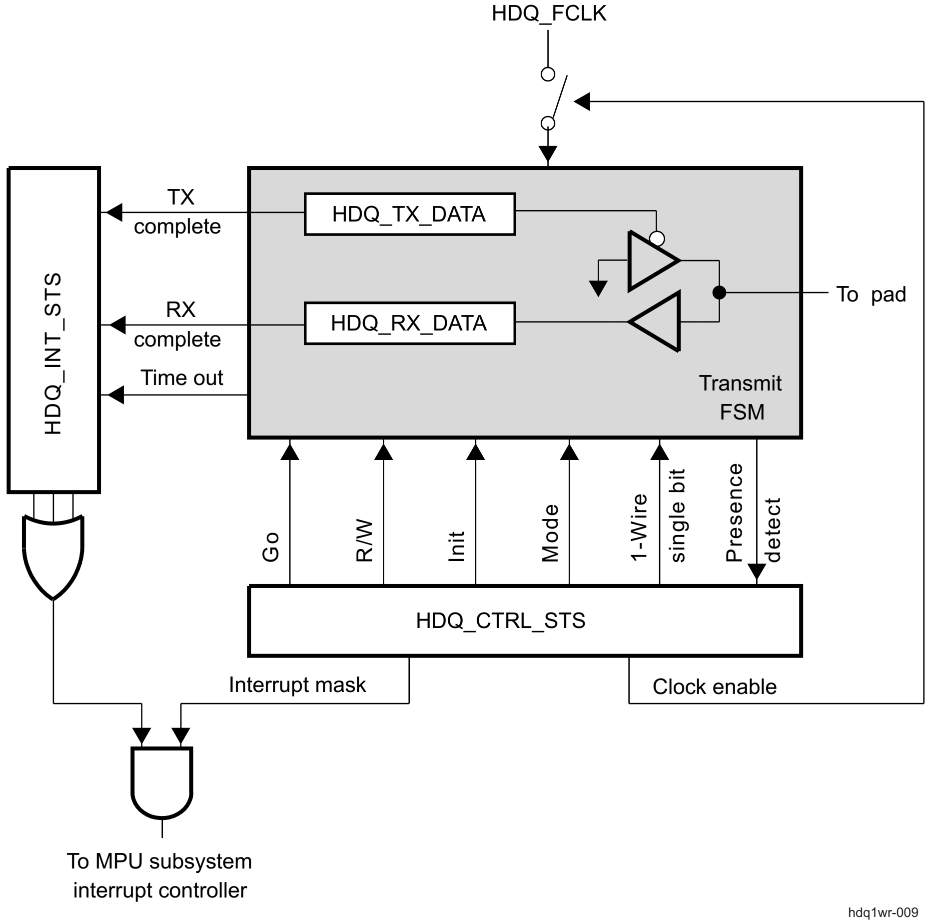


hdq1wr-008

The MODE bit HDQ1W_CTRL_STS[0] allows selection between the HDQ and 1-Wire protocols. This bit is assumed static for design purposes. The configuration is in HDQ mode by default.

Figure 23-8 shows the protocol-dedicated register scheme.

Figure 23-8. Protocol Registers Description



The receive and transmit operations are performed with respect to the HDQ protocol timing. The module is implemented once in the device and is clocked with a single clock whether it runs HDQ or 1-Wire protocol. Although the data exchange is slower than the capabilities of the 1-Wire mode, this mode still meets the timing requirements and practical considerations. That is, the 1-Wire protocol runs at the HDQ protocol speed, which is slower (5K bits/s). The timing parameters and protocol are different in the two modes.

23.3.4 HDQ Mode (Default)

23.3.4.1 HDQ Mode Features

The HDQ mode supports the following:

- Benchmark HDQ protocol
- Power-down mode

23.3.4.2 Description

In the HDQ mode, there is no need for the host to create an initialization pulse to the slave. However, the host can reset the slave by using an initialization pulse (also known as a break pulse). Setting the INITIALIZATION bit HDQ1W_CTRL_STS[2] creates this pulse by pulling the line down. When the slave receives the pulse, it is ready for communication but does not respond with a presence pulse.

In a typical write operation, two bytes are sent to the slave. The first byte corresponds to the command/address byte, and the second byte corresponds to the data to be written.

In a typical read operation, the host sends a command/address byte and the slave returns a byte of data.

The master is implemented to send and receive bytes. Sending the command/address and data is controlled by the firmware. The master provides only a single data TX register.

The HDQ protocol is a return-to-1 protocol. Consequently, after a byte is sent to the slave (either command/address + data for a write, or just command/address for a read), the host pulls the line up. The line is set to the high-impedance state in the device and an external pullup brings it to a logical high level.

In the case of a read operation, the slave also drives the line to a logic-low state before sending the requested data.

If the host initiates a read and does not receive data within a specified interval of time (that is, the slave does not drive the line low within this interval), the TIMEOUT bit HDQ1W_INT_STS[0] is set, thereby indicating a read failure. The TIMEOUT bit remains set until the host reads the interrupt status register (HDQ1W_INT_STS).

An interrupt condition indicates either a TX-complete, an RX-complete, or a time-out on a transaction. The corresponding bit is set in the interrupt status register (HDQ1W_INT_STS). This register is cleared as soon as it is read.

Only one interrupt signal is sent to the MPU, and only an overall mask can enable or disable the interrupts. These interrupts cannot be individually masked.

23.3.4.3 Single-Bit Mode

In HDQ mode, the single-bit mode (1_WIRE_SINGLE_BIT bit HDQ1W_CTRL_STS[7] set to 1) has no effect because the HDQ protocol supports only byte transfers.

23.3.4.4 Interrupt Conditions

The HDQ/1-Wire module provides the following interrupt status:

1. Transmission complete:

   A write operation of one byte was completed. Successful or failed completion is not indicated, because there is no acknowledgment from the slave in HDQ protocol. This interrupt condition is cleared by reading the interrupt status register (HDQ1W_INT_STS).

2. Read complete:

   In HDQ mode, the interrupt status indicates that a byte has been successfully read. This interrupt condition is cleared by reading the interrupt status register (HDQ1W_INT_STS).

3. Presence detect/time-out:

   In HDQ mode, the interrupt status indicates that after a read command initiated by the host, the slave did not pull the line low within the specified time. This interrupt condition is cleared by reading the interrupt status register (HDQ1W_INT_STS).

   In HDQ mode, a time-out condition is also used to indicate the successful completion of a break pulse. That is, if the master has sent the break pulse, it is indicated with a time-out instead of a TX-complete.

Only one interrupt is generated to the MPU based on any of these interrupt conditions. A read operation on the interrupt status register clears all the interrupt status bits that were previously set.

23.3.5 1-Wire Mode

23.3.5.1 1-Wire Mode Features

The 1-Wire mode supports the following:

- Dallas Semiconductor 1-Wire protocol
- Power-down mode
- Single-bit mode

23.3.5.2 Description

The 1-Wire mode requires an initialization pulse to be sent to the slave(s) connected on the interface. If a slave is present, it responds with a presence pulse.

The initialization pulse is sent after INITIALIZATION bit HDQ1W_CTRL_STS[2] is set. The firmware sends the initialization pulse depending on the value of this bit.

The slave presence on the line is detected by a presence bit in the control and status register. When the slave receives the initialization pulse, it sends back its presence pulse by pulling down the line. The module detects this low-going edge and sets the PRESENCEDETECT bit HDQ1W_CTRL_STS[3].

In a similar way, if a presence pulse is not received from the slave after an initialization pulse is sent, the PRESENCEDETECT bit remains cleared.

Whether or not a presence pulse is detected after an initialization pulse is sent, the TIMEOUT bit HDQ1W_INT_STS[0] is set and an interrupt condition is generated.

In 1-Wire mode, the generated interrupt condition means the maximum time allowed for receiving the response has elapsed and the software must check the PRESENCEDETECT bit to determine whether or not there was a presence pulse.

The INITIALIZATION bit is cleared at the end of the initialization pulse at the same time as the TIMEOUT bit is set. The TIMEOUT bit is cleared when the interrupt status register (HDQ1W_INT_STS) is read.

For read operations, 1-Wire is a bit-by-bit protocol, which means the slave must be clocked by the host for each bit of the byte to read.

The line is pulled up at the end of the command/address byte. On the first read, the host creates a low-going edge to initiate a bit read. The line is then pulled up (pulled to the high-impedance state by the host and set to a high logical level by the external pullup) and the slave either drives the line low to transmit a 0 or does not drive the line to transmit a 1. This sequence is repeated for each bit to read.

The first bit the host receives is the LSB, and the last bit is the most significant bit (MSB) in the receive data register (HDQ1W_RX_DATA).

An interrupt condition indicates either a TX-complete, an RX-complete, or a time-out condition (that is, the time allowed for the slave to indicate its presence has elapsed). A read operation on the interrupt status register clears the interrupt conditions previously set. As in the HDQ mode, only one interrupt signal is sent to the MPU. Only an overall mask bit can enable or disable the interrupt (the interrupt conditions cannot be masked individually).

23.3.5.3 1-Wire Single-Bit Mode Operation

A single-bit mode can be entered by setting the appropriate bit in the control and status register (1_WIRE_SINGLE_BIT bit HDQ1W_CTRL_STS[7]). In this mode, only one bit of data at a time is transferred between the master and the slave. After the bit is transferred, an interrupt is generated (that is, there is an RX-complete for a read operation and a TX-complete for a write operation). Bit 0 of the RX register (HDQ1W_RX_DATA) is updated each time a bit is received from the slave; bit 0 of the TX register (HDQ1W_TX_DATA) contains the bit to be sent.

23.3.5.4 Interrupt Conditions

The HDQ/1-Wire module provides the following interrupt status:

1. Transmission complete:

   A write operation of one byte was completed. Successful or failed completion is not indicated, because there is no acknowledgment from the slave in 1-Wire protocol. This interrupt condition is cleared by reading the interrupt status register (HDQ1W_INT_STS).
2. Read complete:

   In 1-Wire mode, the interrupt status indicates that a byte has been successfully read. This interrupt condition is cleared by reading the interrupt status register (HDQ1W_INT_STS).
3. Presence detect/time-out:

   In 1-Wire mode, the interrupt status indicates that it is now valid to check the PRESENCEDETECT bit. This interrupt condition is cleared by reading the interrupt status register (HDQ1W_INT_STS).

Only one interrupt is generated to the MPU based on any of these interrupt conditions. A read operation on the interrupt status register clears all interrupt status bits that were previously set.

23.3.5.5 Status Flags

The presence-condition-detected status flag is contained in the PRESENCEDETECT bit HDQ1W_CTRL_STS[3]. This is valid only in 1-Wire mode. The flag is updated when TIMEOUT bit HDQ1W_INT_STS[0] is set. Therefore, its correct value shows only after the interrupt is generated. The firmware must wait for the time-out condition; otherwise, the flag keeps its previous value and is undefined.

23.3.6 Module Power Saving

23.3.6.1 Autoidle Mode

The HDQ/1-Wire module provides an autoidle function in its interconnect clock domain.

The interconnect clock autoidle power-saving mode is enabled or disabled through the AUTOIDLE bit HDQ1W_SYSCONFIG[0]. When this mode is enabled and there is no activity on the interconnect interface, the interconnect clock (HDQ_ICLK) is disabled inside the module, thereby reducing power consumption. When there is new activity on the interconnect interface, the interconnect clock is restarted with no latency penalty. This mode is disabled by default after a reset.

This mode is recommended to reduce power consumption.

23.3.6.2 Power-Down Mode

The HDQ/1-Wire module also provides a power-saving function in its functional clock domain.

Setting the CLOCKENABLE bit in the control and status register (CLOCKENABLE bit HDQ1W_CTRL_STS[5]) to 0 shuts off the functional clock (HDQ_FCLK) to the state-machine. The state-machine is reset when the functional clock is disabled; if any transaction is ongoing, it is aborted into the reset state.

The register values are not affected by disabling the functional clock.

Do not access the module registers after the software has put the module in power-down mode except to write to the CLOCKENABLE bit to take the module out of power-down mode.

23.3.7 System Power Management and Wakeup

As part of the system-wide power-management scheme, the HDQ/1-Wire module can go into idle state at the request of the PRCM module (for more information, see Chapter 6, *Power, Reset, and Clock Management*). However, the HDQ/1-Wire module does not support handshake protocol with the PRCM. The HDQ/1-Wire module can go into idle mode only as part of the L4 interconnect clock domain (both CORE_L4_ICLK and FUNC_12M_CLK belong to the L4 interconnect clock domain).

When the AUTO_HDQ bit PRCM.CM_AUTOIDLE1_CORE[22] is set, the HDQ/1-Wire module behavior follows the L4 interconnect clock domain behavior. If the whole domain is put into idle, the HDQ/1-Wire is also put into idle.

Software must ensure correct clock management.

CAUTION

There is no hardware mechanism to prevent cutting off the HDQ/1-Wire clocks while the module is performing a transfer. The result would be a loss of data being transferred.

23.4 Programming Model

The HDQ/1-Wire module can be considered a simple byte engine because it only implements the hardware interface layer for both HDQ and 1-Wire protocols. The correct sequencing is controlled by the firmware, which is described in this section.

23.4.1 Module Initialization Sequence

23.4.1.1 Mode Selection

MODE bit HDQ1W_CTRL_STS[0] allows selection between the HDQ and 1-Wire protocols. When set to 0, the protocol is HDQ; when set to 1, the protocol is 1-Wire. The bit is assumed static for design purposes. The configuration is in HDQ mode by default.

Although this bit can be modified at any point, it is strongly recommended that it be modified only as part of the boot-up configuration.

23.4.1.2 Reset/Initialization

No slave presence test is required in HDQ mode; however, the slave can be reset by setting the INITIALIZATION bit HDQ1W_CTRL_STS[2] and the GO bit HDQ1W_CTRL_STS[4]. The line is then pulled down (break pulse) and the bit returns to 0 after the pulse is sent. Upon completion, a time-out interrupt is also generated. The slave does not respond to this pulse.

In 1-Wire mode, the slave returns a presence pulse when it receives the initialization pulse. The protocol for initialization is as follows:

1. Set the INITIALIZATION bit HDQ1W_CTRL_STS[2] to 1 and the GO bit HDQ1W_CTRL_STS[4] to 1 to send the pulse. When the pulse is sent, the bit is cleared in the register.
2. Wait for the presence detect flag (TIMEOUT bit HDQ1W_INT_STS[0]) to generate an interrupt. This flag is set when the response time allowed to the slave has elapsed, whether it has sent a pulse or not.
3. Read the HDQ1W_CTRL_STS register to check whether the presence pulse has been received before starting any transfer.

23.4.2 HDQ Protocol Basic Programming Model

23.4.2.1 Write Operation

The write operation sequence is as follows:

1. Write the command/address or data value to the TX write register (HDQ1W_TX_DATA).

NOTE: Steps 2 and 3 can be performed simultaneously.

2. Set DIR bit HDQ1W_CTRL_STS[1] to indicate a write.
3. Set GO bit HDQ1W_CTRL_STS[4] to start the transmission.
   a. The hardware sends the byte from the TX write register.
   b. In a write operation, the TIMEOUT bit is always cleared, because there is no acknowledge mechanism from the slave.
   c. When the write operation is completed, the TX-complete flag is set in the interrupt status register (TXCOMPLETE bit HDQ1W_INT_STS[2]). If interrupts are masked (that is, the corresponding bit has been previously set in the control and status register), no interrupt signal is generated.
   d. The GO bit HDQ1W_CTRL_STS[4] is cleared at the end of a write operation.
4. The software must read the interrupt status register to clear the interrupt.
5. Repeat step 1 through step 4 for each successive byte to write.

23.4.2.2 Read Operation

The read operation sequence is as follows:

1. Write the command value to the TX write register (HDQ1W_TX_DATA).
2. Set DIR bit HDQ1W_CTRL_STS[1] to 0 and GO bit HDQ1W_CTRL_STS[4] to 1 and wait for the TX-complete interrupt.

NOTE: Steps 3 and 4 can be performed simultaneously.

3. Set DIR bit HDQ1W_CTRL_STS[1] to 1 to indicate a read.
4. Write 1 to GO bit HDQ1W_CTRL_STS[4] to initiate the read.
    a. The hardware detects a low-going edge on the line (generated by the slave) and receives 8 bits of data in the RX receive buffer register (HDQ1W_RX_DATA). The first bit received is the LSB and the last bit is the MSB. The master performs this step as soon as the slave sends the data, irrespective of the state of GO bit HDQ1W_CTRL_STS[4]. However, an RX-complete interrupt is generated only when the software writes the GO bit.
    b. If a time-out occurs, the TIMEOUT bit HDQ1W_CTRL_STS[0] is set.
    c. Completion of the operation is indicated by setting the RX-complete flag in the interrupt status register (RXCOMPLETE bit HDQ1W_INT_STS[1]. If interrupts are masked (that is, the corresponding bit was previously set in the control and status register), no interrupt signal is generated.
    d. At the end of the read operation, the GO bit HDQ1W_CTRL_STS[4] is cleared. It is also cleared if a time-out is detected
5. The software must read the interrupt status register (HDQ1W_INT_STS) to determine whether an RX was successfully completed or a time-out occurred.
6. The software reads the RX receive buffer register (HDQ1W_RX_DATA) to retrieve the read data from the slave.
7. Repeat step 1 through step 6 for each successive byte.

NOTE: In HDQ mode, the address/command is written only once to the slave. However, after the first byte is received, an RX-complete interrupt is set. Therefore, the software must initiate the read of the second byte by setting the GO bit in the control and status register. The first byte received is shadowed and provided to the software while the hardware is fetching the second byte of data.

23.4.3 1-Wire Mode (SDQ) Basic Programming Model

23.4.3.1 Write Operation

The write operation sequence is as follows:

1. Write the ID, command, or data value to the TX write register (HDQ1W_TX_DATA).

NOTE: Steps 2 and 3 can be performed simultaneously.

2. Set DIR bit HDQ1W_CTRL_STS[1] to 0 to indicate a write operation.
3. Set GO bit HDQ1W_CTRL_STS[4] to 1 to start the transmission.
    a. The hardware sends the byte stored in the TX write register.
    b. In a write operation, the TIMEOUT bit is always cleared.
    c. When the operation is completed, the TX-complete flag is set in the interrupt status register (TXCOMPLETE bit HDQ_INT_STS [2]). No interrupt signal is generated if interrupts are masked (that is, the corresponding bit was previously set in the control and status register).
    d. The GO bit of the control and status register is cleared at the end of a write operation.
4. The software must read the interrupt status register (HDQ1W_INT_STS) to clear the interrupt.
5. Repeat step 1 through step 4 for each successive byte to write.

23.4.3.2 Read Operation

The read operation sequence is as follows:

1. Write the address to the TX write register (HDQ1W_TX_DATA).
2. Set DIR bit HDQ1W_CTRL_STS[1] to 0 and the GO bit HDQ1W_CTRL_STS[4] to 1 and wait for the TX-complete interrupt flag.
3. Write the command value to the TX write register (HDQ1W_TX_DATA).
4. Set DIR bit HDQ1W_CTRL_STS[1] to 0 and GO bit HDQ1W_CTRL_STS[4] to 1 and wait for the TX-complete interrupt flag.

NOTE: Steps 5 and 6 can be performed simultaneously.

5. Set DIR bit HDQ1W_CTRL_STS[1] to 1 to indicate a read.
6. Set GO bit HDQ1W_CTRL_STS[4] to 1 to start the transmission.
   a. The hardware (master) generates a low-going edge and clocks 8 bits of data into the RX receive register (HDQ1W_RX_DATA). The first bit received is the LSB and the last bit is the MSB.
   b. TIMEOUT bit HDQ1W_INT_STS[0] is always cleared in a read operation.
   c. When the operation is complete, the RX-complete flag is set in the interrupt status register (RXCOMPLETE bit HDQ1W_INT_STS[1]). No interrupt signal is generated if the interrupts are masked (that is, the corresponding bit was previously set in the control and status register).
   d. GO bit HDQ1W_CTRL_STS[4] is cleared at the end of the read. It is also cleared if a time-out occurs.
7. The software must read the interrupt status register to determine whether an RX was successfully completed or a time-out occurred.
8. The software reads the RX receive buffer register (HDQ1W_RX_DATA) to retrieve the read data from the slave.
9. Repeat step 1 through step 8 for each successive byte.

23.4.3.3 1-Wire Bit Mode Operation

Select the single-bit mode by setting the 1_WIRE_SINGLE_BIT bit HDQ1W_CTRL_STS[7] to 1. In this mode, only one bit of data at a time is transferred between the master and the slave. After the bit is transferred, the corresponding interrupt flag is set (that is, there is an RX-complete (RXCOMPLETE bit HDQ1W_INT_STS[1]) for a read operation and a TX-complete (TXCOMPLETE bit HDQ1W_INT_STS[2]) for a write operation). Bit 0 of the RX register (HDQ1W_RX_DATA) is updated each time a bit is received; bit 0 of the TX register (HDQ1W_TX_DATA) contains the bit of data to be sent.

23.4.4 Power Management

The software has independent control of the two clock domains (interconnect clock: HDQ_ICLK and functional clock: HDQ_FCLK). Because there is no acknowledge mechanism from the HDQ/1-Wire module to an idle request, the software must ensure that a clock is not shut off while a transfer is being processed (the data would be lost).

If the autoidle function (AUTOIDLE bit HDQ1W_SYSCONFIG[0] set to 1) provides a safe transfer (the module wakes up the HDQ_ICLK as soon as a transfer is initiated), the power-down mode and the PRCM idle requests (through the whole L4 clock domain idle request) must be handled carefully.

The following sections describe the steps to follow to use the power-down and idle modes.

23.4.4.1 Module Power-Down Mode

1. Before shutting off the HDQ_FCLK, wait for an RX-complete or a TX-complete interrupt.
   - In a read operation, the transfer is completed when the RX-complete flag (RXCOMPLETE bit HDQ1W_INT_STS [1]) generates an interrupt.
   - In a write operation, the transfer is completed when the TX-complete flag (TXCOMPLETE bit HDQ1W_INT_STS [2]) generates an interrupt. The software must check whether the interrupt was generated after the address/command byte was sent or after the data byte was sent. The clock must not be shut off after the command/ address byte is sent; otherwise, the data is not written to the slave.

   HDQ1W_INT_STS must be read to clear the interrupt condition.
2. Set the CLOCKENABLE bit HDQ1W_CTRL_STS[5] to 0 to disable the clock.

   Do not access the module registers after the software has put the module into power-down mode except to write to the clock-enable bit to take the module out of power-down mode.

23.4.4.2 System Idle Mode

This section describes the steps to follow at the module level before enabling the idle mode at the system level (for more information about the system power management scheme, see Chapter 6, *Power, Reset, and Clock Management*.

As part of the L4 interconnect clock domain, the HDQ/1-Wire clocks can be cut at the PRCM level. HDQ_FCLK can be cut if the EN_HDQ bit PRCM.CM_FCLKEN1_CORE [22] is set to 0 and no other modules require CORE_12M_FCLK. The software must verify that no transfer is in progress.

In a read operation:

1. Wait for an RX-complete interrupt.

   In a read operation, the transfer is completed when the RX-complete flag (RXCOMPLETE bit HDQ1W_INT_STS [1]) generates an interrupt.
2. Read the HDQ1W_INT_STS to clear the read-complete interrupt flag.
3. Read the HDQ1W_RX_DATA to retrieve the read data.
4. The HDQ_ICLK can be shut off by entering the system idle mode.

In a write operation:

1. Wait for a TX-complete interrupt.

   In a write operation, the transfer is completed when the TX-complete flag (TXCOMPLETE bit HDQ1W_INT_STS[2]) generates an interrupt. The software must check whether the interrupt was generated after the address/command byte was sent or after the data byte was sent. The clock must not be shut off after the command/address byte is sent; otherwise, the data is not written to the slave.
2. Read the HDQ1W_INT_STS to clear the TX-complete interrupt flag.
3. The HDQ_ICLK can be shut off by entering the system idle mode.

Concerning HDQ_ICLK, two situations can occur:

- The clock is no longer required and EN_HDQ bit PRCM.CM_ ICLKEN1_CORE[22] is set by software. In this case, the clock is cut off, provided no other modules require it. Before setting the EN_HDQ bit, the software must follow the steps described in this section.
- AUTO_HDQ bit PRCM.CM_AUTOIDLE1_CORE[22] is set. In this case, the software must verify that all HDQ/1-Wire transfers are complete before enabling the L4 interconnect clock domain idle mode. Otherwise, the HDQ/1-Wire has no way to prevent the clock from being cut, because no hardware mechanism exists. The steps listed in this section must be verified before putting the L4 clock domain into idle state.

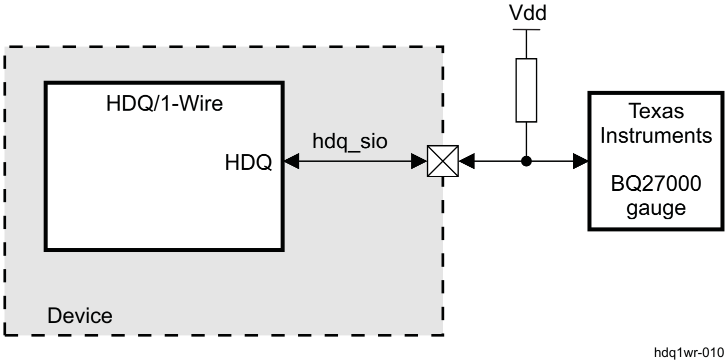
23.5 Use Cases

23.5.1 *How to Configure the HDQ/1-Wire when Connected with a BQ27000 Gauge*

23.5.1.1 Environment

Figure 23-9 details the device connections with the BQ27000 gauge.

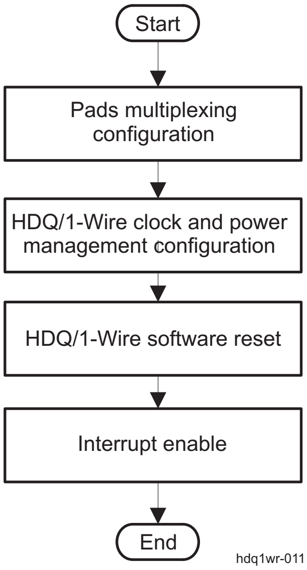
Figure 23-9. Environment



23.5.1.2 Programming Flow

This section details the programming flow of the HDQ/1-Wire. Figure 23-10 shows the main steps of this configuration. The BQ27000 gauge uses the HDQ mode.

Figure 23-10. HDQ/1-Wire Configuration in HDQ Mode



23.5.1.3 Pad Configuration and HDQ/1-Wire Clock and Power Management

Table 23-6 shows the pad multiplexing and the clock and power management configuration to select for the HDQ/1-Wire module.

Table 23-6. Registers Print for HDQ/1-Wire Configuration

Register Name	Address	Value	Value description
CTRL_CONF_UART3_RTSN	0x44E1 0A34	0xxxxx 0001	Configure hdq_sio pad in mode 1
PRCM_CM_PER_HDQ1W_CLKCTRL	0x4DF 8CA0	0x0000 0002	Enable HDQ/1-Wire Functional clock
HDQ_CTRL_STS	0x4834 700C	0x0000 0020	Enable clocks and select the HDQ mode
HDQ_SYSCONFIG	0x4834 70014	0x0000 0000	Module clock is free-running (Disable autoidle mode)

23.5.1.4 HDQ/1-Wire Software Reset

Perform a software reset as described in Figure 23-11.

Figure 23-11. Software Reset Flowchart

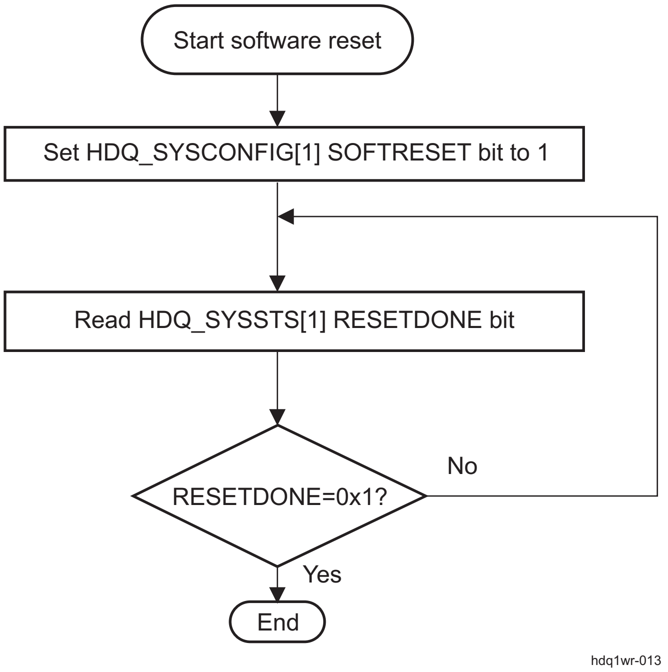


Table 23-7 describes the registers to be configured for the HDQ/1-Wire software reset step.

Table 23-7. Registers Print for HDQ/1-Wire Software Reset

Register Name	Address	Value	Value description
HDQ_SYSCONFIG	0x4834 7014	0x0000 0002	Initiate a software reset. The HDQ_SYSCONFIG[1] SOFTRESET is automatically reset by hardware.
HDQ_SYSSTS	0x4834 7018	0x0000 0001	The HDQ_SYSSTATUS[0] RESETDONE is set to 1 when the reset sequence is done.

23.5.1.5 Interrupts Enable

Table 23-8 describes the registers to be configured for the interrupts enable step and the use case values.

Table 23-8. Registers Print for HDQ/1-Wire Interrupts Enable

Register Name	Address	Value	Value description
HDQ_CTRL_STS	0x4834 700C	0x0000 0060	Enable Interrupts

23.5.1.6 Read and Write Operations

The Read and Write operations in HDQ mode are described in Section 23.4.2.

Some write operations are needed to configure the BQ27000 gauge: for example, it is necessary to write a COMMAND KEY (0xA9 or 0x56) in the Device Control Register of the gauge. For more information, see the TI BQ27000 datasheet (SLUS556).

23.6 HDQ/1-Wire Registers

23.6.1 HDQ1W Registers

Table 23-9 lists the memory-mapped registers for the HDQ1W. All register offset addresses not listed in Table 23-9 should be considered as reserved locations and the register contents should not be modified.

Table 23-9. HDQ1W REGISTERS

Offset	Acronym	Register Name	Section
0h	HDQ1W_REVISION		Section 23.6.1.1
4h	HDQ1W_TX_DATA		Section 23.6.1.2
8h	HDQ1W_RX_DATA		Section 23.6.1.3
Ch	HDQ1W_CTRL_STS		Section 23.6.1.4
10h	HDQ1W_INT_STS		Section 23.6.1.5
14h	HDQ1W_SYSCONFIG		Section 23.6.1.6
18h	HDQ1W_SYSSTS		Section 23.6.1.7

23.6.1.1 HDQ1W_REVISION Register (offset = 0h) [reset = 8h]

Register mask: FFFFFFFFh

HDQ1W_REVISION is shown in Figure 23-12 and described in Table 23-10.

This register contains the IP revision code

Figure 23-12. HDQ1W_REVISION Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">RESERVED</td><td colspan="8">REV</td></tr>
<tr><td colspan="24">R-0h</td><td colspan="8">R-8h</td></tr>
</table>

Table 23-10. HDQ1W_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reads returns 0
7-0	REV	R	8h	IP revision

23.6.1.2 HDQ1W_TX_DATA Register (offset = 4h) [reset = 0h]

Register mask: FFFFFFFFh

HDQ1W_TX_DATA is shown in Figure 23-13 and described in Table 23-11.

This register contains the data to be transmitted.

Figure 23-13. HDQ1W_TX_DATA Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	TX_DATA
R-0h	R/W-0h

Table 23-11. HDQ1W_TX_DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reads returns 0
7-0	TX_DATA	R/W	0h	transmit data

23.6.1.3 HDQ1W_RX_DATA Register (offset = 8h) [reset = 0h]

Register mask: FFFFFFFFh

HDQ1W_RX_DATA is shown in Figure 23-14 and described in Table 23-12.

This register contains the data to be received.

Figure 23-14. HDQ1W_RX_DATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_DATA							
R-0h																								R-0h							

Table 23-12. HDQ1W_RX_DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reads returns 0
7-0	RX_DATA	R	0h	receive data

23.6.1.4 HDQ1W_CTRL_STS Register (offset = Ch) [reset = 0h]

Register mask: FFFFFFFFh

HDQ1W_CTRL_STS is shown in Figure 23-15 and described in Table 23-13.

This register provides status information about the module.

Figure 23-15. HDQ1W_CTRL_STS Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED					BITFSM_DELAY		
R/W-0h					R/W-0h		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ONE_WIRE_SINGLE_BIT	INTRMASK	CLOCKEN	GO	PRESENCEDETECT	INITIALIZATION	DIR	MODE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h

Table 23-13. HDQ1W_CTRL_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	Reads returns 0
10-8	BITFSM_DELAY	R/W	0h	BITFSM delay value
7	ONE_WIRE_SINGLE_BIT	R/W	0h	
6	INTRMASK	R/W	0h	
5	CLOCKEN	R/W	0h	
4	GO	R/W	0h	
3	PRESENCEDETECT	R	0h	
2	INITIALIZATION	R/W	0h	
1	DIR	R/W	0h	
0	MODE	R/W	0h	

23.6.1.5 HDQ1W_INT_STS Register (offset = 10h) [reset = 0h]

Register mask: FFFFFFFFh

HDQ1W_INT_STS is shown in Figure 23-16 and described in Table 23-14.

This register controls interrupts status

Figure 23-16. HDQ1W_INT_STS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED					TXCOMPLETE	RXCOMPLETE	TIMEOUT
R-0h					R-0h	R-0h	R-0h

Table 23-14. HDQ1W_INT_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reads returns 0
2	TXCOMPLETE	R	0h	txcomplete
1	RXCOMPLETE	R	0h	rxcomplete
0	TIMEOUT	R	0h	timeout

23.6.1.6 HDQ1W_SYSCONFIG Register (offset = 14h) [reset = 0h]

Register mask: FFFFFFFFh

HDQ1W_SYSCONFIG is shown in Figure 23-17 and described in Table 23-15.

This register controls various bits

Figure 23-17. HDQ1W_SYSCONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							

23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							

7	6	5	4	3	2	1	0
RESERVED						SOFTRESET	AUTOIDLE
R/W-0h						R/W-0h	R/W-0h

Table 23-15. HDQ1W_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	Reads returns 0
1	SOFTRESET	R/W	0h	When '1', start softreset
0	AUTOIDLE	R/W	0h	OCP idle

23.6.1.7 HDQ1W_SYSSTS Register (offset = 18h) [reset = 1h]

Register mask: FFFFFFFFh

HDQ1W_SYSSTS is shown in Figure 23-18 and described in Table 23-16.

This register monitors the reset sequence.

Figure 23-18. HDQ1W_SYSSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							RESETDONE
R-0h							R-1h

Table 23-16. HDQ1W_SYSSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reads returns 0
0	RESETDONE	R	1h	reset done

Multichannel Audio Serial Port (McASP)

This chapter describes the McASP of the device.

Topic **Page**

24.1 Introduction

24.1.1 Purpose of the Peripheral

The multichannel audio serial port (McASP) functions as a general-purpose audio serial port optimized for the needs of multichannel audio applications. The McASP is useful for time-division multiplexed (TDM) stream, Inter-Integrated Sound (I2S) protocols, and intercomponent digital audio interface transmission (DIT). The McASP consists of transmit and receive sections that may operate synchronized, or completely independently with separate master clocks, bit clocks, and frame syncs, and using different transmit modes with different bit-stream formats. The McASP module also includes serializers that can be individually enabled to either transmit or receive.

24.1.2 Features

Features of the McASP include:

- Two independent clock generator modules for transmit and receive.
  - Clocking flexibility allows the McASP to receive and transmit at different rates. For example, the McASP can receive data at 48 kHz but output up-sampled data at 96 kHz or 192 kHz.
- Independent transmit and receive modules, each includes:
  - Programmable clock and frame sync generator.
  - TDM streams from 2 to 32, and 384 time slots.
  - Support for time slot sizes of 8, 12, 16, 20, 24, 28, and 32 bits.
  - Data formatter for bit manipulation.
- Individually assignable serial data pins (up to 6 pins).
- Glueless connection to audio analog-to-digital converters (ADC), digital-to-analog converters (DAC), codec, digital audio interface receiver (DIR), and S/PDIF transmit physical layer components.
- Wide variety of I2S and similar bit-stream format.
- Integrated digital audio interface transmitter (DIT) supports (up to 10 transmit pins):
  - S/PDIF, IEC60958-1, AES-3 formats.
  - Enhanced channel status/user data RAM.
- 384-slot TDM with external digital audio interface receiver (DIR) device.
  - For DIR reception, an external DIR receiver integrated circuit should be used with I2S output format and connected to the McASP receive section.
- Extensive error checking and recovery.
  - Transmit underruns and receiver overruns due to the system not meeting real-time requirements.
  - Early or late frame sync in TDM mode.
  - Out-of-range high-frequency master clock for both transmit and receive.
  - External error signal coming into the AMUTEIN input.
  - DMA error due to incorrect programming.

24.1.3 Protocols Supported

The McASP supports a wide variety of protocols.

- Transmit section supports:
  - Wide variety of I2S and similar bit-stream formats.
  - TDM streams from 2 to 32 time slots.
  - S/PDIF, IEC60958-1, AES-3 formats.
- Receive section supports:
  - Wide variety of I2S and similar bit-stream formats.
  - TDM streams from 2 to 32 time slots.
  - TDM stream of 384 time slots specifically designed for easy interface to external digital interface receiver (DIR) device transmitting DIR frames to McASP using the I2S protocol (one time slot for each DIR subframe).

The transmit and receive sections may each be individually programmed to support the following options on the basic serial protocol:

- Programmable clock and frame sync polarity (rising or falling edge): ACLKR/X, AHCLKR/X, and AFSR/X.
- Slot length (number of bits per time slot): 8, 12, 16, 20, 24, 28, 32 bits supported.
- Word length (bits per word): 8, 12, 16, 20, 24, 28, 32 bits; always less than or equal to the time slot length.
- First-bit data delay: 0, 1, 2 bit clocks.
- Left/right alignment of word inside slot.
- Bit order: MSB first or LSB first.
- Bit mask/pad/rotate function.
  - Automatically aligns data internally in either Q31 or integer formats.
  - Automatically masks nonsignificant bits (sets to 0, 1, or extends value of another bit).

In DIT mode for McASP, additional features of the transmitter are:

- Transmit-only mode 384 time slots (subframe) per frame.
- Bi-phase encoded 3.3 V output.
- Support for consumer and professional applications.
- Channel status RAM (384 bits).
- User data RAM (384 bits).
- Separate valid bit (V) for subframe A, B.

In I2S mode, the transmit and receive sections can support simultaneous transfers on up to all serial data pins operating as 192 kHz stereo channels.

In DIT mode, the transmitter can support a 192 kHz frame rate (stereo) on up to all serial data pins simultaneously (note that the internal bit clock for DIT runs two times faster than the equivalent bit clock for I2S mode, due to the need to generate Biphase Mark Encoded Data).

24.1.4 Unsupported McASP Features

The unsupported McASP features in this device include the following.

Table 24-1. Unsupported McASP Features

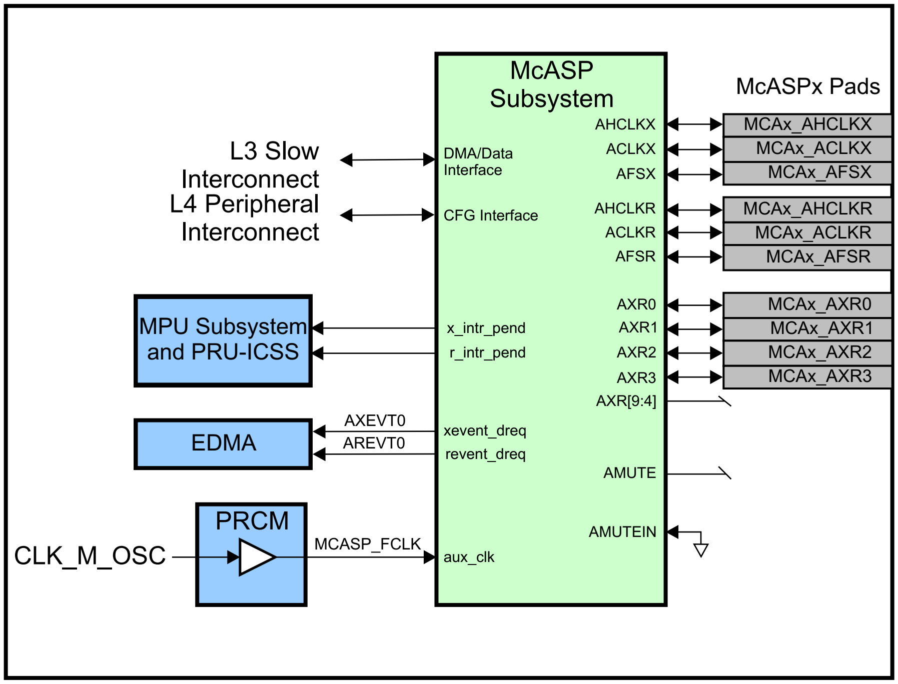
Feature	Reason
AXR[9:4]	Signals are not pinned out
AMUTE	Not connected
AMUTEIN	Not connected

24.2 Integration

The device contains two instantiations of the McASP subsystem: McASP0 and McASP1. The McASP subsystem includes a McASP peripheral, and transmit/receive buffers.

Each McASP is configured with four serializers.

Figure 24-1. McASP0–1 Integration



24.2.1 McASP Connectivity Attributes

The general connectivity attributes for the McASP modules are summarized in Table 24-2

Table 24-2. McASP Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L3S_GCLK (OCP Clock) PD_PER_MCASP_FCLK (Aux Clock)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	1 Transmit Interrupt per instance x_intr_pend - to MPU Subsystem (MCATXINTx) and PRU-ICSS (mcasp_x_intr_pend) 1 Receive Interrupt r_intr_pend - to MPU Subsystem (MCARXINTx) and PRU-ICSS (mcasp_r_intr_pend)
DMA Requests	2 DMA requests per instance to EDMA (Transmit: AXEVTx, Receive: AREVTx)
Physical Address	L3 Slow slave port (data) L4 Peripheral slave port (CFG)

24.2.2 McASP Clock and Reset Management

The McASP module uses functional clocks either generated internally (master mode) or supplied from its serial interface (slave mode). The internal interconnect interface clock is used for the module internal OCP interface. Internal registers select the source of the functional clocks and the applied divider ratio.

Table 24-3. McASP Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
ocp_clk Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l3s_gclk From PRCM
auxclk Functional clock	26 MHz	CLK_M_OSC	pd_per_mcasp_fclk From PRCM

24.2.3 McASP Pin List

The McASP external interface signals are shown in Table 24-4.

Table 24-4. McASP Pin List

Pin	Type	Description
McASP*x*_AXR[3:0]	I/O	Audio transmit/receive pin
McASP*x*_ACLKX(1)	I/O	Transmit clock
McASP*x*_FSX(1)	I/O	Frame synch for transmit
McASP*x*_AHCLKX(1)	I/O	High speed transmit clock
McASP*x*_ACLKR(1)	I/O	Receive clock
McASP*x*_FSR(1)	I/O	Frame synch for receive
McASP*x*_AHCLKR(1)	I/O	High speed receive clock

(1) These signals are also used as inputs to re-time or sync data. The associated CONF_<module>_pin_RXACTIVE bit for these signals must be set to 1 to enable the inputs back to the module. It is also recommended to place a 33-ohm resistor in series (close to the processor) on each of these signals to avoid signal reflections.

24.3 Functional Description

24.3.1 Overview

Figure 24-2 shows the major blocks of the McASP. The McASP has independent receive/transmit clock generators and frame sync generators, error-checking logic, and up to four serial data pins. See the device-specific data manual for the number of data pins available on your device.

All the McASP pins on the device may be individually programmed as general-purpose I/O (GPIO) if they are not used for serial port functions.

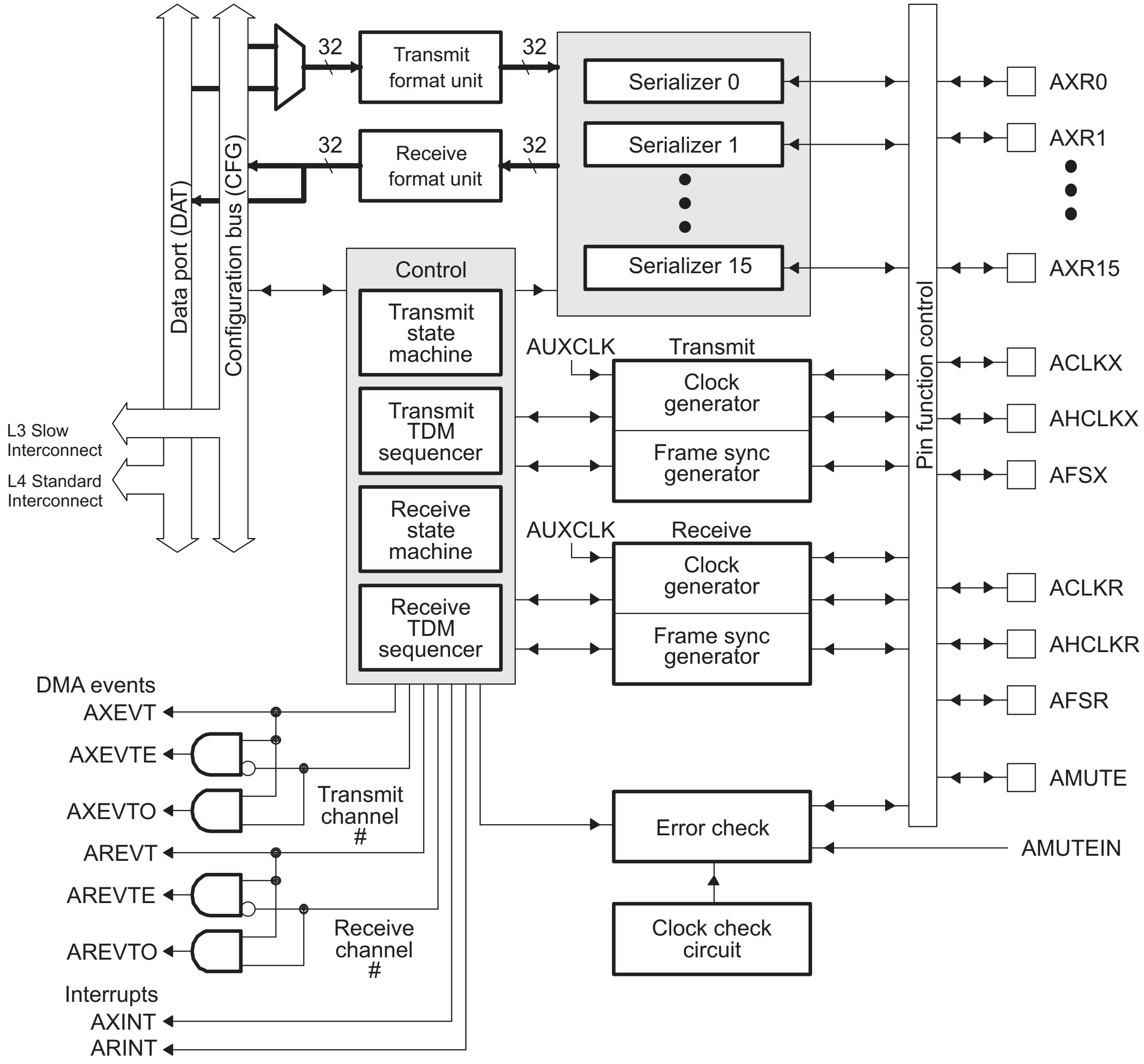
The McASP includes the following pins:

- Serializers;
  - Data pins AXRn: Up to four pins.
- Transmit clock generator:
  - AHCLKX: McASP transmit high-frequency master clock.
  - ACLKX: McASP transmit bit clock.
- Transmit Frame Sync Generator;
  - AFSX: McASP transmit frame sync or left/right clock (LRCLK).
- Receive clock generator;
  - AHCLKR: McASP receive high-frequency master clock.
  - ACLKR: McASP receive bit clock.
- Receive Frame Sync Generator;
  - AFSR: McASP receive frame sync or left/right clock (LRCLK).
- Mute in/out;
  - AMUTEIN: McASP mute input (from external device).
  - AMUTE: McASP mute output.
  - Data pins AXRn.

24.3.2 Functional Block Diagram

Figure 24-2 shows the major blocks of the McASP. The McASP has independent receive/transmit clock generators and frame sync generators.

Figure 24-2. McASP Block Diagram



A McASP has 6 serial data pins.

B AMUTEIN is not a dedicated McASP pin, but typically comes from one of the external interrupt pins.

24.3.2.1 System Level Connections

Figure 24-3 through Figure 24-7 show examples of McASP usage in digital audio encoder/decoder systems.

Figure 24-3. McASP to Parallel 2-Channel DACs

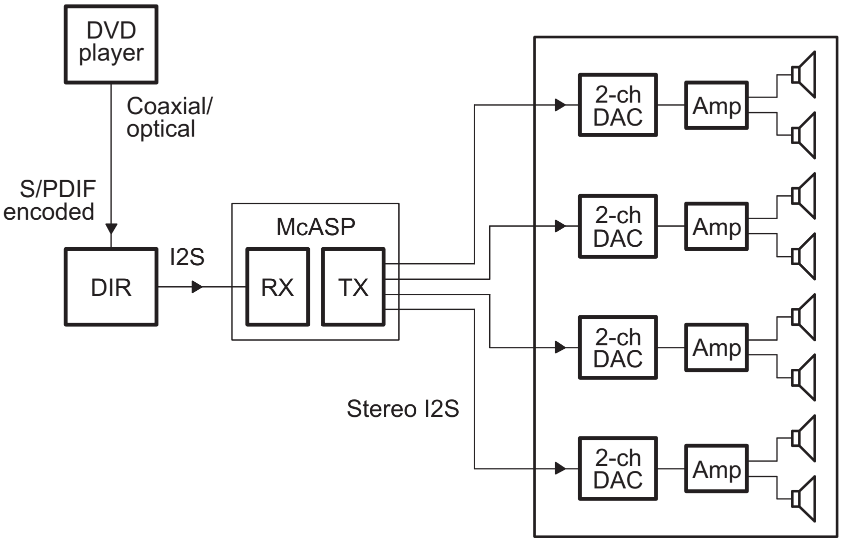


Figure 24-4. McASP to 6-Channel DAC and 2-Channel DAC

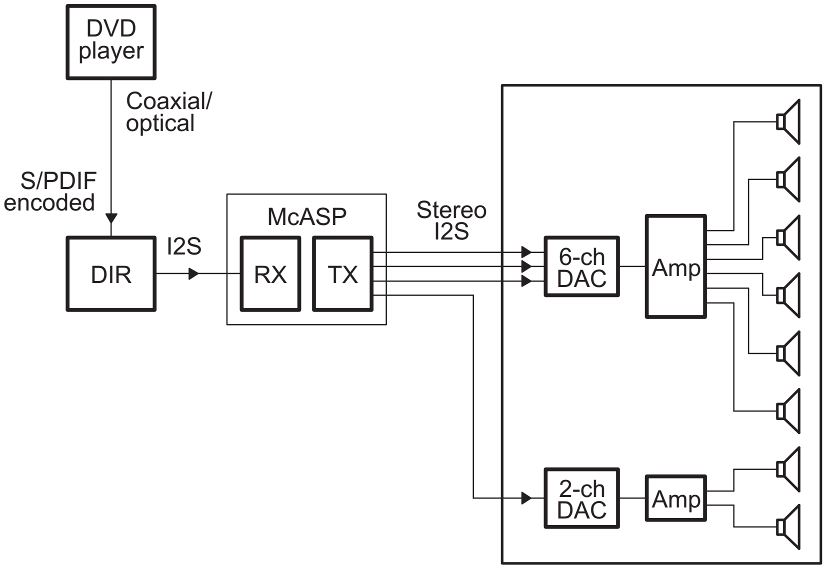


Figure 24-5. McASP to Digital Amplifier

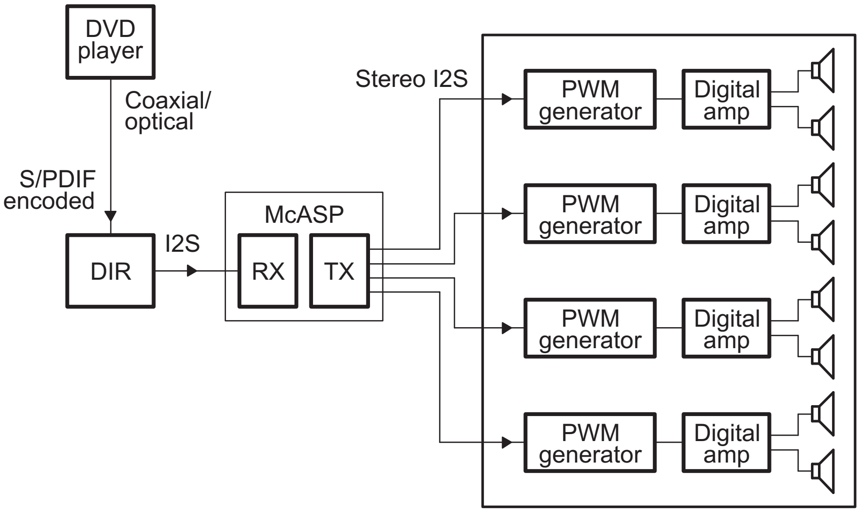


Figure 24-6. McASP as Digital Audio Encoder

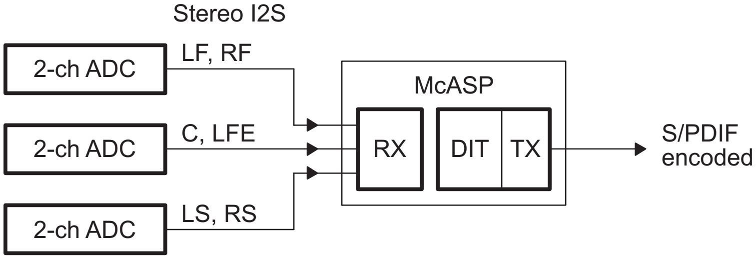
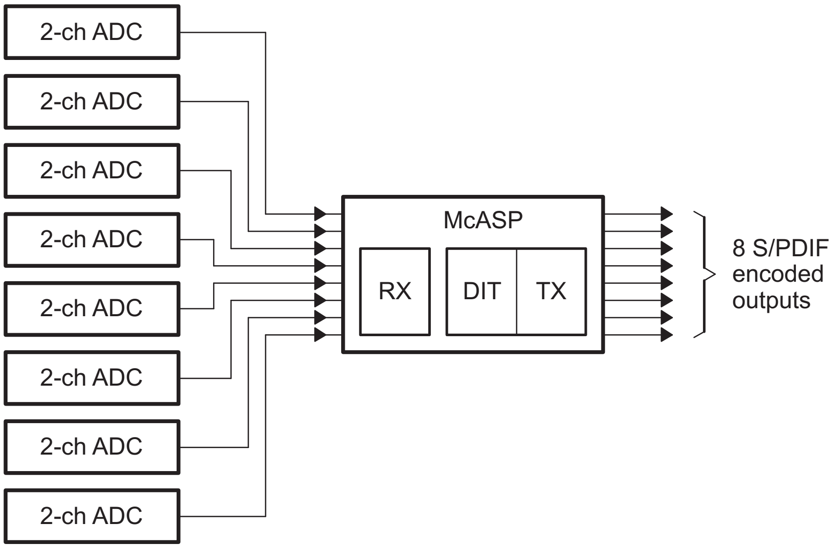


Figure 24-7. McASP as 16 Channel Digital Processor



24.3.3 Industry Standard Compliance Statement

The McASP supports the following industry standard interfaces.

24.3.3.1 TDM Format

The McASP transmitter and receiver support the multichannel, synchronous time-division-multiplexed (TDM) format via the TDM transfer mode. Within this transfer mode, a wide variety of serial data formats are supported, including formats compatible with devices using the Inter-Integrated Sound (I2S) protocol. This section briefly discusses the TDM format and the I2S protocol.

24.3.3.1.1 TDM Format

The TDM format is typically used when communicating between integrated circuit devices on the same printed circuit board or on another printed circuit board within the same piece of equipment. For example, the TDM format is used to transfer data between the processor and one or more analog-to-digital converter (ADC), digital-to-analog converter (DAC), or S/PDIF receiver (DIR) devices.

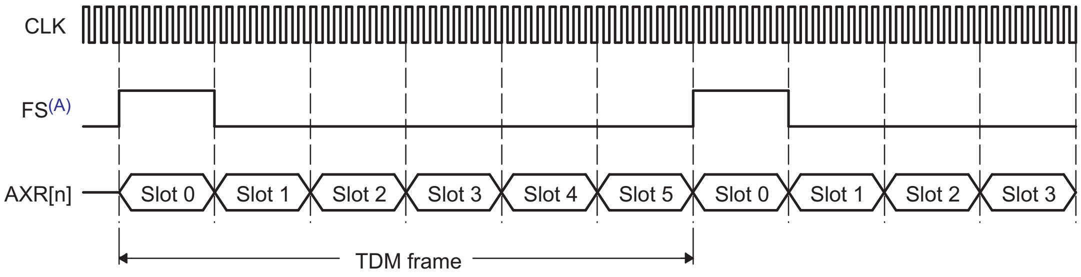
The TDM format consists of three components in a basic synchronous serial transfer: the clock, the data, and the frame sync. In a TDM transfer, all data bits (AXRn) are synchronous to the serial clock (ACLKX or ACLKR). The data bits are grouped into words and slots (as defined in Section 24.3.4). The "slots" are also commonly referred to as "time slots" or "channels" in TDM terminology. A frame consists of multiple slots (or channels). Each TDM frame is defined by the frame sync signal (AFSX or AFSR). Data transfer is continuous and periodic, since the TDM format is most commonly used to communicate with data converters that operate at a fixed sample rate.

There are no delays between slots. The last bit of slot N is followed immediately on the next serial clock cycle with the first bit of slot N + 1, and the last bit of the last slot is followed immediately on the next serial clock with the first bit of the first slot. However, the frame sync may be offset from the first bit of the first slot with a 0, 1, or 2-cycle delay.

It is required that the transmitter and receiver in the system agree on the number of bits per slot, since the determination of a slot boundary is not made by the frame sync signal (although the frame sync marks the beginning of slot 0 and the beginning of a new frame).

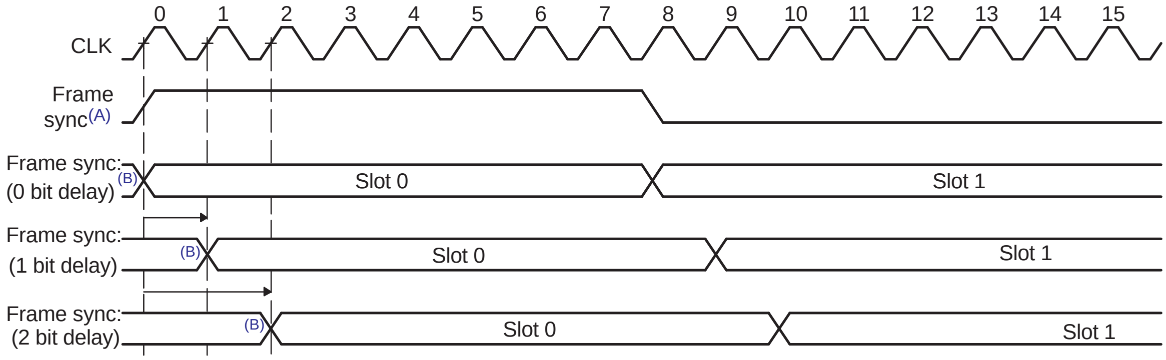
Figure 24-8 shows the TDM format. Figure 24-9 shows the different bit delays from the frame sync.

Figure 24-8. TDM Format–6 Channel TDM Example



A FS duration of slot is shown. FS duration of single bit is also supported.

Figure 24-9. TDM Format Bit Delays from Frame Sync



A FS duration of slot is shown. FS duration of single bit is also supported.

B Last bit of last slot of previous frame. No gap between this bit and the first bit of slot 0 is allowed.

In a typical audio system, one frame of data is transferred during each data converter sample period f_s. To support multiple channels, the choices are to either include more time slots per frame (thus operating with a higher bit clock rate), or to use additional data pins to transfer the same number of channels (thus operating with a slower bit clock rate).

For example, a particular six channel DAC may be designed to transfer over a single serial data pin AXRn as shown in Figure 24-8. In this case the serial clock must run fast enough to transfer a total of 6 channels within each frame period. Alternatively, a similar six channel DAC may be designed to use three serial data pins AXR[0,1,2], transferring two channels of data on each pin during each sample period. In the latter case, if the sample period remains the same, the serial clock can run three times slower than the former case. The McASP is flexible enough to support either type of DAC.

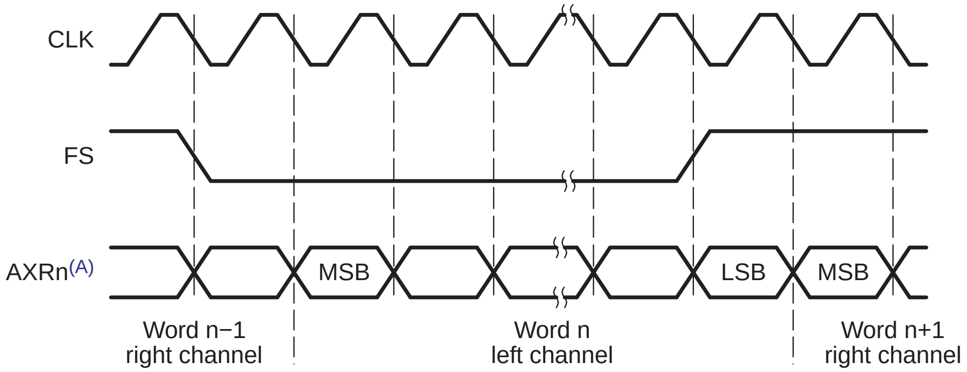
24.3.3.1.2 Inter-Integrated Sound (I2S) Format

The inter-integrated sound (I2S) format is used extensively in audio interfaces. The TDM transfer mode of the McASP supports the I2S format when configured to 2 slots per frame.

I2S format is specifically designed to transfer a stereo channel (left and right) over a single data pin AXRn. "Slots" are also commonly referred to as "channels". The frame width duration in the I2S format is the same as the slot size. The frame signal is also referred to as "word select" in the I2S format. Figure 24-10 shows the I2S protocol.

The McASP supports transfer of multiple stereo channels over multiple AXRn pins.

Figure 24-10. Inter-Integrated Sound (I2S) Format



A 1 to 6 data pins may be supported.

24.3.3.2 S/PDIF Coding Format

The McASP transmitter supports the S/PDIF format with 3.3V biphase-mark encoded output. The S/PDIF format is supported by the digital audio interface transmit (DIT) transfer mode of the McASP. This section briefly discusses the S/PDIF coding format.

24.3.3.2.1 Biphase-Mark Code (BMC)

In S/PDIF format, the digital signal is coded using the biphase-mark code (BMC). The clock, frame, and data are embedded in only one signal—the data pin AXRn. In the BMC system, each data bit is encoded into two logical states (00, 01, 10, or 11) at the pin. These two logical states form a cell. The duration of the cell, which equals to the duration of the data bit, is called a time interval. A logical 1 is represented by two transitions of the signal within a time interval, which corresponds to a cell with logical states 01 or 10. A logical 0 is represented by one transition within a time interval, which corresponds to a cell with logical states 00 or 11. In addition, the logical level at the start of a cell is inverted from the level at the end of the previous cell. Figure 24-11 and Table 24-5 show how data is encoded to the BMC format.

As shown in Figure 24-11, the frequency of the clock is twice the unencoded data bit rate. In addition, the clock is always programmed to 128 × f_s, where f_s is the sample rate (see Section 24.3.3.2.3 for details on how this clock rate is derived based on the S/PDIF format). The device receiving in S/PDIF format can recover the clock and frame information from the BMC signal.

Figure 24-11. Biphase-Mark Code (BMC)

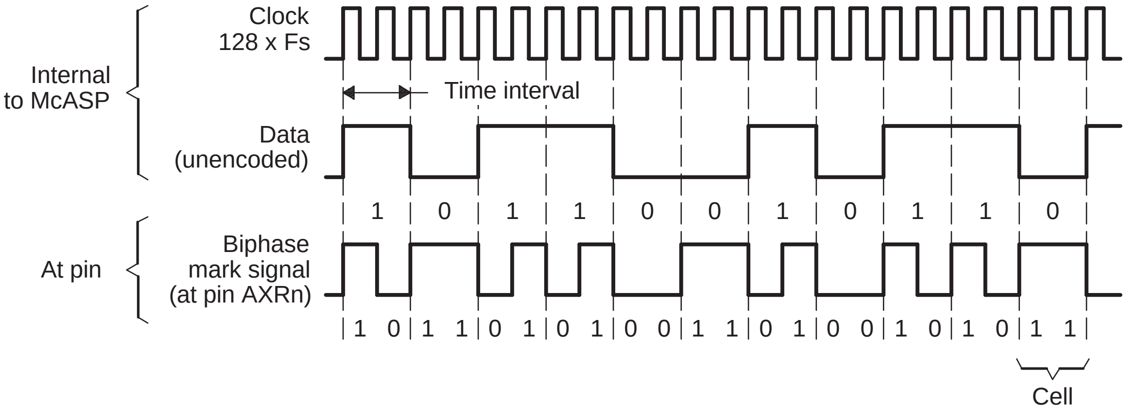


Table 24-5. Biphase-Mark Encoder

Data (Unencoded)	Previous State at Pin AXRn	BMC-Encoded Cell Output at AXRn
0	0	11
0	1	00
1	0	10
1	1	01

24.3.3.2.2 Subframe Format

Every audio sample transmitted in a subframe consists of 32 S/PDIF time intervals (or cells), numbered from 0 to 31. Figure 24-12 shows a subframe.

- **Time intervals 0-3** carry one of the three permitted preambles to signify the type of audio sample in the current subframe. The preamble is *not* encoded in BMC format, and therefore the preamble code can contain more than two consecutive 0 or 1 logical states in a row. See Table 24-6.
- **Time intervals 4-27** carry the audio sample word in linear 2s-complement representation. The most-significant bit (MSB) is carried by time interval 27. When a 24-bit coding range is used, the least-significant bit (LSB) is in time interval 4. When a 20-bit coding range is used, time intervals 8-27 carry the audio sample word with the LSB in time interval 8. Time intervals 4-7 may be used for other applications and are designated auxiliary sample bits.
- If the source provides fewer bits than the interface allows (either 20 or 24), the unused LSBs are set to logical 0. For a nonlinear PCM audio application or a data application, the main data field may carry any other information.
- **Time interval 28** carries the validity bit (V) associated with the main data field in the subframe.
- **Time interval 29** carries the user data channel (U) associated with the main data field in the subframe.
- **Time interval 30** carries the channel status information (C) associated with the main data field in the subframe. The channel status indicates if the data in the subframe is digital audio or some other type of data.
- **Time interval 31** carries a parity bit (P) such that time intervals 4-31 carry an even number of 1s and an even number of 0s (even parity). As shown in Table 24-6, the preambles (time intervals 0-3) are also defined with even parity.

Figure 24-12. S/PDIF Subframe Format

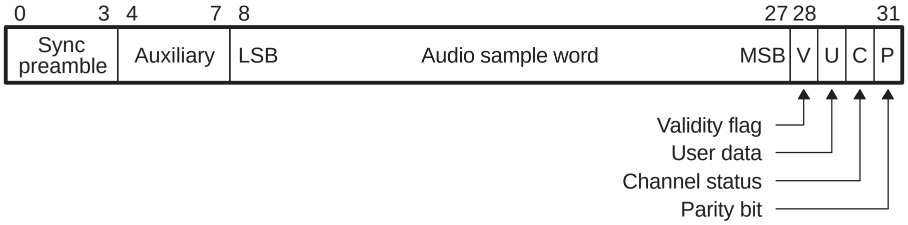


Table 24-6. Preamble Codes

Preamble Code[1]	Previous Logical State	Logical States on pin AXRn[2]	Description
B (or Z)	0	1110 1000	Start of a block and subframe 1
M (or X)	0	1110 0010	Subframe 1
W (or Y)	0	1110 0100	Subframe 2

[1] Historically, preamble codes are referred to as B, M, W. For use in professional applications, preambles are referred to as Z, X, Y, respectively.

[2] The preamble is not BMC encoded. Each logical state is synchronized to the serial clock. These 8 logical states make up time slots (cells) 0 to 3 in the S/PDIF stream.

As shown in Table 24-6, the McASP DIT only generates one polarity of preambles and it assumes the previous logical state to be 0. This is because the McASP assures an even-polarity encoding scheme when transmitting in DIT mode. If an underrun condition occurs, the DIT resynchronizes to the correct logic level on the AXRn pin before continuing with the next transmission.

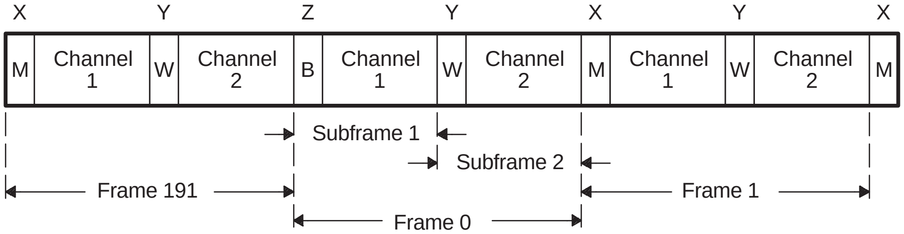
24.3.3.2.3 Frame Format

An S/PDIF frame is composed of two subframes (Figure 24-13). For linear coded audio applications, the rate of frame transmission normally corresponds exactly to the source sampling frequency f_s. The S/PDIF format clock rate is therefore 128 × f_s (128 = 32 cells/subframe × 2 clocks/cell × 2 subframes/sample). For example, for an S/PDIF stream at a 192 kHz sampling frequency, the serial clock is 128 × 192 kHz = 24.58 MHz.

In 2-channel operation mode, the samples taken from both channels are transmitted by time multiplexing in consecutive subframes. Both subframes contain valid data. The first subframe (**left** or **A** channel in stereophonic operation and **primary** channel in monophonic operation) normally starts with preamble M. However, the preamble of the first subframe changes to preamble B once every 192 frames to identify the start of the block structure used to organize the channel status information. The second subframe (**right** or **B** channel in stereophonic operation and **secondary** channel in monophonic operation) always starts with preamble W.

In single-channel operation mode in a professional application, the frame format is the same as in the 2-channel mode. Data is carried in the first subframe and may be duplicated in the second subframe. If the second subframe is not carrying duplicate data, cell 28 (validity bit) is set to logical 1.

Figure 24-13. S/PDIF Frame Format



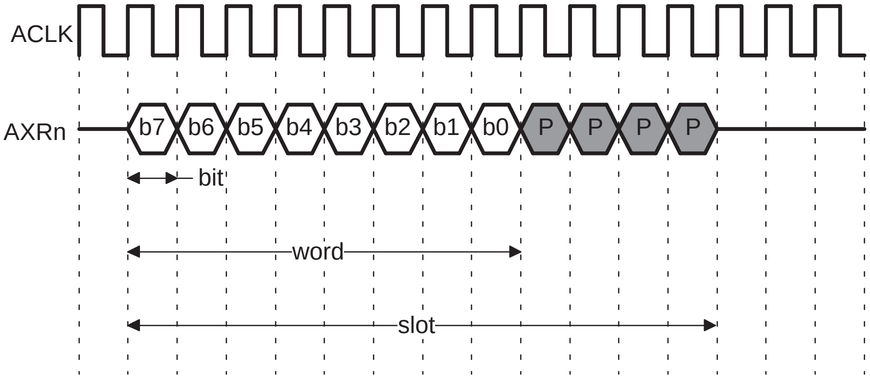
24.3.4 Definition of Terms

The serial bit stream transmitted or received by the McASP is a long sequence of 1s and 0s, either output or input on one of the audio transmit/receive pins (AXRn). However, the sequence has a hierarchical organization that can be described in terms of frames of data, slots, words, and bits.

A basic synchronous serial interface consists of three important components: clock, frame sync, and data. Figure 24-14 shows two of the three basic components—the clock (ACLK) and the data (AXRn). Figure 24-14 does not specify whether the clock is for transmit (ACLKX) or receive (ACLKR) because the definitions of terms apply to both receive and transmit interfaces. In operation, the transmitter uses ACLKX as the serial clock, and the receiver uses ACLKR as the serial clock. Optionally, the receiver can use ACLKX as the serial clock when the transmitter and receiver of the McASP are configured to operate synchronously.

Bit	A bit is the smallest entity in the serial data stream. The beginning and end of each bit is marked by an edge of the serial clock. The duration of a bit is a serial clock period. A 1 is represented by a logic high on the AXRn pin for the entire duration of the bit. A 0 is represented by a logic low on the AXRn pin for the entire duration of the bit.
Word	A word is a group of bits that make up the data being transferred between the processor and the external device. Figure 24-14 shows an 8-bit word.
Slot	A slot consists of the bits that make up the word, and may consist of additional bits used to pad the word to a convenient number of bits for the interface between the processor and the external device. In Figure 24-14, the audio data consists of only 8 bits of useful data (8-bit word), but it is padded with 4 zeros (12-bit slot) to satisfy the desired protocol in interfacing to an external device. Within a slot, the bits may be shifted in/out of the McASP on the AXRn pin either MSB or LSB first. When the word size is smaller than the slot size, the word may be aligned to the left (beginning) of the slot or to the right (end) of the slot. The additional bits in the slot not belonging to the word may be padded with 0, 1, or with one of the bits (the MSB or the LSB typically) from the data word. These options are shown in Figure 24-15.

Figure 24-14. Definition of Bit, Word, and Slot

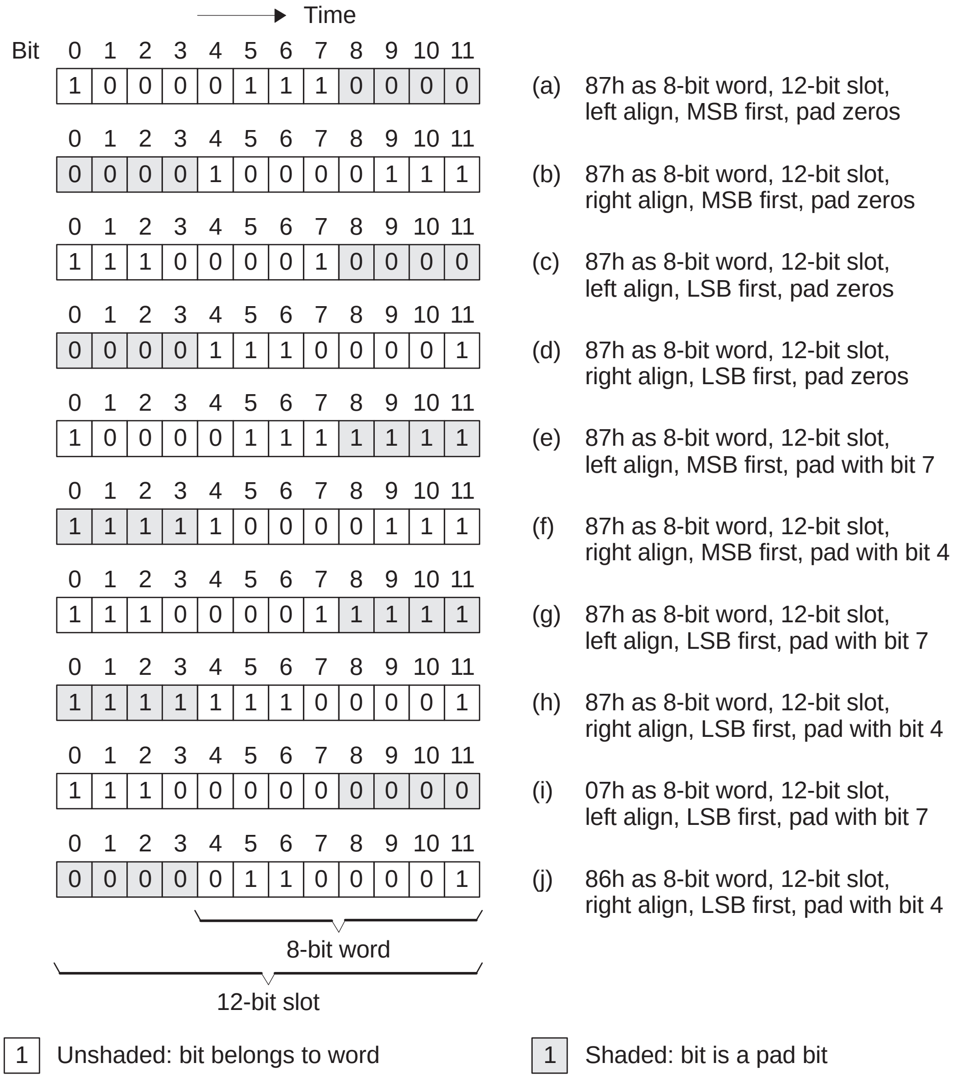


(1) b7:b0 - bits. Bits b7 to b0 form a word.

(2) P - pad bits. Bits b7 to b0, together with the four pad bits, form a slot.

(3) In this example, the data is transmitted MSB first, left aligned.

Figure 24-15. Bit Order and Word Alignment Within a Slot Examples



Bit	0	1	2	3	4	5	6	7	8	9	10	11	
	1	0	0	0	0	1	1	1	0	0	0	0	(a) 87h as 8-bit word, 12-bit slot, left align, MSB first, pad zeros
	0	0	0	0	1	0	0	0	0	1	1	1	(b) 87h as 8-bit word, 12-bit slot, right align, MSB first, pad zeros
	1	1	1	0	0	0	0	1	0	0	0	0	(c) 87h as 8-bit word, 12-bit slot, left align, LSB first, pad zeros
	0	0	0	0	1	1	1	0	0	0	0	1	(d) 87h as 8-bit word, 12-bit slot, right align, LSB first, pad zeros
	1	0	0	0	0	1	1	1	1	1	1	1	(e) 87h as 8-bit word, 12-bit slot, left align, MSB first, pad with bit 7
	1	1	1	1	1	0	0	0	0	1	1	1	(f) 87h as 8-bit word, 12-bit slot, right align, MSB first, pad with bit 4
	1	1	1	0	0	0	0	1	1	1	1	1	(g) 87h as 8-bit word, 12-bit slot, left align, LSB first, pad with bit 7
	1	1	1	1	1	1	1	0	0	0	0	1	(h) 87h as 8-bit word, 12-bit slot, right align, LSB first, pad with bit 4
	1	1	1	0	0	0	0	0	0	0	0	0	(i) 07h as 8-bit word, 12-bit slot, left align, LSB first, pad with bit 7
	0	0	0	0	0	1	1	0	0	0	0	1	(j) 86h as 8-bit word, 12-bit slot, right align, LSB first, pad with bit 4

1 Unshaded: bit belongs to word

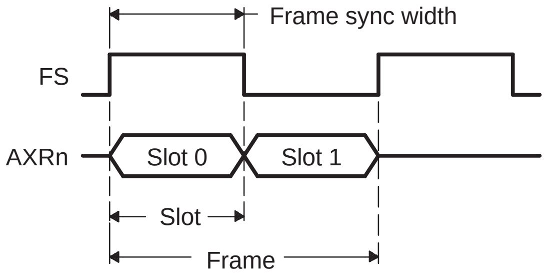
1 Shaded: bit is a pad bit

The third basic element of a synchronous serial interface is the frame synchronization signal, also referred to as frame sync in this document.

Frame	A frame contains one or multiple slots, as determined by the desired protocol. Figure 24-16 shows an example frame of data and the frame definitions. Figure 24-16 does not specify whether the frame sync (FS) is for transmit (AFSX) or receive (AFSR) because the definitions of terms apply to both receive and transmit interfaces. In operation, the transmitter uses AFSX and the receiver uses AFSR. Optionally, the receiver can use AFSX as the frame sync when the transmitter and receiver of the McASP are configured to operate synchronously.

This section only shows the generic definition of the frame sync. See Section 24.3.3 and Section 24.3.8.1 for details on the frame sync formats required for the different transfer modes and protocols (burst mode, TDM mode and I2S format, DIT mode and S/PDIF format).

Figure 24-16. Definition of Frame and Frame Sync Width



(1) In this example, there are two slots in a frame, and FS duration of slot length is shown.

Other terms used throughout the document:

TDM	Time-division multiplexed. See Section 24.3.3.1 for details on the TDM protocol.
DIR	Digital audio interface receive. The McASP does not natively support receiving in the S/PDIF format. The McASP supports I2S format output by an external DIR device.
DIT	Digital audio interface transmit. The McASP supports transmitting in S/PDIF format on up to all data pins configured as outputs.
I2S	Inter-Integrated Sound protocol, commonly used on audio interfaces. The McASP supports the I2S protocol as part of the TDM mode (when configured as a 2-slot frame).
Slot or Time Slot	For TDM format, the term time slot is interchangeable with the term slot defined in this section. For DIT format, a McASP time slot corresponds to a DIT subframe.

24.3.5 Clock and Frame Sync Generators

The McASP clock generators are able to produce two independent clock zones: transmit and receive clock zones. The serial clock generators may be programmed independently for the transmit section and the receive section, and may be completely asynchronous to each other. The serial clock (clock at the bit rate) may be sourced:

- Internally - by passing through two clock dividers off the internal clock source (AUXCLK).
- Externally - directly from ACLKR/X pin.
- Mixed - an external high-frequency clock is input to the McASP on either the AHCLKX or AHCLKR pins, and divided down to produce the bit rate clock.

In the internal/mixed cases, the bit rate clock is generated internally and should be driven out on the ACLKX (for transmit) or ACLKR (for receive) pins. In the internal case, an internally-generated high-frequency clock may be driven out onto the AHCLKX or AHCLKR pins to serve as a reference clock for other components in the system.

The McASP requires a minimum of a bit clock and a frame sync to operate, and provides the capability to reference these clocks from an external high-frequency master clock. In DIT mode, it is possible to use only internally-generated clocks and frame syncs.

24.3.5.1 Transmit Clock

The transmit bit clock, ACLKX, (Figure 24-17) may be either externally sourced from the ACLKX pin or internally generated, as selected by the CLKXM bit. If internally generated (CLKXM = 1), the clock is divided down by a programmable bit clock divider (CLKXDIV) from the transmit high-frequency master clock (AHCLKX).

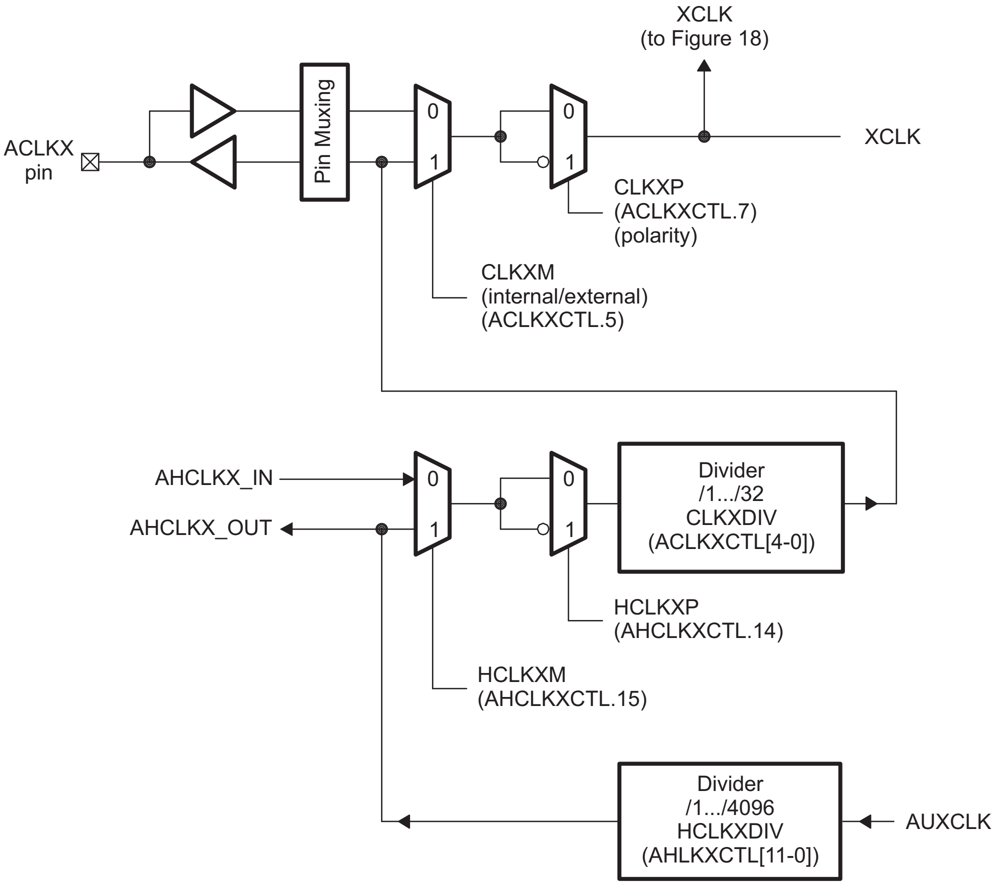
Internally, the McASP always shifts transmit data at the rising edge of the internal transmit clock, XCLK, (Figure 24-17). The CLKXP mux determines if ACLKX needs to be inverted to become XCLK. If CLKXP = 0, the CLKXP mux directly passes ACLKX to XCLK. As a result, the McASP shifts transmit data at the rising edge of ACLKX. If CLKXP = 1, the CLKX mux passes the inverted version of ACLKX to XCLK. As a result, the McASP shifts transmit data at the falling edge of ACLKX.

The transmit high-frequency master clock, AHCLKX, may be either externally sourced from the AHCLKX pin or internally generated, as selected by the HCLKXM bit. If internally generated (HCLKXM = 1), the clock is divided down by a programmable high clock divider (HCLKXDIV) from McASP internal clock source AUXCLK. The transmit high-frequency master clock may be (but is not required to be) output on the AHCLKX pin where it is available to other devices in the system.

The transmit clock configuration is controlled by the following registers:

- ACLKXCTL.
- AHCLKXCTL.

Figure 24-17. Transmit Clock Generator Block Diagram



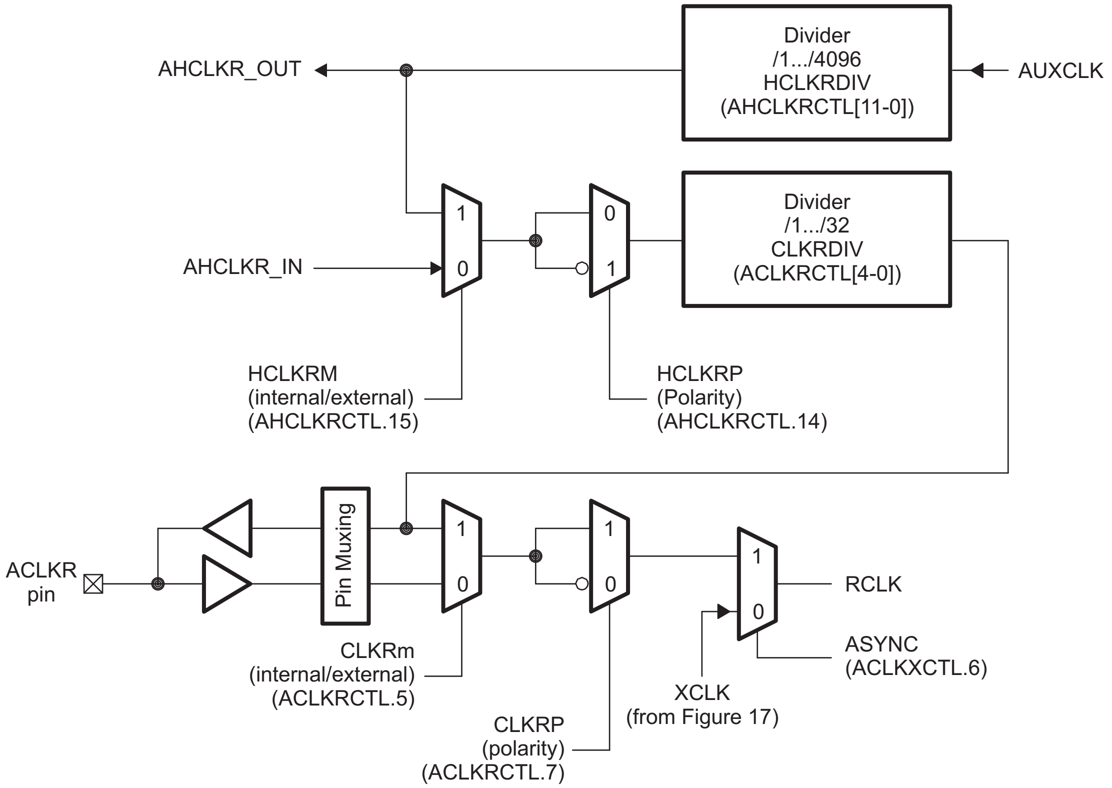
24.3.5.2 Receive Clock

The receiver also has the option to operate synchronously from the ACLKX and AFSX signals. This is achieved when the ASYNC bit in the transmit clock control register (ACLKXCTL) is cleared to 0 (see Figure 24-18). The receiver may be configured with different polarity (CLKRP) and frame sync data delay options from those options of the transmitter.

The receive clock configuration is controlled by the following registers:

- ACLKRCTL.
- AHCLKRCTL.

Figure 24-18. Receive Clock Generator Block Diagram



24.3.5.3 Frame Sync Generator

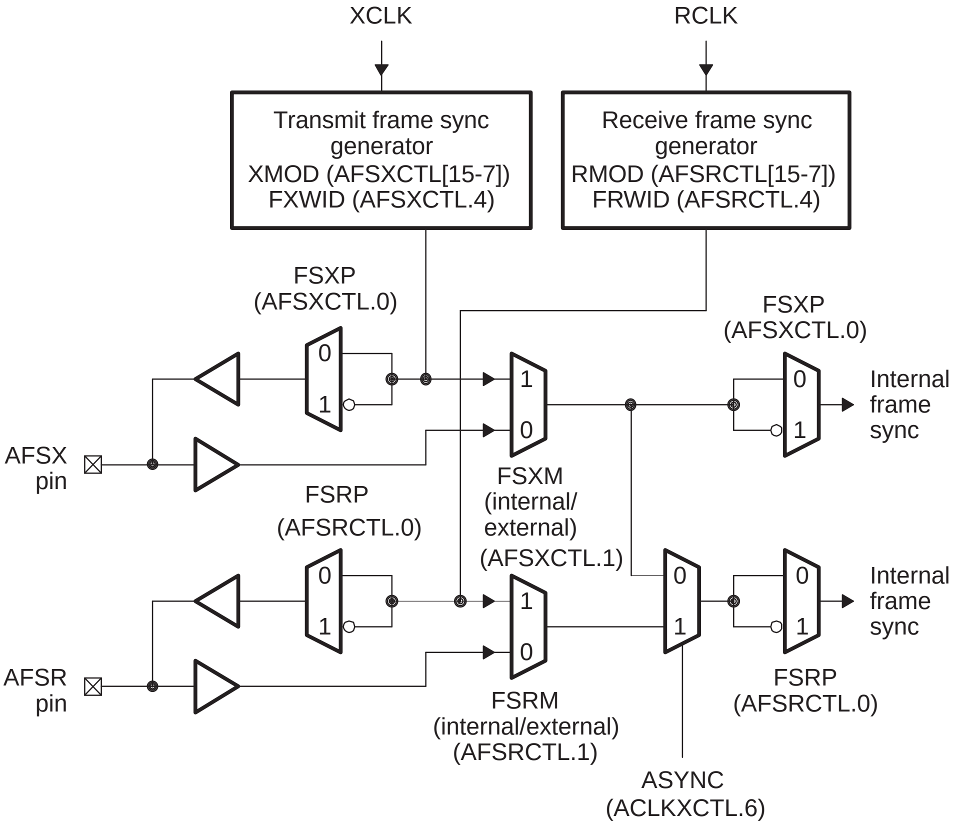
There are two different modes for frame sync: burst and TDM. A block diagram of the frame sync generator is shown in Figure 24-19. The frame sync options are programmed by the receive and transmit frame sync control registers (AFSRCTL and AFSXCTL). The options are:

- Internally-generated or externally-generated.
- Frame sync polarity: rising edge or falling edge.
- Frame sync width: single bit or single word.
- Bit delay: 0, 1, or 2 cycles before the first data bit.

The transmit frame sync pin is AFSX and the receive frame sync pin is AFSR. A typical usage for these pins is to carry the left/right clock (LRCLK) signal when transmitting and receiving stereo data.

Regardless if the AFSX/AFSR is internally generated or externally sourced, the polarity of AFSX/AFSR is determined by FSXP/FSRP, respectively, to be either rising or falling edge. If FSXP/FSRP = 0, the frame sync polarity is rising edge. If FSRP/FSRP = 1, the frame sync polarity is falling edge.

Figure 24-19. Frame Sync Generator Block Diagram



24.3.5.4 Clocking Examples

Some examples of processes using the McASP clocking and frame flexibility are:

- Receive data from a DVD at 48 kHz, but output up-sampled or decoded audio at 96 kHz or 192 kHz. This could be accomplished by inputting a high-frequency master clock (for example, 512 × receive FS), receiving with an internally-generated bit clock ratio of divide-by-8, and transmitting with an internally-generated bit clock ratio of divide-by-4 or divide-by-2.
- Transmit/receive data based on one sample rate (for example, 44.1 kHz), and transmit/receive data at a different sample rate (for example, 48 kHz).

24.3.5.5 Crystal Considerations

When choosing a high frequency input crystal for the device, consider the maximum functional clock frequency of the McASP module versus the maximum external receive/transmit clock AHCLKR/AHCLKX, especially when planning to generate the clocks internally. For example, if the maximum transmit clock AHCLKX is 25 MHz, a master oscillator CLK_M_OSC of 26 MHz requires a divide-by-2. However, master oscillator of 25 MHz enables full performance of the transmit clock.

24.3.6 Signal Descriptions

The signals used on the McASP audio interface are listed in Table 24-7.

Table 24-7. McASP Interface Signals

Pin	I/O/Z	Device Reset ($\overline{\text{RESET}}$ = 0)	Description
			Transmitter Control
AHCLKX	I/O/Z	Input	Transmit high-frequency master clock
AFSX	I/O/Z	Input	Transmit frame sync or left/right clock (LRCLK)
ACLKX	I/O/Z	Input	Transmit bit clock
			Receiver Control
AHCLKR	I/O/Z	Input	Receive high-frequency master clock
AFSR	I/O/Z	Input	Receive frame sync or left/right clock (LRCLK)
ACLKR	I/O/Z	Input	Receive bit clock
			Mute
AMUTE	I/O/Z	Input	Mute output
AMUTEIN	I/O/Z	Input	Mute input
			Data
AXRn	I/O/Z	Input	TX/RX data pins

24.3.7 Pin Multiplexing

The McASP signals share pins with other processor functions. For detailed information on the McASP pin multiplexing and configuration, see the pin multiplexing information in the device-specific data manual.

24.3.8 Transfer Modes

24.3.8.1 Burst Transfer Mode

The McASP supports a burst transfer mode, which is useful for nonaudio data such as passing control information between two processors. Burst transfer mode uses a synchronous serial format similar to the TDM mode. The frame sync generation is not periodic or time-driven as in TDM mode, but data driven, and the frame sync is generated for each data word transferred.

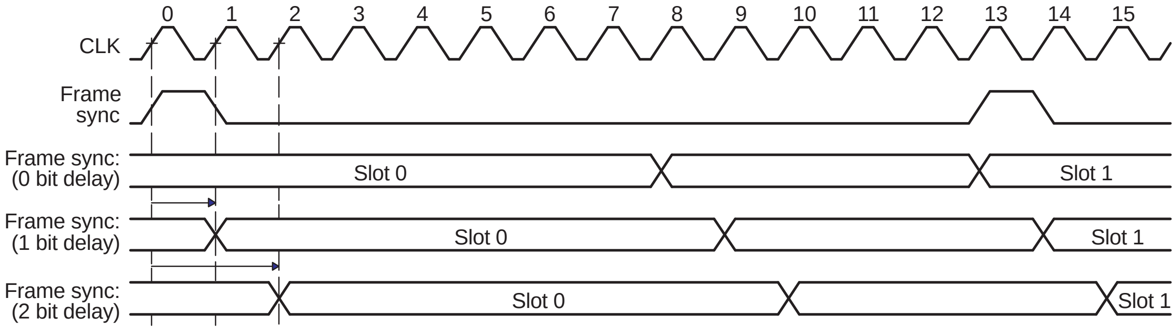
When operating in burst frame sync mode (Figure 24-20), as specified for transmit (XMOD = 0 in AFSXCTL) and receive (RMOD = 0 in AFSRCTL), one slot is shifted for each active edge of the frame sync signal that is recognized. Additional clocks after the slot and before the next frame sync edge are ignored.

In burst frame sync mode, the frame sync delay may be specified as 0, 1, or 2 serial clock cycles. This is the delay between the frame sync active edge and the start of the slot. The frame sync signal lasts for a single bit clock duration (FRWID = 0 in AFSRCTL, FXWID = 0 in AFSXCTL).

For transmit, when generating the transmit frame sync internally, the frame sync begins when the previous transmission has completed and when all the XBUF[n] (for every serializer set to operate as a transmitter) has been updated with new data.

For receive, when generating the receive frame sync internally, frame sync begins when the previous transmission has completed and when all the RBUF[n] (for every serializer set to operate as a receiver) has been read.

Figure 24-20. Burst Frame Sync Mode



The control registers must be configured as follows for the burst transfer mode. The burst mode specific bit fields are in bold face:

- PFUNC: The clock, frame, data pins must be configured for McASP function.
- PDIR: The clock, frame, data pins must be configured to the direction desired.
- PDOUT, PDIN, PDSET, PDCLR: Not applicable. Leave at default.
- GBLCTL: Follow the initialization sequence in Section 24.3.12.2 to configure this register.
- AMUTE: Not applicable. Leave at default.
- DLBCTL: If loopback mode is desired, configure this register according to Section 24.3.10.5, otherwise leave this register at default.
- DITCTL: DITEN must be left at default 0 to select non-DIT mode. Leave the register at default.
- RMASK/XMASK: Mask desired bits according to Section 24.3.9.2 and Section 24.3.10.3.
- RFMT/XFMT: Program all fields according to data format desired. See Section 24.3.10.3.
- AFSRCTL/AFSXCTL: Clear **RMOD/XMOD** bits to 0 to indicate burst mode. Clear **FRWID/FXWID** bits to 0 for single bit frame sync duration. Configure other fields as desired.
- ACLKRCTL/ACLKXCTL: Program all fields according to bit clock desired. See Section 24.3.5.
- AHCLKRCTL/AHCLKXCTL: Program all fields according to high-frequency clock desired. See Section 24.3.5.

- RTDM/XTDM: Program RTDMS0/XTDMS0 to 1 to indicate one active slot only. Leave other fields at default.
- RINTCTL/XINTCTL: Program all fields according to interrupts desired.
- RCLKCHK/XCLKCHK: Not applicable. Leave at default.
- SRCTLn: Program SRMOD to inactive/transmitter/receiver as desired. DISMOD is not applicable and should be left at default.
- DITCSRA[n], DITCSRB[n], DITUDRA[n], DITUDRB[n]: Not applicable. Leave at default.

24.3.8.2 Time-Division Multiplexed (TDM) Transfer Mode

The McASP time-division multiplexed (TDM) transfer mode supports the TDM format discussed in Section 24.3.3.1.

Transmitting data in the TDM transfer mode requires a minimum set of pins:

- ACLKX - transmit bit clock.
- AFSX - transmit frame sync (or commonly called left/right clock).
- One or more serial data pins, AXRn, whose serializers have been configured to transmit.

The transmitter has the option to receive the ACLKX bit clock as an input, or to generate the ACLKX bit clock by dividing down the AHCLKX high-frequency master clock. The transmitter can either generate AHCLKX internally or receive AHCLKX as an input. See Section 24.3.5.1.

Similarly, to receive data in the TDM transfer mode requires a minimum set of pins:

- ACLKR - receive bit clock.
- AFSR - receive frame sync (or commonly called left/right clock).
- One or more serial data pins, AXRn, whose serializers have been configured to receive.

The receiver has the option to receive the ACLKR bit clock as an input or to generate the ACLKR bit clock by dividing down the AHCLKR high-frequency master clock. The receiver can either generate AHCLKR internally or receive AHCLKR as an input. See Section 24.3.5.2 and Section 24.3.5.3.

The control registers must be configured as follows for the TDM mode. The TDM mode specific bit fields are in bold face:

- PFUNC: The clock, frame, data pins must be configured for McASP function.
- PDIR: The clock, frame, data pins must be configured to the direction desired.
- PDOUT, PDIN, PDSET, PDCLR: Not applicable. Leave at default.
- GBLCTL: Follow the initialization sequence in Section 24.3.12.2 to configure this register.
- AMUTE: Program all fields according to mute control desired.
- DLBCTL: If loopback mode is desired, configure this register according to Section 24.3.10.5, otherwise leave this register at default.
- DITCTL: DITEN must be left at default 0 to select TDM mode. Leave the register at default.
- RMASK/XMASK: Mask desired bits according to Section 24.3.9.2 and Section 24.3.10.3.
- RFMT/XFMT: Program all fields according to data format desired. See Section 24.3.10.3.
- AFSRCTL/AFSXCTL: Set **RMOD/XMOD** bits to 2-32 for TDM mode. Configure other fields as desired.
- ACLKRCTL/ACLKXCTL: Program all fields according to bit clock desired. See Section 24.3.5.
- AHCLKRCTL/AHCLKXCTL: Program all fields according to high-frequency clock desired. See Section 24.3.5.
- RTDM/XTDM: Program all fields according to the time slot characteristics desired.
- RINTCTL/XINTCTL: Program all fields according to interrupts desired.
- RCLKCHK/XCLKCHK: Program all fields according to clock checking desired.
- SRCTLn: Program all fields according to serializer operation desired.
- DITCSRA[n], DITCSRB[n], DITUDRA[n], DITUDRB[n]: Not applicable. Leave at default.

24.3.8.2.1 TDM Time Slots

TDM mode on the McASP can extend to support multiprocessor applications, with up to 32 time slots per frame. For each of the time slots, the McASP may be configured to participate or to be inactive by configuring XTDM and/or RTDM (this allows multiple processors to communicate on the same TDM serial bus).

The TDM sequencer (separate ones for transmit and receive) functions in this mode. The TDM sequencer counts the slots beginning with the frame sync. For each slot, the TDM sequencer checks the respective bit in either XTDM or RTDM to determine if the McASP should transmit/receive in that time slot.

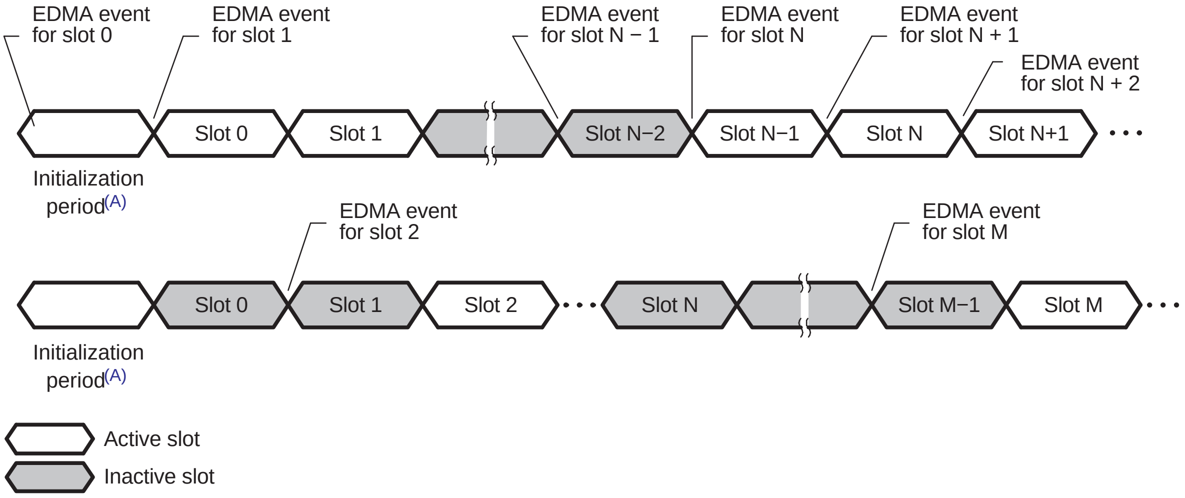
If the transmit/receive bit is active, the McASP functions normally during that time slot; otherwise, the McASP is inactive during that time slot; no update to the buffer occurs, and no event is generated. Transmit pins are automatically set to a high-impedance state, 0, or 1 during that slot, as determined by bit DISMOD in SRCTL[n].

Figure 24-21 shows when the transmit DMA event AXEVT is generated. See Section 24.3.10.1.1 for details on data ready and the initialization period indication. The transmit DMA event for an active time slot (slot N) is generated during the previous time slot (slot N - 1), regardless if the previous time slot (slot N - 1) is active or inactive.

During an active transmit time slot (slot N), if the next time slot (slot N + 1) is configured to be active, the copy from XRBUF[n] to XRSR[n] generates the DMA event for time slot N + 1. If the next time slot (slot N + 1) is configured to be inactive, then the DMA event will be delayed to time slot M - 1. In this case, slot M is the next active time slot. The DMA event for time slot M is generated during the first bit time of slot M - 1.

The receive DMA request generation does not need this capability, since the receive DMA event is generated after data is received in the buffer (looks back in time). If a time slot is disabled, then no data is copied to the buffer for that time slot and no DMA event is generated.

Figure 24-21. Transmit DMA Event (AXEVT) Generation in TDM Time Slots



A See Section 24.3.12.2, step 7a.

24.3.8.2.2 Special 384 Slot TDM Mode for Connection to External DIR

The McASP receiver also supports a 384 time slot TDM mode (DIR mode), to support S/PDIF, AES-3, IEC-60958 receiver ICs whose natural block (block corresponds to McASP frame) size is 384 samples. The advantage to using the 384 time slot TDM mode is that interrupts may be generated synchronous to the S/PDIF, AES-3, IEC-60958, such as the last slot interrupt.

The receive TDM time slot register (RTDM) should be programmed to all 1s during reception of a DIR block. Other TDM functionalities (for example, inactive slots) are not supported (only the slot counter counts the 384 subframes in a block).

To receive data in the DIR mode, the following pins are typically needed:

- ACLKR - receive bit clock.
- AFSR - receive frame sync (or commonly called left/right clock). In this mode, AFSR should be connected to a DIR which outputs a start of block signal, instead of LRCLK.
- One or more serial data pins, AXRn, whose serializers have been configured to receive.

For this special DIR mode, the control registers can be configured just as for TDM mode, except set RMOD in AFSRCTL to 384 to receive 384 time slots.

24.3.8.3 Digital Audio Interface Transmit (DIT) Transfer Mode

In addition to the TDM and burst transfer modes, which are suitable for transmitting audio data between ICs inside the same system, the digital audio interface transmit (DIT) transfer mode of the McASP also supports transmission of audio data in the S/PDIF, AES-3, or IEC-60958 format. These formats are designed to carry audio data between different systems through an optical or coaxial cable. The DIT mode only applies to serializers configured as transmitters, not receivers. See Section 24.3.3.2 for a description of the S/PDIF format.

24.3.8.3.1 Transmit DIT Encoding

The McASP operation in DIT mode is basically identical to the 2 time slot TDM mode, but the data transmitted is output as a biphase mark encoded bit stream, with preamble, channel status, user data, validity, and parity automatically stuffed into the bit stream by the McASP. The McASP includes separate validity bits for even/odd subframes and two 384-bit RAM modules to hold channel status and user data bits.

The transmit TDM time slot register (XTDM) should be programmed to all 1s during DIT mode. TDM functionality is not supported in DIT mode, except that the TDM slot counter counts the DIT subframes.

To transmit data in the DIT mode, the following pins are typically needed:

- AHCLKX - transmit high-frequency master clock.
- One or more serial data pins, AXRn, whose serializers have been configured to transmit.

AHCLKX is optional (the internal clock source may be used instead), but if used as a reference, the processor provides a clock check circuit that continually monitors the AHCLKX input for stability.

If the McASP is configured to transmit in the DIT mode on more than one serial data pin, the bit streams on all pins will be synchronized. In addition, although they will carry unique audio data, they will carry the same channel status, user data, and validity information.

The actual 24-bit audio data must always be in bit positions 23-0 after passing through the first three stages of the transmit format unit.

For left-aligned Q31 data, the following transmit format unit settings process the data into right aligned 24-bit audio data ready for transmission:

- XROT = 010 (rotate right by 8 bits).
- XRVRS = 0 (no bit reversal, LSB first).
- XMASK = FFFF FF00h-FFFF 0000h (depending upon whether 24, 23, 22, 21, 20, 19, 18, 17, or 16 valid audio data bits are present).
- XPAD = 00 (pad extra bits with 0).

For right-aligned data, the following transmit format unit settings process the data into right aligned 24-bit audio data ready for transmission:

- XROT = 000 (rotate right by 0 bits).
- XRVRS = 0 (no bit reversal, LSB first).
- XMASK = 00FF FFFFh to 0000 FFFFh (depending upon whether 24, 23, 22, 21, 20, 19, 18, 17, or 16 valid audio data bits are present).
- XPAD = 00 (pad extra bits with 0).

24.3.8.3.2 Transmit DIT Clock and Frame Sync Generation

The DIT transmitter only works in the following configuration:

- In transmit frame control register (AFSXCTL):
  - Internally-generated transmit frame sync, FSXM = 1.
  - Rising-edge frame sync, FSXP = 0.
  - Bit-width frame sync, FXWID = 0.
  - 384-slot TDM, XMOD = 1 1000 0000b.
- In transmit clock control register (ACLKXCTL), ASYNC = 1.
- In transmit bitstream format register (XFMT), XSSZ = 1111 (32-bit slot size).

All combinations of AHCLKX and ACLKX are supported.

This is a summary of the register configurations required for DIT mode. The DIT mode specific bit fields are in bold face:

- PFUNC: The data pins must be configured for McASP function. If AHCLKX is used, it must also be configured for McASP function.
- PDIR: The data pins must be configured as outputs. If AHCLKX is used as an input reference, it should be configured as input. If internal clock source AUXCLK is used as the reference clock, it may be output on the AHCLKX pin by configuring AHCLKX as an output.
- PDOUT, PDIN, PDSET, PDCLR: Not applicable for DIT operation. Leave at default.
- GBLCTL: Follow the initialization sequence in Section 24.3.12.2 to configure this register.
- AMUTE: Program all fields according to mute control desired.
- DLBCTL: Not applicable. Loopback is not supported for DIT mode. Leave at default.
- DITCTL: **DITEN** bit must be set to 1 to enable DIT mode. Configure other bits as desired.
- RMASK: Not applicable. Leave at default.
- RFMT: Not applicable. Leave at default.
- AFSRCTL: Not applicable. Leave at default.
- ACLKRCTL: Not applicable. Leave at default.
- AHCLKRCTL: Not applicable. Leave at default.
- RTDM: Not applicable. Leave at default.
- RINTCTL: Not applicable. Leave at default.
- RCLKCHK: Not applicable. Leave at default.
- **XMASK**: Mask desired bits according to the discussion in this section, depending upon left-aligned or right-aligned internal data.

- **XFMT: XDATDLY** = 0. **XRVRS** = 0. **XPAD** = 0. **XPBIT** = default (not applicable). **XSSZ** = Fh (32-bit slot). XBUSEL = configured as desired. **XROT** bit is configured according to the discussion in this section, either 0 or 8-bit rotate.
- **AFSXCTL**: Configure the bits according to the discussion in this section.
- **ACLKXCTL**: **ASYNC** = 1. Program CLKXDIV bits to obtain the bit clock rate desired. Configure CLKXP and CLKXM bits as desired, because CLKX is not actually used in the DIT protocol.
- **AHCLKXCTL**: Program all fields according to high-frequency clock desired.
- **XTDM**: Set to FFFF FFFFh for all active slots for DIT transfers.
- XINTCTL: Program all fields according to interrupts desired.
- XCLKCHK: Program all fields according to clock checking desired.
- SRCTLn: Set **SRMOD** = 1 (transmitter) for the DIT pins. DISMOD field is don't care for DIT mode.
- **DITCSRA[n], DITCSRB[n]**: Program the channel status bits as desired.
- **DITUDRA[n], DITUDRB[n]**: Program the user data bits as desired.

24.3.8.3.3 DIT Channel Status and User Data Register Files

The channel status registers (DITCSRA*n* and DITCSRB*n*) and user data registers (DITUDRA*n* and DITUDRB*n*) are not double buffered. Typically the programmer uses one of the synchronizing interrupts, such as last slot, to create an event at a safe time so the register may be updated. In addition, the CPU reads the transmit TDM slot counter to determine which word of the register is being used.

It is a requirement that the software avoid writing to the word of user data and channel status that are being used to encode the current time slot; otherwise, it will be indeterminate whether the old or new data is used to encode the bitstream.

The DIT subframe format is defined in Section 24.3.3.2.2. The channel status information (C) and User Data (U) are defined in these DIT control registers:

- DITCSRA0 to DITCSRA5: The 192 bits in these six registers contain the channel status information for the LEFT channel within each frame.
- DITCSRB0 to DITCSRB5: The 192 bits in these six registers contain the channel status information for the RIGHT channel within each frame.
- DITUDRA0 to DITUDRA5: The 192 bits in these six registers contain the user data information for the LEFT channel within each frame.
- DITUDRB0 to DITUDRB5: The 192 bits in these six registers contain the user data information for the RIGHT channel within each frame.

The S/PDIF block format is shown in Figure 24-13. There are 192 frames within a block (frame 0 to frame 191). Within each frame there are two subframes (subframe 1 and 2 for left and right channels, respectively). The channel status and user data information sent on each subframe is summarized in Table 24-8.

Table 24-8. Channel Status and User Data for Each DIT Block

Frame	Subframe	Preamble	Channel Status defined in:	User Data defined in:
Defined by DITCSRA0, DITCSRB0, DITUDRA0, DITUDRB0				
0	1 (L)	B	DITCSRA0[0]	DITUDRA0[0]
0	2 (R)	W	DITCSRB0[0]	DITUDRB0[0]
1	1 (L)	M	DITCSRA0[1]	DITUDRA0[1]
1	2 (R)	W	DITCSRB0[1]	DITUDRB0[1]
2	1 (L)	M	DITCSRA0[2]	DITUDRA0[2]
2	2 (R)	W	DITCSRB0[2]	DITUDRB0[2]
…	…	…	…	…
31	1 (L)	M	DITCSRA0[31]	DITUDRA0[31]
31	2 (R)	W	DITCSRB0[31]	DITUDRB0[31]
Defined by DITCSRA1, DITCSRB1, DITUDRA1, DITUDRB1				
32	1 (L)	M	DITCSRA1[0]	DITUDRA1[0]
32	2 (R)	W	DITCSRB1[0]	DITUDRB1[0]
…	…	…	…	…
63	1 (L)	M	DITCSRA1[31]	DITUDRA1[31]
63	2 (R)	W	DITCSRB1[31]	DITUDRB1[31]
Defined by DITCSRA2, DITCSRB2, DITUDRA2, DITUDRB2				
64	1 (L)	M	DITCSRA2[0]	DITUDRA2[0]
64	2 (R)	W	DITCSRB2[0]	DITUDRB2[0]
…	…	…	…	…
95	1 (L)	M	DITCSRA2[31]	DITUDRA2[31]
95	2 (R)	W	DITCSRB2[31]	DITUDRB2[31]
Defined by DITCSRA3, DITCSRB3, DITUDRA3, DITUDRB3				
96	1 (L)	M	DITCSRA3[0]	DITUDRA3[0]
96	2 (R)	W	DITCSRB3[0]	DITUDRB3[0]
…	…	…	…	…
127	1 (L)	M	DITCSRA3[31]	DITUDRA3[31]
127	2 (R)	W	DITCSRB3[31]	DITUDRB3[31]
Defined by DITCSRA4, DITCSRB4, DITUDRA4, DITUDRB4				
128	1 (L)	M	DITCSRA4[0]	DITUDRA4[0]
128	2 (R)	W	DITCSRB4[0]	DITUDRB4[0]
…	…	…	…	…
159	1 (L)	M	DITCSRA4[31]	DITUDRA4[31]
159	2 (R)	W	DITCSRB4[31]	DITUDRB4[31]
Defined by DITCSRA5, DITCSRB5, DITUDRA5, DITUDRB5				
160	1 (L)	M	DITCSRA5[0]	DITUDRA5[0]
160	2 (R)	W	DITCSRB5[0]	DITUDRB5[0]
…	…	…	…	…
191	1 (L)	M	DITCSRA5[31]	DITUDRA5[31]
191	2 (R)	W	DITCSRB5[31]	DITUDRB5[31]

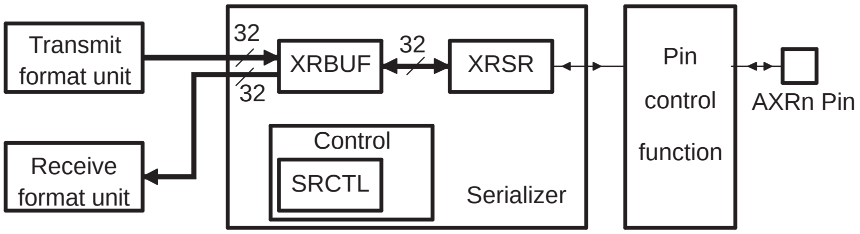
24.3.9 General Architecture

24.3.9.1 Serializers

Figure 24-22 shows the block diagram of the serializer and its interface to other units within the McASP. The serializers take care of shifting serial data in and out of the McASP. Each serializer consists of a shift register (XRSR), a data buffer (XRBUF), a control register (SRCTL), and logic to support the data alignment options of the McASP. For each serializer, there is a dedicated serial data pin (AXRn) and a dedicated control register (SRCTL[n]). The control register allows the serializer to be configured as a transmitter, receiver, or as inactive. When configured as a transmitter the serializer shifts out data to the serial data pin AXRn. When configured as a receiver, the serializer shifts in data from the AXRn pin. The serializer is clocked from the transmit/receive section clock (ACLKX/ACLKR) if configured to transmit/receive respectively.

All serializers that are configured to transmit operate in lock-step. Similarly, all serializers that are configured to receive also operate in lock-step. This means that at most there are two zones per McASP, one for transmit and one for receive.

Figure 24-22. Individual Serializer and Connections Within McASP



For receive, data is shifted in through the AXRn pin to the shift register XRSR. Once the entire slot of data is collected in the XRSR, the data is copied to the data buffer XRBUF. The data is now ready to be read by the processor through the RBUF register, which is an alias of the XRBUF for receive. When the processor reads from the RBUF, the McASP passes the data from RBUF through the receive format unit and returns the formatted data to the processor.

For transmit, the processor services the McASP by writing data into the XBUF register, which is an alias of the XRBUF for transmit. The data automatically passes through the transmit format unit before actually reaching the XRBUF register in the serializer. The data is then copied from XRBUF to XRSR, and shifted out from the AXRn synchronously to the serial clock.

In DIT mode, in addition to the data, the serializer shifts out other DIT-specific information accordingly (preamble, user data, etc.).

The serializer configuration is controlled by SRCTL[n].

24.3.9.2 Format Unit

The McASP has two data formatting units, one for transmit and one for receive. These units automatically remap the data bits within the transmitted and received words between a natural format for the processor (such as a Q31 representation) and the required format for the external serial device (such as "I2S format"). During the remapping process, the format unit also can mask off certain bits or perform sign extension.

Since all transmitters share the same data formatting unit, the McASP only supports one transmit format at a time. For example, the McASP will not transmit in "I2S format" on serializer 0, while transmitting "Left Justified" on serializer 1. Likewise, the receiver section of the McASP only supports one data format at a time, and this format applies to all receiving serializers. However, the McASP can transmit in one format while receiving in a completely different format.

This formatting unit consists of three stages:

- Bit mask and pad (masks off bits, performs sign extension)
- Rotate right (aligns data within word)
- Bit reversal (selects between MSB first or LSB first)

Figure 24-23 shows a block diagram of the receive formatting unit, and Figure 24-24 shows the transmit formatting unit. Note that the order in which data flows through the three stages is different between the transmit and receive formatting units.

Figure 24-23. Receive Format Unit

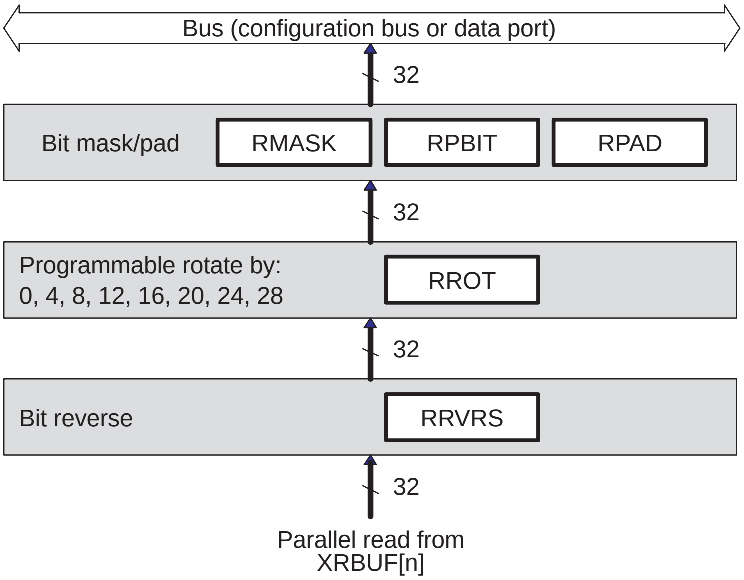
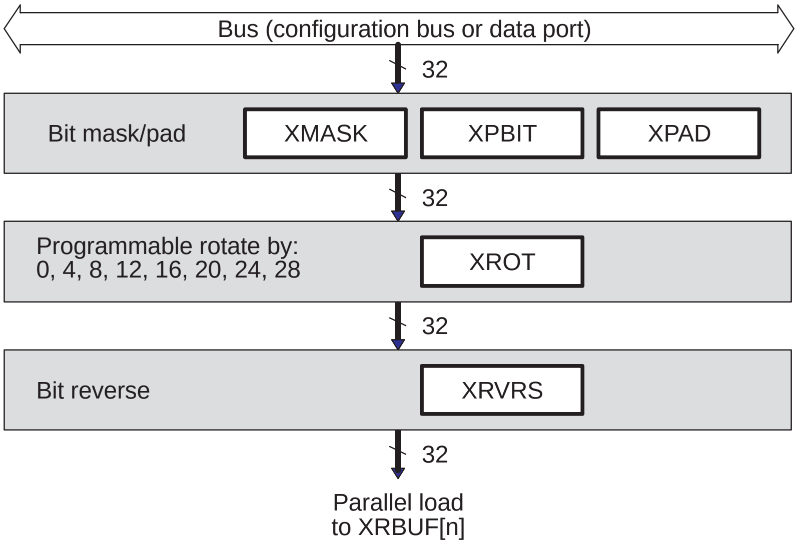


Figure 24-24. Transmit Format Unit



The bit mask and pad stage includes a full 32-bit mask register, allowing selected individual bits to either pass through the stage unchanged, or be masked off. The bit mask and pad then pad the value of the masked off bits by inserting either a 0, a 1, or one of the original 32 bits as the pad value. The last option allows for sign-extension when the sign bit is selected to pad the remaining bits.

The rotate right stage performs bitwise rotation by a multiple of 4 bits (between 0 and 28 bits), programmable by the (R/X)FMT register. Note that this is a rotation process, not a shifting process, so bit 0 gets shifted back into bit 31 during the rotation.

The bit reversal stage either passes all 32 bits directly through, or swaps them. This allows for either MSB or LSB first data formats. If bit reversal is not enabled, then the McASP will naturally transmit and receive in an LSB first order.

Finally, note that the (R/X)DATDLY bits in (R/X)FMT also determine the data format. For example, the difference between I2S format and left-justified is determined by the delay between the frame sync edge and the first data bit of a given time slot. For I2S format, (R/X)DATDLY should be set to a 1-bit delay, whereas for left-justified format, it should be set to a 0-bit delay.

The combination of all the options in (R/X)FMT means that the McASP supports a wide variety of data formats, both on the serial data lines, and in the internal processor representation.

Section 24.3.10.3 provides more detail and specific examples. The examples use internal representation in integer and Q31 notation, but other fractional notations are also possible.

24.3.9.3 State Machine

The receive and transmit sections have independent state machines. Each state machine controls the interactions between the various units in the respective section. In addition, the state machine keeps track of error conditions and serial port status.

No serial transfers can occur until the respective state machine is released from reset. See initialization sequence for details (Section 24.3.12).

The receive state machine is controlled by the RFMT register, and it reports the McASP status and error conditions in the RSTAT register. Similarly, the transmit state machine is controlled by the XFMT register, and it reports the McASP status and error conditions in the XSTAT register.

24.3.9.4 TDM Sequencer

There are separate TDM sequencers for the transmit section and the receive section. Each TDM sequencer keeps track of the slot count. In addition, the TDM sequencer checks the bits of (R/X)TDM and determines if the McASP should receive/transmit in that time slot.

If the McASP should participate (transmit/receive bit is active) in the time slot, the McASP functions normally. If the McASP should not participate (transmit/receive bit is inactive) in the time slot, no transfers between the XRBUF and XRSR registers in the serializer would occur during that time slot. In addition, the serializers programmed as transmitters place their data output pins in a predetermined state (logic low, high, or high impedance) as programmed by each serializer control register (SRCTL). Refer also to Section 24.3.8.2 for details on how DMA event or interrupt generations are handled during inactive time slots in TDM mode.

The receive TDM sequencer is controlled by register RTDM and reports current receive slot to RSLOT. The transmit TDM sequencer is controlled by register XTDM and reports current transmit slot to XSLOT.

24.3.9.5 Clock Check Circuit

A common source of error in audio systems is a serial clock failure due to instabilities in the off-chip DIR circuit. To detect a clock error quickly, a clock-check circuit is included in the McASP for both transmit and receive clocks, since both may be sourced from off chip.

The clock check circuit can detect and recover from transmit and receive clock failures. See Section 24.3.10.4.6 for implementation and programming details.

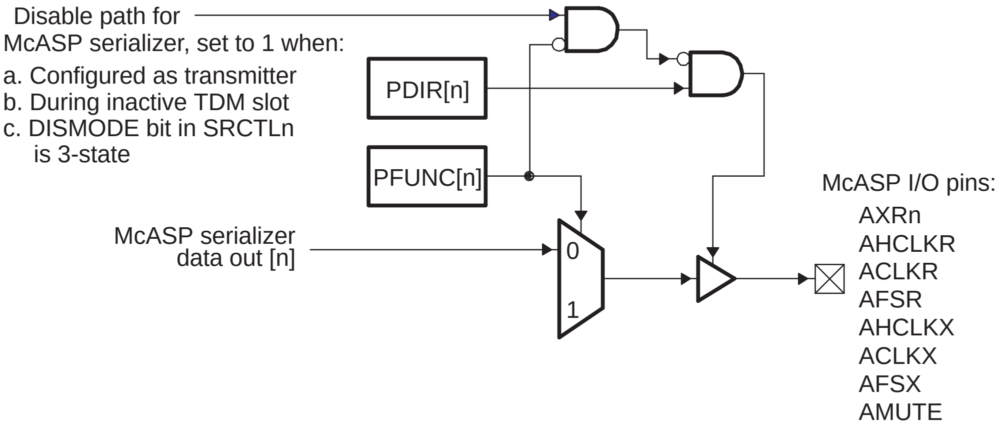
24.3.9.6 Pin Function Control

All McASP pins except AMUTEIN are bidirectional input/output pins. In addition, these bidirectional pins function either as McASP or general-purpose I/O (GPIO) pins. The following registers control the pin functions:

- Pin function register (PFUNC): selects pin to function as McASP or GPIO.
- Pin direction register (PDIR): selects pin to be input or output.
- Pin data input register (PDIN): shows data input at the pin.
- Pin data output register (PDOUT): data to be output at the pin if the pin is configured as GPIO output (PFUNC[n] = 1 and PDIR[n] = 1). Not applicable when the pin is configured as McASP pin (PFUNC[n] = 0).
- Pin data set register (PDSET): alias of PDOUT. Writing a 1 to PDSET[n] sets the respective PDOUT[n] to 1. Writing a 0 has no effect. Applicable only when the pin is configured as GPIO output (PFUNC[n] = 1 and PDIR[n] = 1).
- Pin data clear register (PDCLR): alias of PDOUT. Writing a 1 to PDCLR[n] clears the respective PDOUT[n] to 0. Writing a 0 has no effect. Applicable only when the pin is configured as GPIO output (PFUNC[n] = 1 and PDIR[n] = 1).

See the register descriptions in for details on the mapping of each McASP pin to the register bits. Figure 24-25 shows the pin control block diagram.

Figure 24-25. McASP I/O Pin Control Block Diagram



24.3.9.6.1 McASP Pin Control-Transmit and Receive

You must correctly set the McASP GPIO registers PFUNC and PDIR, even when McASP pins are used for their serial port (non-GPIO) function.

Serial port functions include:

- Clock pins (ACLKX, ACLKR, AHCLKX, AHCLKR, AFSX, AFSR) used as clock inputs and outputs.
- Serializer data pins (AXRn) used to transmit or receive.
- AMUTE used as a mute output signal.

When using these pins in their serial port function, you must clear PFUNC[n] to 0 for each pin.

Also, certain outputs require PDIR[n] = 1, such as clock pins used as clock outputs, serializer data pins used to transmit, and AMUTE used as mute output.

Clock inputs and serializers configured to receive must have PDIR[n] = 0.

PFUNC and PDIR do not control the AMUTEIN signal, it is usually tied to a device level interrupt pin (consult device datasheet). If used as a mute input, this pin needs to be configured as an input in the appropriate peripheral.

Finally, there is an important advantage to having separate control of pin direction (by PDIR), and the choice of internal versus external clocking (by CLKRM/CLKXM). Depending on the specific device and usage, you might select an external clock (CLKRM = 0), while enabling the internal clock divider, and the clock pin as an output in the PDIR register (PDIR[ACLKR] = 1). In this case, the bit clock is an output (PDIR[ACLKR] = 1) and, therefore, routed to the ACLKR pin. However, because CLKRM = 0, the bit clock is then routed back to the McASP module as an "external" clock source. This may result in less skew between the clock inside the McASP and the clock in the external device, thus producing more balanced setup and hold times for a particular system. As a result, this may allow a higher serial clock rate interface.

24.3.10 Operation

This section discusses the operation of the McASP.

24.3.10.1 Data Transmission and Reception

The processor services the McASP by writing data to the XBUF register(s) for transmit operations, and by reading data from the RBUF register(s) for receive operations. The McASP sets status flag and notifies the processor whenever data is ready to be serviced. Section 24.3.10.1.1 discusses data ready status in detail.

The XBUF and RBUF registers can be accessed through one of the two peripheral ports of the device:

- The data port (DAT): This port is dedicated for data transfers on the device.
- The configuration bus (CFG): This port is used for both data transfers and peripheral configuration control on the device.

Section 24.3.10.1.2 and Section 24.3.10.1.3 discuss how to perform transfers through the data port and the configuration bus.

Either the CPU or the DMA can be used to service the McASP through any of these two peripheral ports. The CPU and DMA usages are discussed in Section 24.3.10.1.4 and Section 24.3.14.2.

24.3.10.1.1 Data Ready Status and Event/Interrupt Generation

24.3.10.1.1.1 Transmit Data Ready

The transmit data ready flag XDATA bit in the XSTAT register reflects the status of the XBUF register. The XDATA flag is set when data is transferred from the XRBUF[n] buffers to the XRSR[n] shift registers, indicating that the XBUF is empty and ready to accept new data from the processor. This flag is cleared when the XDATA bit is written with a 1, or when all the serializers configured as transmitters are written by the processor.

Whenever XDATA is set, an DMA event AXEVT is automatically generated to notify the DMA of the XBUF empty status. An interrupt AXINT is also generated if XDATA interrupt is enabled in the XINTCTL register (See Section 24.3.13.2 for details).

For DMA requests, the McASP does not require XSTAT to be read between DMA events. This means that even if XSTAT already has the XDATA flag set to 1 from a previous request, the next transfer triggers another DMA request.

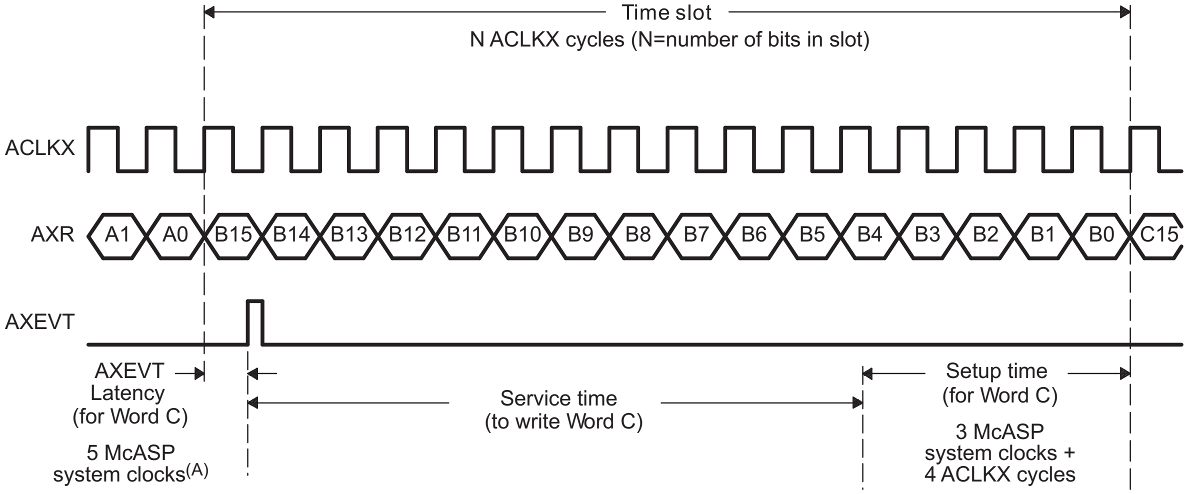
Since all serializers act in lockstep, only one DMA event is generated to indicate that all active transmit serializers are ready to be written to with new data.

Figure 24-26 shows the timing details of when AXEVT is generated at the McASP boundary. In this example, as soon as the last bit (bit A0) of Word A is transmitted, the McASP sets the XDATA flag and generates an AXEVT event. However, it takes up to 5 McASP system clocks (AXEVT Latency) before AXEVT is active at the McASP boundary. Upon AXEVT, the processor can begin servicing the McASP by writing Word C into the XBUF (Processor Service Time). The processor must write Word C into the XBUF no later than the setup time required by the McASP (Setup Time).

The maximum Processor Service Time (Figure 24-26) can be calculated as:

Processor Service Time = Time Slot - AXEVT Latency - Setup Time

Figure 24-26. Processor Service Time Upon Transmit DMA Event (AXEVT)



A Refer to the device-specific data manual for the McASP system clock source. This is not the same as AUXCLK.

Example 24-1. Processor Service Time Calculation for Transmit DMA Event (AXEVT)

The following is an example to show how to calculate Processor Service Time. Assume the following setup:

- McASP transmits in I2S format at 192 kHz frame rate. Assume slot size is 32 bit.

With the above setup, we obtain the following parameters corresponding to Figure 24-26:

- Calculation of McASP system clock cycle:
  - System functional clock = 26 MHz
  - Therefore, McASP system clock cycle = 1/26MHz = 38.5 ns.
- Calculation of ACLKX clock cycle:
  - This example has two 32-bit slots per frame, for a total of 64 bits per frame.
  - ACLKX clock cycle is (1/192 kHz)/64 = 81.4 ns.
- Time Slot between AXEVT events:
  - For I2S format, McASP generates two AXEVT events per 192 kHz frame.
  - Therefore, Time Slot between AXEVT events is (1/192 kHz)/2 = 2604 ns.
- AXEVT Latency:
  = 5 McASP system clocks
  = 38.5 ns × 5 = 192.5 ns
- Setup Time
  = 3 McASP system clocks + 4 ACLKX cycles
  = (38.5 ns × 3) + (81.4 ns × 4)
  = 441.1 ns
- Processor Service Time
  = Time Slot - AXEVT Latency - Setup Time
  = 2604 ns - 441.1 ns - 192.5 ns
  = 1970.4 ns

24.3.10.1.1.2 Receive Data Ready

Similarly, the receive data ready flag RDATA bit in the RSTAT reflects the status of the RBUF register. The RDATA flag is set when data is transferred from the XRSR[n] shift registers to the XRBUF[n] buffers, indicating that the RBUF contains received data and is ready to have the processor read the data. This flag is cleared when the RDATA bit is written with a 1, or when all the serializers configured as receivers are read.

Whenever RDATA is set, an DMA event AREVT is automatically generated to notify the DMA of the RBUF ready status. An interrupt ARINT is also generated if RDATA interrupt is enabled in the RINTCTL register (See Section 24.3.13.3 for details).

For DMA requests, the McASP does not require RSTAT to be read between DMA events. This means that even if RSTAT already has the RDATA flag set to 1 from a previous request, the next transfer triggers another DMA request.

Since all serializers act in lockstep, only one DMA event is generated to indicate that all active receive serializers are ready to receive new data.

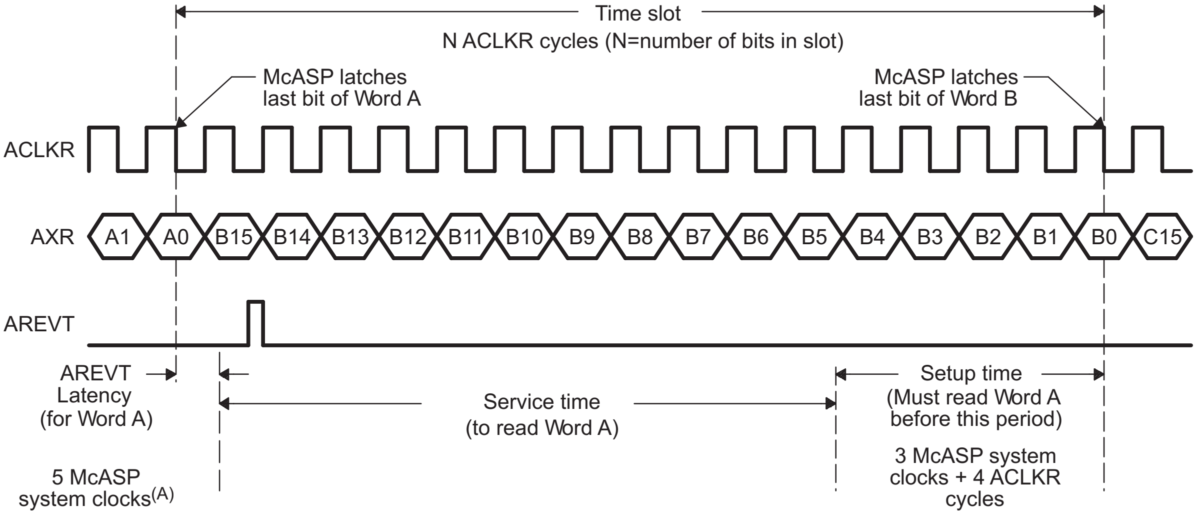
Figure 24-27 shows the timing details of when AREVT is generated at the McASP boundary. In this example, as soon as the last bit (bit A0) of Word A is received, the McASP sets the RDATA flag and generates an AREVT event. However, it takes up to 5 McASP system clocks (AREVT Latency) before AREVT is active at the McASP boundary. Upon AREVT, the processor can begin servicing the McASP by reading Word A from the RBUF (Processor Service Time). The processor must read Word A from the XBUF no later than the setup time required by the McASP (Setup Time).

The maximum Processor Service Time (Figure 24-27) can be calculated as:

Processor Service Time = Time Slot - AREVT Latency - Setup Time

The Processor Service Time calculation for receive is similar to the calculation for transmit. See Example 24-1 for Processor Service Time calculation using transmit as an example.

Figure 24-27. Processor Service Time Upon Receive DMA Event (AREVT)



A The device uses SYSCLK2 as the McASP system clock source.

24.3.10.1.2 Transfers Through the Data Port (DAT)

NOTE: To perform internal transfers through the data port, clear XBUSEL/RBUSEL bit to 0 in the respective XFMT/RFMT registers. Failure to do so will result in software malfunction.

Typically, you will access the McASP XRBUF registers through the data port. To access through the data port, simply have the CPU or DMA access the XRBUF through its data port location. Through the data port, the DMA/CPU can service all the serializers through a single address. The McASP automatically cycles through the appropriate serializers.

For transmit operations through the data port, the DMA/CPU should write to the same XBUF data port address to service all of the active transmit serializers. In addition, the DMA/CPU should write to the XBUF for all active transmit serializers in incremental (although not necessarily consecutive) order. For example, if serializers 0, 4, and 5, are set up as active transmitters, the DMA/CPU should write to the XBUF data port address four times with data for serializers 0, 4, and 5, upon each transmit data ready event. This exact servicing order must be followed so that data appears in the appropriate serializers.

Similarly, for receive operations through the data port, the DMA/CPU should read from the same RBUF data port address to service all of the active receive serializers. In addition, reads from the active receive serializers through the data port return data in incremental (although not necessarily consecutive) order. For example, if serializers 1, 2, and 3, are set up as active receivers, the DMA/CPU should read from the RBUF data port address four times to obtain data for serializers 1, 2, and 3, in this exact order, upon each receive data ready event.

When transmitting, the DMA/CPU must write data to each serializer configured as "active" and "transmit" within each time slot. Failure to do so results in a buffer underrun condition (Section 24.3.10.4.2). Similarly, when receiving, data must be read from each serializer configured as "active" and "receive" within each time slot. Failure to do results in a buffer overrun condition (Section 24.3.10.4.3).

To perform internal transfers through the data port, clear XBUSEL/RBUSEL bit to 0 in the respective XFMT/RFMT registers.

24.3.10.1.3 Transfers Through the Configuration Bus (CFG)

NOTE: To perform internal transfers through the configuration bus, set XBUSEL/RBUSEL bit to 1 in the respective XFMT/RFMT registers. Failure to do so will result in software malfunction.

In this method, the DMA/CPU accesses the XRBUF registers through the configuration bus address. The exact XRBUF register address for any particular serializer is determined by adding the offset for that particular serializer to the base address for the particular McASP. XRBUF for the serializers configured as transmitters is given the name XBUF*n*. For example, the XRBUF associated with transmit serializer 2 is named XBUF2. Similarly, XRBUF for the serializers configured as receivers is given the name RBUF*n*.

Accessing the XRBUF registers through the data port is different because the CPU/DMA only needs to access one single address. When accessing through the configuration bus, the CPU/DMA must provide the exact XBUF*n* or RBUF*n* address for each access.

When transmitting, DMA/CPU must write data to each serializer configured as "active" and "transmit" within each time slot. Failure to do so results in a buffer underrun condition (Section 24.3.10.4.2). Similarly when receiving, data must be read from each serializer configured as "active" and "receive" within each time slot. Failure to do results in a buffer overrun condition (Section 24.3.10.4.3).

To perform internal transfers through the configuration bus, set XBUSEL/RBUSEL bit to 1 in the respective XFMT/RFMT registers.

24.3.10.1.4 Using the CPU for McASP Servicing

The CPU can be used to service the McASP through interrupt (upon AXINT/ARINT interrupts) or through polling the XDATA bit in the XSTAT register. As discussed in Section 24.3.10.1.2 and Section 24.3.10.1.3, the CPU can access through either the data port or through the configuration bus.

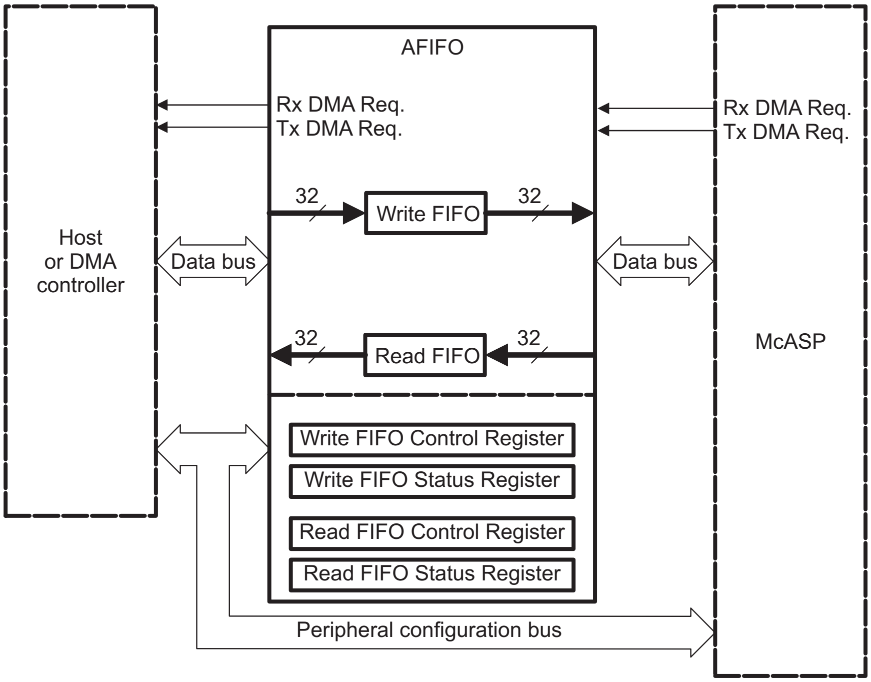
To use the CPU to service the McASP through interrupts, the XDATA/RDATA bit must be enabled in the respective XINTCTL/RINTCTL registers, to generate interrupts AXINT/ARINT to the CPU upon data ready.

24.3.10.2 McASP Audio FIFO (AFIFO)

The AFIFO contains two FIFOs: one Read FIFO (RFIFO), and one Write FIFO (WFIFO). To ensure backward compatibility with existing software, both the Read and Write FIFOs are disabled by default. See Figure 24-28 for a high-level block diagram of the AFIFO.

The AFIFO may be enabled/disabled and configured via the WFIFOCTL and RFIFOCTL registers. Note that if the Read or Write FIFO is to be enabled, it must be enabled prior to initializing the receive/transmit section of the McASP (see Section 24.3.12.2 for details).

Figure 24-28. McASP Audio FIFO (AFIFO) Block Diagram



24.3.10.2.1 AFIFO Data Transmission

When the Write FIFO is disabled, transmit DMA requests pass through directly from the McASP to the host/DMA controller. Whether the WFIFO is enabled or disabled, the McASP generates transmit DMA requests as needed; the AFIFO is “invisible” to the McASP.

When the Write FIFO is enabled, transmit DMA requests from the McASP are sent to the AFIFO, which in turn generates transmit DMA requests to the host/DMA controller.

If the Write FIFO is enabled, upon a transmit DMA request from the McASP, the WFIFO writes *WNUMDMA* 32-bit words to the McASP if and when there are at least *WNUMDMA* words in the Write FIFO. If there are not, the WFIFO waits until this condition has been satisfied. At that point, it writes *WNUMDMA* words to the McASP. (See description for WFIFOCTL.WNUMDMA.)

If the host CPU writes to the Write FIFO, independent of a transmit DMA request, the WFIFO will accept host writes until full. After this point, excess data will be discarded.

Note that when the WFIFO is first enabled, it will immediately issue a transmit DMA request to the host. This is because it begins in an empty state, and is therefore ready to accept data.

24.3.10.2.1.1 Transmit DMA Event Pacer

The AFIFO may be configured to delay making a transmit DMA request to the host until the Write FIFO has enough space for a specified number of words. In this situation, the number of transmit DMA requests to the host or DMA controller is reduced.

If the Write FIFO has space to accept *WNUMEVT* 32-bit words, it generates a transmit DMA request to the host and then waits for a response. Once *WNUMEVT* words have been written to the FIFO, it checks again to see if there is space for *WNUMEVT* 32-bit words. If there is space, it generates another transmit DMA request to the host, and so on. In this fashion, the Write FIFO will attempt to stay filled.

Note that if transmit DMA event pacing is desired, WFIFOCTL.WNUMEVT should be set to a non-zero integer multiple of the value in WFIFOCTL.WNUMDMA. If transmit DMA event pacing is not desired, then the value in WFIFOCTL.WNUMEVT should be set equal to the value in WFIFOCTL.WNUMDMA.

24.3.10.2.2 AFIFO Data Reception

When the Read FIFO is disabled, receive DMA requests pass through directly from McASP to the host/DMA controller. Whether the RFIFO is enabled or disabled, the McASP generates receive DMA requests as needed; the AFIFO is “invisible” to the McASP.

When the Read FIFO is enabled, receive DMA requests from the McASP are sent to the AFIFO, which in turn generates receive DMA requests to the host/DMA controller.

If the Read FIFO is enabled and the McASP makes a receive DMA request, the RFIFO reads *RNUMDMA* 32-bit words from the McASP, if and when the RFIFO has space for *RNUMDMA* words. If it does not, the RFIFO waits until this condition has been satisfied; at that point, it reads *RNUMDMA* words from the McASP. (See description for RFIFOCTL.RNUMDMA.)

If the host CPU reads the Read FIFO, independent of a receive DMA request, and the RFIFO at that time contains less than *RNUMEVT* words, those words will be read correctly, emptying the FIFO.

24.3.10.2.2.1 Receive DMA Event Pacer

The AFIFO may be configured to delay making a receive DMA request to the host until the Read FIFO contains a specified number of words. In this situation, the number of receive DMA requests to the host or DMA controller is reduced.

If the Read FIFO contains at least *RNUMEVT* 32-bit words, it generates a receive DMA request to the host and then waits for a response. Once *RNUMEVT* 32-bit words have been read from the RFIFO, the RFIFO checks again to see if it contains at least another *RNUMEVT* words. If it does, it generates another receive DMA request to the host, and so on. In this fashion, the Read FIFO will attempt to stay empty.

Note that if receive DMA event pacing is desired, RFIFOCTL.RNUMEVT should be set to a non-zero integer multiple of the value in RFIFOCTL.RNUMDMA. If receive DMA event pacing is not desired, then the value in RFIFOCTL.RNUMEVT should be set equal to the value in RFIFOCTL.RNUMDMA.

24.3.10.2.3 Arbitration Between Transmit and Receive DMA Requests

If both the WFIFO and the RFIFO are enabled and a transmit DMA request and receive DMA request occur simultaneously, priority is given to the transmit DMA request. Once a transfer is in progress, it is allowed to complete.

If only the WFIFO is enabled and a transmit DMA request and receive DMA request occur simultaneously, priority is given to the transmit DMA request. Once a transfer is in progress, it is allowed to complete.

If only the RFIFO is enabled and a transmit DMA request and receive DMA request occur simultaneously, priority is given to the receive DMA request. Once a transfer is in progress, it is allowed to complete.

24.3.10.3 Formatter

24.3.10.3.1 Transmit Bit Stream Data Alignment

The McASP transmitter supports serial formats of:

- Slot (or Time slot) size = 8, 12, 16, 20, 24, 28, 32 bits.
- Word size ≤ Slot size.
- Alignment: when more bits/slot than bits/words, then:
  - Left aligned = word shifted first, remaining bits are pad.
  - Right aligned = pad bits are shifted first, word occupies the last bits in slot.
- Order: order of bits shifted out:
  - MSB: most-significant bit of word is shifted out first, last bit is LSB.
  - LSB: least-significant bit of word is shifted out last, last bit is MSB.

Hardware support for these serial formats comes from the programmable options in the transmit bitstream format register (XFMT):

- XRVRS: bit reverse (1) or no bit reverse (0).
- XROT: rotate right by 0, 4, 8, 12, 16, 20, 24, or 28 bits.
- XSSZ: transmit slot size of 8, 12, 16, 20, 24, 28, or 32 bits.

XSSZ should always be programmed to match the slot size of the serial stream. The word size is not directly programmed into the McASP, but rather is used to determine the rotation needed in the XROT field.

Table 24-9 and Figure 24-29 show the XRVRS and XROT fields for each serial format and for both integer and Q31 fractional internal representations.

This discussion assumes that all slot size (SLOT in Table 24-9) and word size (WORD in Table 24-9) options are multiples of 4, since the transmit rotate right unit only supports rotation by multiples of 4. However, the bit mask/pad unit does allow for any number of significant digits. For example, a Q31 number may have 19 significant digits (word) and be transmitted in a 24-bit slot; this would be formatted as a word size of 20 bits and a slot size of 24 bits. However, it is possible to set the bit mask unit to only pass the 19 most-significant digits (program the mask value to FFFF E000h). The digits that are not significant can be set to a selected pad value, which can be any one of the significant digits, a fixed value of 0, or a fixed value of 1.

The transmit bit mask/pad unit operates on data as an initial step of the transmit format unit (see Figure 24-24), and the data is aligned in the same representation as it is written to the transmitter by the processor (typically Q31 or integer).

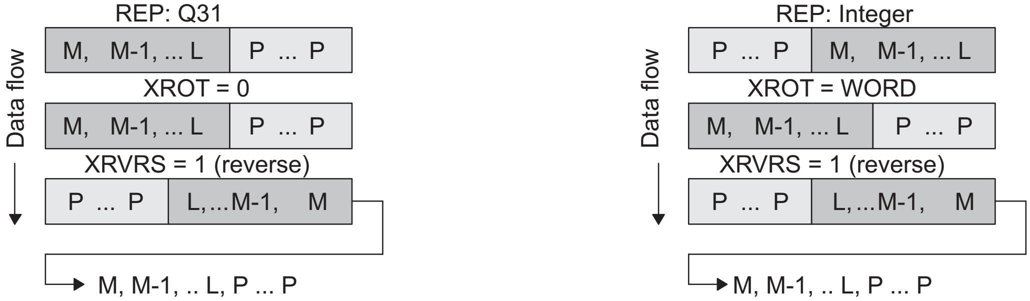
Table 24-9. Transmit Bitstream Data Alignment

Figure 24-29	Bit Stream Order	Bit Stream Alignment	Internal Numeric Representation	XFMT Bit	
				XROT[1]	XRVRS
(a)[2]	MSB first	Left aligned	Q31 fraction	0	1
(b)	MSB first	Right aligned	Q31 fraction	SLOT - WORD	1
(c)	LSB first	Left aligned	Q31 fraction	32 - WORD	0
(d)	LSB first	Right aligned	Q31 fraction	32 - SLOT	0
(e)[2]	MSB first	Left aligned	Integer	WORD	1
(f)	MSB first	Right aligned	Integer	SLOT	1
(g)	LSB first	Left aligned	Integer	0	0
(h)	LSB first	Right aligned	Integer	(32 - (SLOT - WORD)) % 32	0

[1] WORD = Word size rounded up to the nearest multiple of 4; SLOT = slot size; % = modulo operator

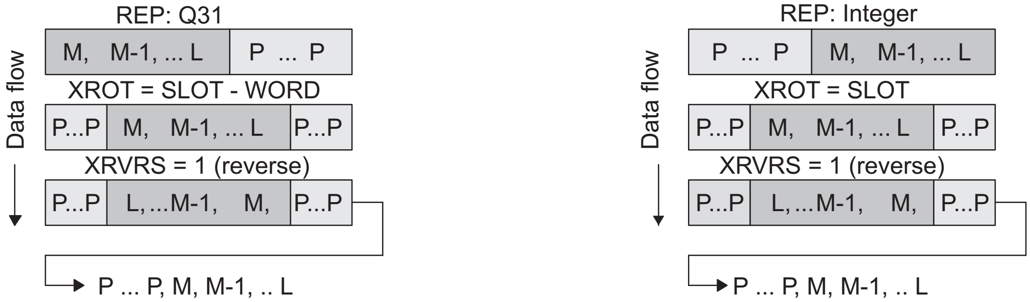
[2] To transmit in I2S format, use MSB first, left aligned, and also select XDATDLY = 01 (1 bit delay)

Figure 24-29. Data Flow Through Transmit Format Unit, Illustrated



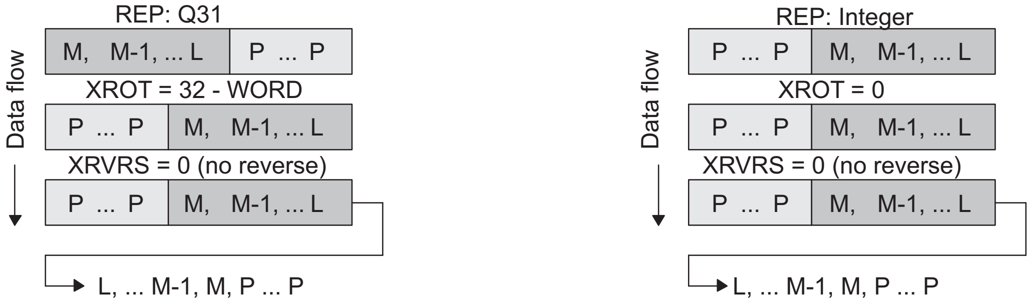
(a) Out: MSB first, LEFT aligned

(e) Out: MSB first, LEFT aligned



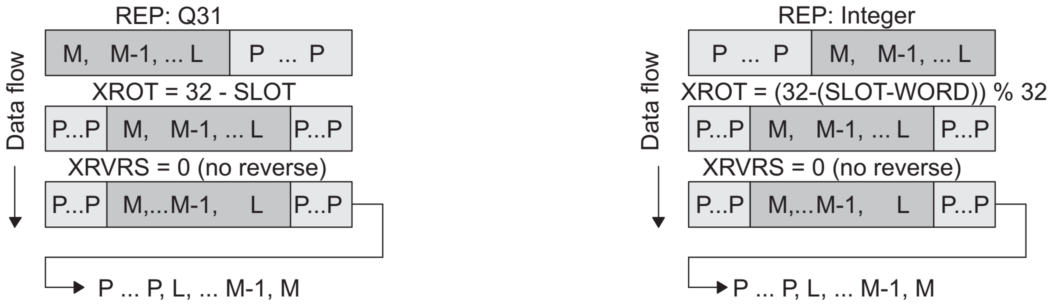
(b) Out: MSB first, RIGHT aligned

(f) Out: MSB first, RIGHT aligned



(c) Out: LSB first, LEFT aligned

(g) Out: LSB first, LEFT aligned



(d) Out: LSB first, RIGHT aligned

(h) Out: LSB first, RIGHT aligned

24.3.10.3.2 Receive Bit Stream Data Alignment

The McASP receiver supports serial formats of:

- Slot or time slot size = 8, 12, 16, 20, 24, 28, 32 bits.
- Word size ≤ Slot size.
- Alignment when more bits/slot than bits/words, then:
  - Left aligned = word shifted first, remaining bits are pad.
  - Right aligned = pad bits are shifted first, word occupies the last bits in slot.
- Order of bits shifted out:
  - MSB: most-significant bit of word is shifted out first, last bit is LSB.
  - LSB: least-significant bit of word is shifted out last, last bit is MSB.

Hardware support for these serial formats comes from the programmable options in the receive bitstream format register (RFMT):

- RRVRS: bit reverse (1) or no bit reverse (0).
- RROT: rotate right by 0, 4, 8, 12, 16, 20, 24, or 28 bits.
- RSSZ: receive slot size of 8, 12, 16, 20, 24, 28, or 32 bits.

RSSZ should always be programmed to match the slot size of the serial stream. The word size is not directly programmed into the McASP, but rather is used to determine the rotation needed in the RROT field.

Table 24-10 and Figure 24-30 show the RRVRS and RROT fields for each serial format and for both integer and Q31 fractional internal representations.

This discussion assumes that all slot size and word size options are multiples of 4; since the receive rotate right unit only supports rotation by multiples of 4. However, the bit mask/pad unit does allow for any number of significant digits. For example, a Q31 number may have 19 significant digits (word) and be transmitted in a 24-bit slot; this would be formatted as a word size of 20 bits and a slot size of 24 bits. However, it is possible to set the bit mask unit to only pass the 19 most-significant digits (program the mask value to FFFF E000h). The digits that are not significant can be set to a selected pad value, which can be any one of the significant digits, a fixed value of 0, or a fixed value of 1.

The receive bit mask/pad unit operates on data as the final step of the receive format unit (see Figure 24-23), and the data is aligned in the same representation as it is read from the receiver by the processor (typically Q31 or integer).

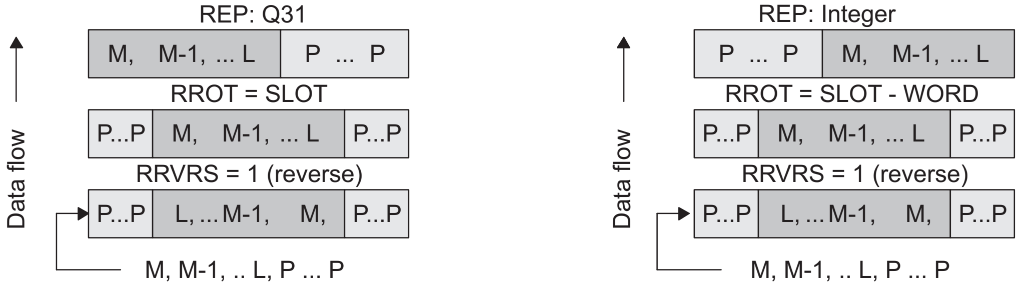
Table 24-10. Receive Bitstream Data Alignment

				RFMT Bit	
Figure 24-30	**Bit Stream Order**	**Bit Stream Alignment**	**Internal Numeric Representation**	**RROT**[1]	**RRVRS**
(a)[2]	MSB first	Left aligned	Q31 fraction	SLOT	1
(b)	MSB first	Right aligned	Q31 fraction	WORD	1
(c)	LSB first	Left aligned	Q31 fraction	(32 - (SLOT - WORD)) % 32	0
(d)	LSB first	Right aligned	Q31 fraction	0	0
(e)[2]	MSB first	Left aligned	Integer	SLOT - WORD	1
(f)	MSB first	Right aligned	Integer	0	1
(g)	LSB first	Left aligned	Integer	32 - SLOT	0
(h)	LSB first	Right aligned	Integer	32 - WORD	0

[1] WORD = Word size rounded up to the nearest multiple of 4; SLOT = slot size; % = modulo operator

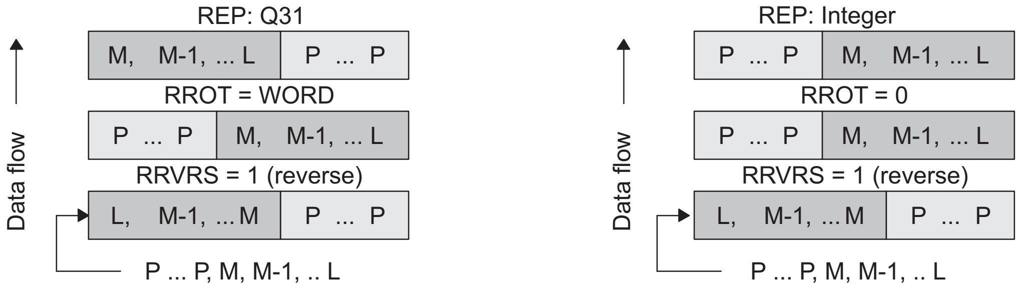
[2] To transmit in I2S format, select MSB first, left aligned, and also select RDATDLY = 01 (1 bit delay)

Figure 24-30. Data Flow Through Receive Format Unit, Illustrated



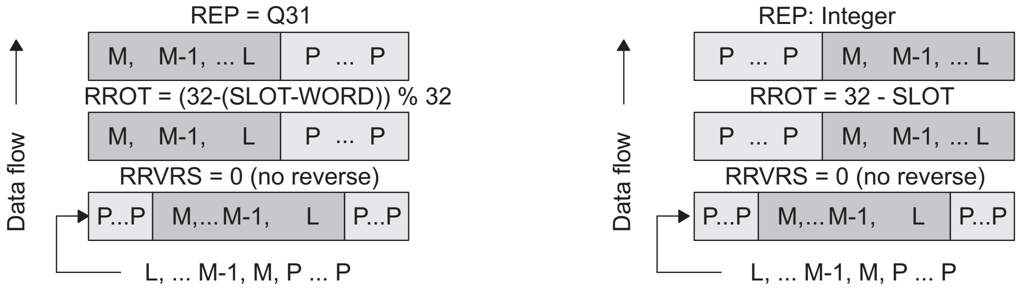
(a) In: MSB first, LEFT aligned

(e) In: MSB first, LEFT aligned



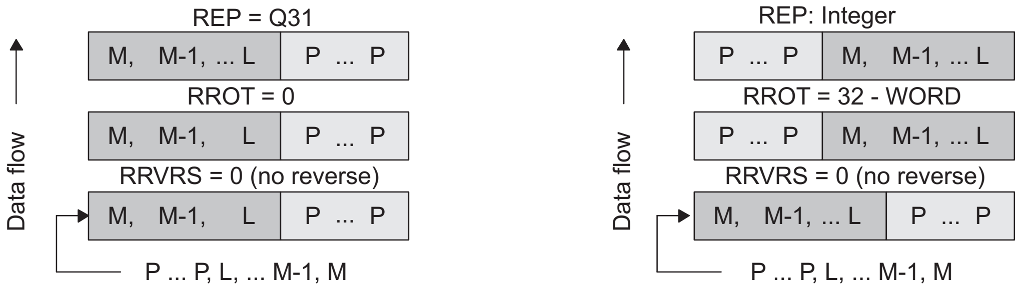
(b) In: MSB first, RIGHT aligned

(f) In: MSB first, RIGHT aligned



(c) In: LSB first, LEFT aligned

(g) In: LSB first, LEFT aligned



(d) In: LSB first, RIGHT aligned

(h) In: LSB first, RIGHT aligned

24.3.10.4 Error Handling and Management

To support the design of a robust audio system, the McASP includes error-checking capability for the serial protocol, data underrun, and data overrun. In addition, the McASP includes a timer that continually measures the high-frequency master clock every 32 AHCLKX/AHCLKR clock cycles. The timer value can be read to get a measurement of the clock frequency and has a minimum and maximum range setting that can set an error flag if the master clock goes out of a specified range.

Upon the detection of any one or more errors (software selectable), or the assertion of the AMUTEIN input pin, the AMUTE output pin may be asserted to a high or low level to immediately mute the audio output. In addition, an interrupt may be generated if desired, based on any one or more of the error sources.

24.3.10.4.1 Unexpected Frame Sync Error

An unexpected frame sync occurs when:

- In burst mode, when the next active edge of the frame sync occurs early such that the current slot will not be completed by the time the next slot is scheduled to begin.
- In TDM mode, a further constraint is that the frame sync must occur exactly during the correct bit clock (not a cycle earlier or later) and only before slot 0. An unexpected frame sync occurs if this condition is not met.

When an unexpected frame sync occurs, there are two possible actions depending upon when the unexpected frame sync occurs:

1. Early: An early unexpected frame sync occurs when the McASP is in the process of completing the current frame and a new frame sync is detected (not including overlap that occurs due to a 1 or 2 bit frame sync delay). When an early unexpected frame sync occurs:
   - Error interrupt flag is set (XSYNCERR, if an unexpected transmit frame sync occurs; RSYNCERR, if an unexpected receive frame sync occurs).
   - Current frame is not resynchronized. The number of bits in the current frame is completed. The next frame sync, which occurs after the current frame is completed, will be resynchronized.
2. Late: A late unexpected frame sync occurs when there is a gap or delay between the last bit of the previous frame and the first bit of the next frame. When a late unexpected frame sync occurs (as soon as the gap is detected):
   - Error interrupt flag is set (XSYNCERR, if an unexpected transmit frame sync occurs; RSYNCERR, if an unexpected receive frame sync occurs).
   - Resynchronization occurs upon the arrival of the next frame sync.

Late frame sync is detected the same way in both burst mode and TDM mode; however, in burst mode, late frame sync is not meaningful and its interrupt enable should not be set.

24.3.10.4.2 Buffer Underrun Error - Transmitter

A buffer underrun can only occur for serializers programmed to be transmitters. A buffer underrun occurs when the serializer is instructed by the transmit state machine to transfer data from XRBUF[n] to XRSR[n], but XRBUF[n] has not yet been written with new data since the last time the transfer occurred. When this occurs, the transmit state machine sets the XUNDRN flag.

An underrun is checked only once per time slot. The XUNDRN flag is set when an underrun condition occurs. Once set, the XUNDRN flag remains set until the processor explicitly writes a 1 to the XUNDRN bit to clear the XUNDRN bit.

In DIT mode, a pair of BMC zeros is shifted out when an underrun occurs (four bit times at 128 × fs). By shifting out a pair of zeros, a clock may be recovered on the receiver. To recover, reset the McASP and start again with the proper initialization.

In TDM mode, during an underrun case, a long stream of zeros are shifted out causing the DACs to mute. To recover, reset the McASP and start again with the proper initialization.

24.3.10.4.3 Buffer Overrun Error - Receiver

A buffer overrun can only occur for serializers programmed to be receivers. A buffer overrun occurs when the serializer is instructed to transfer data from XRSR[n] to XRBUF[n], but XRBUF[n] has not yet been read by either the DMA or the processor. When this occurs, the receiver state machine sets the ROVRN flag. However, the individual serializer writes over the data in the XRBUF[n] register (destroying the previous sample) and continues shifting.

An overrun is checked only once per time slot. The ROVRN flag is set when an overrun condition occurs. It is possible that an overrun occurs on one time slot but then the processor catches up and does not cause an overrun on the following time slots. However, once the ROVRN flag is set, it remains set until the processor explicitly writes a 1 to the ROVRN bit to clear the ROVRN bit.

24.3.10.4.4 DMA Error - Transmitter

A transmit DMA error, as indicated by the XDMAERR flag in the XSTAT register, occurs when the DMA (or CPU) writes more words to the DAT port of the McASP than it should. For each DMA event, the DMA should write exactly as many words as there are serializers enabled as transmitters.

XDMAERR indicates that the DMA (or CPU) wrote too many words to the McASP for a given transmit DMA event. Writing too few words results in a transmit underrun error setting XUNDRN in XSTAT.

While XDMAERR occurs infrequently, an occurrence indicates a serious loss of synchronization between the McASP and the DMA or CPU. You should reinitialize both the McASP transmitter and the DMA to resynchronize them.

24.3.10.4.5 DMA Error - Receiver

A receive DMA error, as indicated by the RDMAERR flag in the RSTAT register, occurs when the DMA (or CPU) reads more words from the DAT port of the McASP than it should. For each DMA event, the DMA should read exactly as many words as there are serializers enabled as receivers.

RDMAERR indicates that the DMA (or CPU) read too many words from the McASP for a given receive DMA event. Reading too few words results in a receiver overrun error setting ROVRN in RSTAT.

While RDMAERR occurs infrequently, an occurrence indicates a serious loss of synchronization between the McASP and the DMA or CPU. You should reinitialize both the McASP receiver and the DMA to resynchronize them.

24.3.10.4.6 Clock Failure Detection

24.3.10.4.6.1 Clock-Failure Check Startup

It is expected, initially, that the clock-failure circuits will generate an error until at least one measurement has been taken. Therefore, the clock failure interrupts, clock switch, and mute functions should not immediately be enabled, but be enabled only after a specific startup procedure. The startup procedure is:

1. For the transmit clock failure check:
   a. Configure transmit clock failure detect logic (XMIN, XMAX, XPS) in the transmit clock check control register (XCLKCHK).
   b. Clear transmit clock failure flag (XCKFAIL) in the transmit status register (XSTAT).
   c. Wait until first measurement is taken (> 32 AHCLKX clock periods).
   d. Verify no clock failure is detected.
   e. Repeat steps b–d until clock is running and is no longer issuing clock failure errors.
   f. After the transmit clock is measured and falls within the acceptable range, the following may be enabled:
      i. transmit clock failure interrupt enable bit (XCKFAIL) in the transmitter interrupt control register (XINTCTL).
      ii. transmit clock failure detect autoswitch enable bit (XCKFAILSW) in the transmit clock check control register (XCLKCHK).
      iii. mute option (XCKFAIL) in the mute control register (AMUTE).
2. For the receive clock failure check:
   a. Configure receive clock failure detect logic (RMIN, RMAX, RPS) in the receive clock check control register (RCLKCHK).
   b. Clear receive clock failure flag (RCKFAIL) in the receive status register (RSTAT).
   c. Wait until first measurement is taken (> 32 AHCLKR clock periods).
   d. Verify no clock failure is detected.
   e. Repeat steps b–d until clock is running and is no longer issuing clock failure errors.
   f. After the receive clock is measured and falls within the acceptable range, the following may be enabled:
      i. receive clock failure interrupt enable bit (RCKFAIL) in the receiver interrupt control register (RINTCTL).
      ii. mute option (RCKFAIL) in the mute control register (AMUTE).

24.3.10.4.6.2 Transmit Clock Failure Check and Recovery

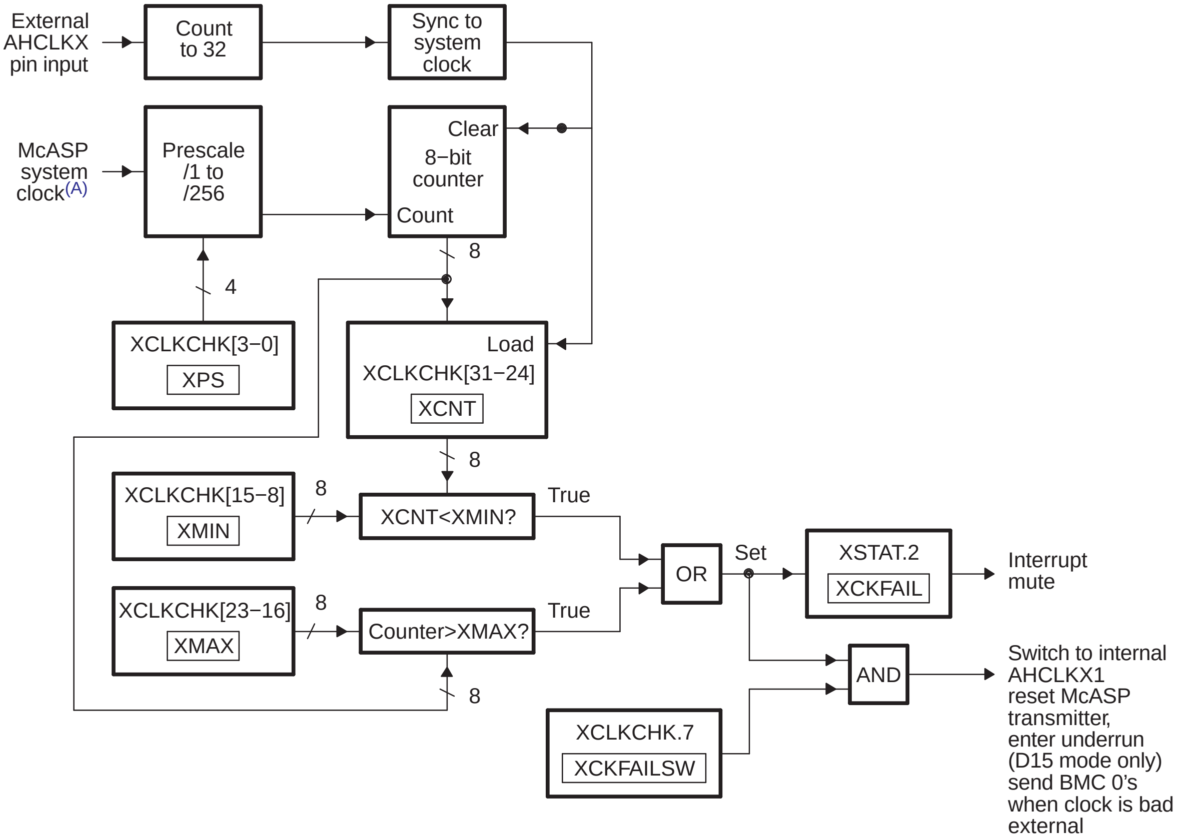
The transmit clock failure check circuit (Figure 24-31) works off both the internal McASP system clock and the external high-frequency serial clock (AHCLKX). It continually counts the number of system clocks for every 32 high rate serial clock (AHCLKX) periods, and stores the count in XCNT of the transmit clock check control register (XCLKCHK) every 32 high rate serial clock cycles.

The logic compares the count against a user-defined minimum allowable boundary (XMIN), and automatically flags an interrupt (XCKFAIL in XSTST) when an out-of-range condition occurs. An out-of-range minimum condition occurs when the count is smaller than XMIN. The logic continually compares the current count (from the running system clock counter) against the maximum allowable boundary (XMAX). This is in case the external clock completely stops, so that the counter value is not copied to XCNT. An out-of-range maximum condition occurs when the count is greater than XMAX. Note that the XMIN and XMAX fields are 8-bit unsigned values, and the comparison is performed using unsigned arithmetic.

An out-of-range count may indicate either that an unstable clock was detected, or that the audio source has changed and a new sample rate is being used.

In order for the transmit clock failure check circuit to operate correctly, the high-frequency serial clock divider must be taken out of reset regardless if AHCLKX is internally generated or externally sourced.

Figure 24-31. Transmit Clock Failure Detection Circuit Block Diagram



A Refer to device data manual for the McASP system clock source. This is not the same as AUXCLK.

The following actions are taken if a clock failure is detected:

1. Transmit clock failure flag (XCKFAIL) in XSTAT is set. This causes an interrupt if transmit clock failure interrupt enable bit (XCKFAIL) in XINTCTL is set.

In addition (only supported for DIT mode), if the transmit clock failure detect autoswitch enable bit (XCKFAILSW) in XCLKCHK is set, the following additional steps are taken to change the clock source from external to internal:

1. High-frequency transmit clock source bit (HCLKXM) in AHCLKXCTL is set to 1 and internal serial clock divider is selected. However, AHCLKX pin direction does not change to an output while XCKFAIL is set.
2. The internal clock divider is reset, so that the next clock it produces is a full period. However, the transmit clock divide ratio bits (HCLKXDIV) in AHCLKXCTL are not affected, so the internal clock divider generates clocks at the rate configured.
3. The transmit section is reset for a single serial clock period.
4. The transmit section is released from reset and attempts to begin transmitting. If data is available, it begins transmitting immediately; otherwise, it enters the underrun state. An initial underrun is certain to occur, the pattern 1100 (BMC zeroes) should be shifted out initially.

To change back to an external clock, take the following actions:

1. Wait for the external clock to stabilize again. This can be checked by polling the transmit clock count (XCNT) in XCLKCHK.
2. Reset the transmit section according to the startup procedure in Section 24.3.10.4.6.1.

24.3.10.4.6.3 Receive Clock Failure Check and Recovery

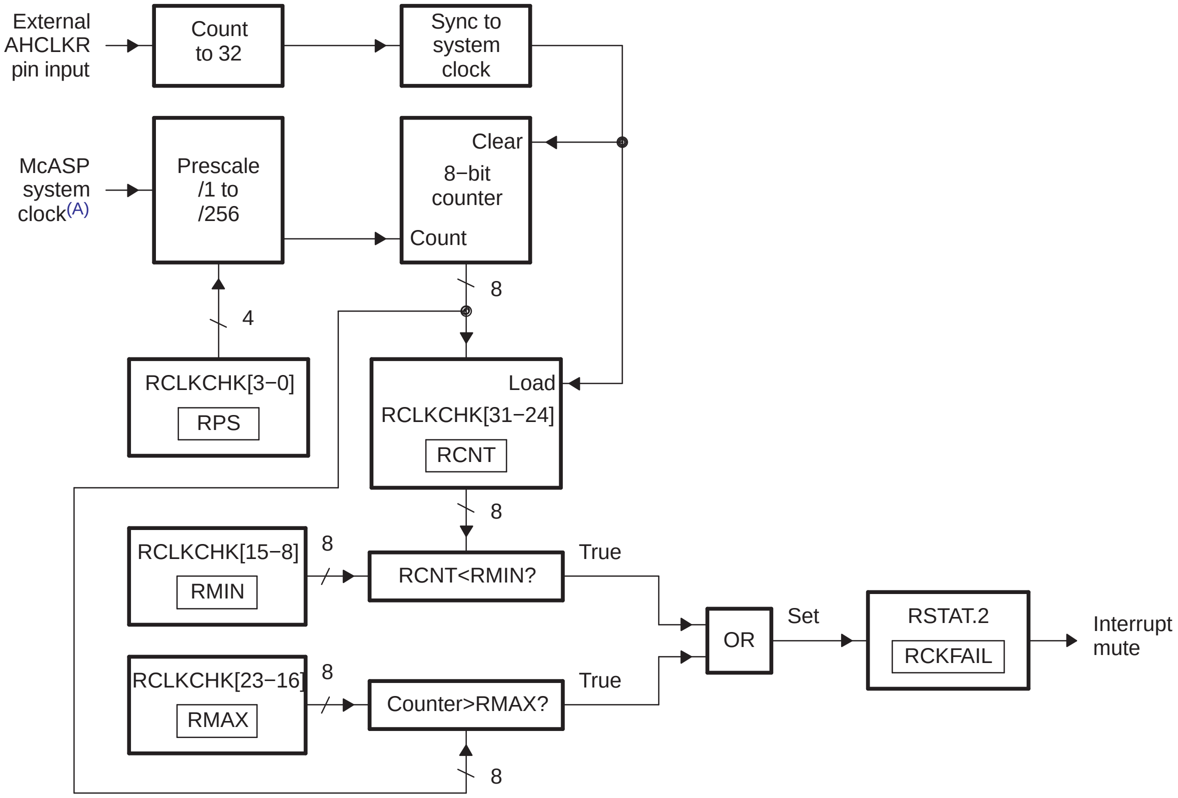
The receive clock failure check circuit (Figure 24-32) works off both the internal McASP system clock and the external high-frequency serial clock (AHCLKR). It continually counts the number of system clocks for every 32 high rate serial clock (AHCLKR) periods, and stores the count in RCNT of the receive clock check control register (RCLKCHK) every 32 high rate serial clock cycles.

The logic compares the count against a user-defined minimum allowable boundary (RMIN) and automatically flags an interrupt (RCKFAIL in RSTAT) when an out-of-range condition occurs. An out-of-range minimum condition occurs when the count is smaller than RMIN. The logic continually compares the current count (from the running system clock counter) against the maximum allowable boundary (RMAX). This is in case the external clock completely stops, so that the counter value is not copied to RCNT. An out-of-range maximum condition occurs when the count is greater than RMAX. Note that the RMIN and RMAX fields are 8-bit unsigned values, and the comparison is performed using unsigned arithmetic.

An out-of-range count may indicate either that an unstable clock was detected or that the audio source has changed and a new sample rate is being used.

In order for the receive clock failure check circuit to operate correctly, the high-frequency serial clock divider must be taken out of reset regardless if AHCLKR is internally generated or externally sourced.

Figure 24-32. Receive Clock Failure Detection Circuit Block Diagram



A Refer to device data manual for the McASP system clock source. This is not the same as AUXCLK.

24.3.10.5 Loopback Modes

The McASP features a digital loopback mode (DLB) that allows testing of the McASP code in TDM mode with a single processor device. In loopback mode, output of the transmit serializers is connected internally to the input of the receive serializers. Therefore, you can check the receive data against the transmit data to ensure that the McASP settings are correct. Digital loopback mode applies to TDM mode only (2 to 32 slots in a frame). It does not apply to DIT mode (XMOD = 180h) or burst mode (XMOD = 0).

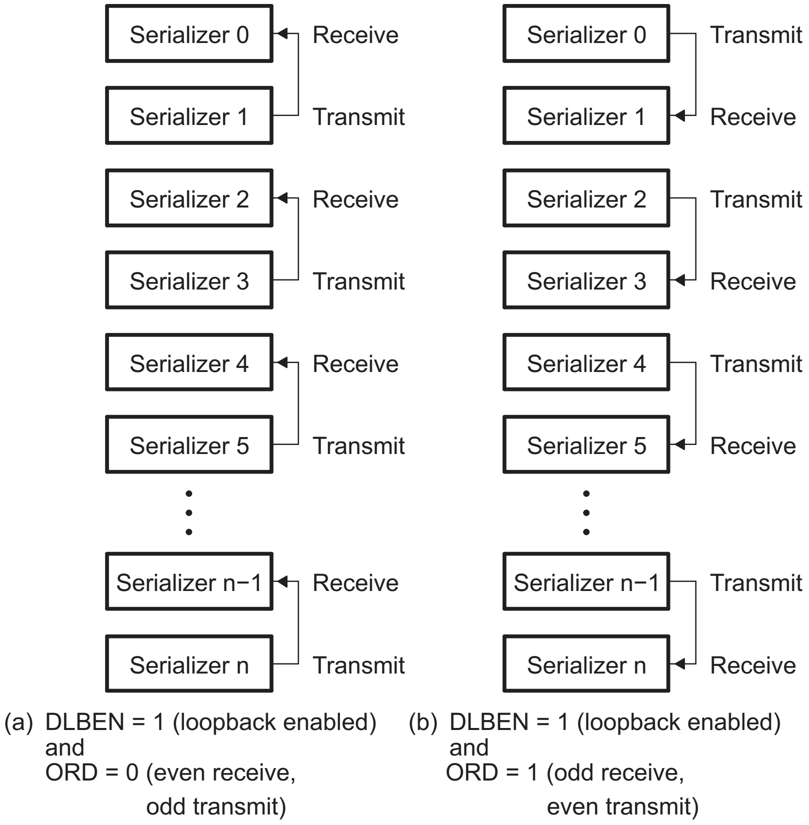
Figure 24-33 shows the basic logical connection of the serializers in loopback mode. Two types of loopback connections are possible, selected by the ORD bit in the digital loopback control register (DLBCTL) as follows:

- ORD = 0: Outputs of odd serializers are connected to inputs of even serializers. If this mode is selected, you should configure odd serializers to be transmitters and even serializers to be receivers.
- ORD = 1: Outputs of even serializers are connected to inputs of odd serializers. If this mode is selected, you should configure even serializers to be transmitters and odd serializers to be receivers.

Data can be externally visible at the I/O pin of the transmit serializer if the pin is configured as a McASP output pin by setting the corresponding PFUNC bit to 0 and PDIR bit to 1.

In loopback mode, the transmit clock and frame sync are used by both the transmit and receive sections of the McASP. The transmit and receive sections operate synchronously. This is achieved by setting the MODE bit of the DLBCTL register to 01b and the ASYNC bit of the ACLKXCTL register to 0.

Figure 24-33. Serializers in Loopback Mode



24.3.10.5.1 Loopback Mode Configurations

This is a summary of the settings required for digital loopback mode for TDM format:

- The DLBEN bit in DLBCTL must be set to 1 to enable loopback mode.
- The MODE bits in DLBCTL must be set to 01b for both the transmit and receive sections to use the transmit clock and frame sync generator.
- The ORD bit in DLBCTL must be programmed appropriately to select odd or even serializers to be transmitters or receivers. The corresponding serializers must be configured accordingly.
- The ASYNC bit in ACLKXCTL must be cleared to 0 to ensure synchronous transmit and receive operations.
- RMOD field in AFSRCTL and XMOD field in AFSXCTL must be set to 2h to 20h to indicate TDM mode. Loopback mode does not apply to DIT or burst mode.

24.3.11 Reset Considerations

The McASP has two reset sources: software reset and hardware reset.

24.3.11.1 Software Reset Considerations

The transmitter and receiver portions of the McASP may be put in reset through the global control register (GBLCTL). Note that a valid serial clock must be supplied to the desired portion of the McASP (transmit and/or receive) in order to assert the software reset bits in GBLCTL. see Section 24.3.12.2 for details on how to ensure reset has occurred.

The entire McASP module may also be reset through the Power and Sleep Controller (PSC). Note that from the McASP perspective, this reset appears as a hardware reset to the entire module.

24.3.11.2 Hardware Reset Considerations

When the McASP is reset due to device reset, the entire serial port (including the transmitter and receiver state machines, and other registers) is reset.

24.3.12 Setup and Initialization

This section discusses steps necessary to use the McASP module.

24.3.12.1 Considerations When Using a McASP

The following is a list of things to be considered for systems using a McASP:

24.3.12.1.1 Clocks

For each receive and transmit section:

- External or internal generated bit clock and high frequency clock?
- If internally generated, what is the bit clock speed and the high frequency clock speed?
- Clock polarity?
- External or internal generated frame sync?
- If internally generated, what is frame sync speed?
- Frame sync polarity?
- Frame sync width?
- Transmit and receive sync or asynchronous?

24.3.12.1.2 Data Pins

For each pin of each McASP:

- McASP or GPIO?
- Input or output?

24.3.12.1.3 Data Format

For each transmit and receive data:

- Internal numeric representation (integer, Q31 fraction)?
- I2S or DIT (transmit only)?
- Time slot delay (0, 1, or 2 bit)?
- Alignment (left or right)?
- Order (MSB first, LSB first)?
- Pad (if yes, pad with what value)?
- Slot size?
- Rotate?
- Mask?

24.3.12.1.4 Data Transfers

- Internal: DMA or CPU?
- External: TDM or burst?
- Bus: configuration bus (CFG) or data port (DAT)?

24.3.12.2 Transmit/Receive Section Initialization

You must follow the following steps to properly configure the McASP. If external clocks are used, they should be present prior to the following initialization steps.

1. Reset McASP to default values by setting GBLCTL = 0.
2. Configure all McASP registers except GBLCTL in the following order:
   a. Power Idle SYSCONFIG: PWRIDLESYSCONFIG.
   b. Receive registers: RMASK, RFMT, AFSRCTL, ACLKRCTL, AHCLKRCTL, RTDM, RINTCTL, RCLKCHK. If external clocks AHCLKR and/or ACLKR are used, they must be running already for proper synchronization of the GBLCTL register.
   c. Transmit registers: XMASK, XFMT, AFSXCTL, ACLKXCTL, AHCLKXCTL, XTDM, XINTCTL, XCLKCHK. If external clocks AHCLKX and/or ACLKX are used, they must be running already for proper synchronization of the GBLCTL register.
   d. Serializer registers: SRCTL[n].
   e. Global registers: Registers PFUNC, PDIR, DITCTL, DLBCTL, AMUTE. Note that PDIR should only be programmed after the clocks and frames are set up in the steps above. This is because the moment a clock pin is configured as an output in PDIR, the clock pin starts toggling at the rate defined in the corresponding clock control register. Therefore you must ensure that the clock control register is configured appropriately before you set the pin to be an output. A similar argument applies to the frame sync pins. Also note that the reset state for the transmit high-frequency clock divide register (HCLKXDIV) is divide-by-1, and the divide-by-1 clocks are not gated by the transmit high-frequency clock divider reset enable (XHCLKRST).
   f. DIT registers: For DIT mode operation, set up registers DITCSRA[n], DITCSRB[n], DITUDRA[n], and DITUDRB[n].
3. Start the respective high-frequency serial clocks AHCLKX and/or AHCLKR. This step is necessary even if external high-frequency serial clocks are used:
   a. Take the respective internal high-frequency serial clock divider(s) out of reset by setting the RHCLKRST bit for the receiver and/or the XHCLKRST bit for the transmitter in GBLCTL. All other bits in GBLCTL should be held at 0.
   b. Read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.

4. Start the respective serial clocks ACLKX and/or ACLKR. This step can be skipped if external serial clocks are used and they are running:
   a. Take the respective internal serial clock divider(s) out of reset by setting the RCLKRST bit for the receiver and/or the XCLKRST bit for the transmitter in GBLCTL. All other bits in GBLCTL should be left at the previous state.
   b. Read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.
5. Setup data acquisition as required:
   a. If DMA is used to service the McASP, set up data acquisition as desired and start the DMA in this step, before the McASP is taken out of reset.
   b. If CPU interrupt is used to service the McASP, enable the transmit and/ or receive interrupt as required.
   c. If CPU polling is used to service the McASP, no action is required in this step.
6. Activate serializers.
   a. Before starting, clear the respective transmitter and receiver status registers by writing XSTAT = FFFFh and RSTAT = FFFFh.
   b. Take the respective serializers out of reset by setting the RSRCLR bit for the receiver and/or the XSRCLR bit for the transmitter in GBLCTL. All other bits in GBLCTL should be left at the previous state.
   c. Read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.
7. Verify that all transmit buffers are serviced. Skip this step if the transmitter is not used. Also, skip this step if time slot 0 is selected as inactive (special cases, see Figure 24-21, second waveform). As soon as the transmit serializer is taken out of reset, XDATA in the XSTAT register is set, indicating that XBUF is empty and ready to be serviced. The XDATA status causes an DMA event AXEVT to be generated, and can cause an interrupt AXINT to be generated if it is enabled in the XINTCTL register.
   a. If DMA is used to service the McASP, the DMA automatically services the McASP upon receiving AXEVT. Before proceeding in this step, you should verify that the XDATA bit in the XSTAT is cleared to 0, indicating that all transmit buffers are already serviced by the DMA.
   b. If CPU interrupt is used to service the McASP, interrupt service routine is entered upon the AXINT interrupt. The interrupt service routine should service the XBUF registers. Before proceeding in this step, you should verify that the XDATA bit in XSTAT is cleared to 0, indicating that all transmit buffers are already serviced by the CPU.
   c. If CPU polling is used to service the McASP, the XBUF registers should be written to in this step.
8. Release state machines from reset.
   I. Take the respective state machine(s) out of reset by setting the RSMRST bit for the receiver and/or the XSMRST bit for the transmitter in GBLCTL. All other bits in GBLCTL should be left at the previous state.
   II. Read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.
9. Release frame sync generators from reset. Note that it is necessary to release the internal frame sync generators from reset, even if an external frame sync is being used, because the frame sync error detection logic is built into the frame sync generator.
   a. Take the respective frame sync generator(s) out of reset by setting the RFRST bit for the receiver, and/or the XFRST bit for the transmitter in GBLCTL. All other bits in GBLCTL should be left at the previous state.
   b. Read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.
10. Upon the first frame sync signal, McASP transfers begin. The McASP synchronizes to an edge on the frame sync pin, not the level on the frame sync pin. This makes it easy to release the state machine and frame sync generators from reset.
   a. For example, if you configure the McASP for a rising edge transmit frame sync, then you do not need to wait for a low level on the frame sync pin before releasing the McASP transmitter state

machine and frame sync generators from reset.

24.3.12.3 Separate Transmit and Receive Initialization

In many cases, it is desirable to separately initialize the McASP transmitter and receiver. For example, you may delay the initialization of the transmitter until the type of data coming in on the receiver is recognized. Or a change in the incoming data stream on the receiver may necessitate a reinitialization of the transmitter.

In this case, you may still follow the sequence outlined in Section 24.3.12.2, but use it for each section (transmit, receive) individually. The GBLCTL register is aliased to RGBLCTL and XGBLCTL to facilitate separate initialization of transmit and receive sections.

24.3.12.4 Importance of Reading Back GBLCTL

In Section 24.3.12.2, steps 3b, 4b, 6c, 8b, and 9b state that GBLCTL should be read back until the bits that were written are successfully latched. This is important, because the transmitter and receiver state machines run off of the respective bit clocks, which are typically about tens to hundreds of times slower than the processor's internal bus clock. Therefore, it takes many cycles between when the processor writes to GBLCTL (or RGBLCTL and XGBLCTL), and when the McASP actually recognizes the write operation. If you skip this step, then the McASP may never see the reset bits in the global control registers get asserted and de-asserted; resulting in an uninitialized McASP.

Therefore, the logic in McASP has been implemented such that once the processor writes GBLCTL, RGBLCTL, or XGBLCTL, the resulting write is not visible by reading back GBLCTL until the McASP has recognized the change. This typically requires two bit clocks plus two processor bus clocks to occur.

Also, if the bit clocks can be completely stopped, any software that polls GBLCTL should be implemented with a time-out. If GBLCTL does not have a time-out, and the bit clock stops, the changes written to GBLCTL will not be reflected until the bit clock restarts.

Finally, please note that while RGBLCTL and XGBLCTL allow separate changing of the receive and transmit halves of GBLCTL, they also immediately reflect the updated value (useful for debug purposes). Only GBLCTL can be used for the read back step.

24.3.12.5 Synchronous Transmit and Receive Operation (ASYNC = 0)

When ASYNC = 0 in ACLKXCTL, the transmit and receive sections operate synchronously from the transmit section clock and transmit frame sync signals (Figure 24-17). The receive section may have a different (but compatible in terms of slot size) data format.

When ASYNC = 0, the receive frame sync generator is internally disabled. If the AFSX pin is configured as an output, it serves as the frame sync signal for both transmit and receive. The AFSR pin should not be used because the transmit frame sync generator output, which is used by both the transmitter and the receiver when ASYNC = 0, is not propagated to the AFSR pin (Figure 24-19).

When ASYNC = 0, the transmit and receive sections must share some common settings, since they both use the same clock and frame sync signals:

- DITEN = 0 in DITCTL (TDM mode is enabled).
- The total number of bits per frame must be the same (that is, RSSZ × RMOD must equal XSSZ × XMOD).
- Both transmit and receive should either be specified as burst or TDM mode, but not mixed.
- The settings in ACLKRCTL are irrelevant.
- FSXM must match FSRM.
- FXWID must match FRWID.

For all other settings, the transmit and receive sections may be programmed independently.

24.3.12.6 Asynchronous Transmit and Receive Operation (ASYNC = 1)

When ASYNC = 1 in ACLKXCTL, the transmit and receive sections operate completely independently and have separate clock and frame sync signals (Figure 24-17, Figure 24-18, and Figure 24-19). The events generated by each section come asynchronously.

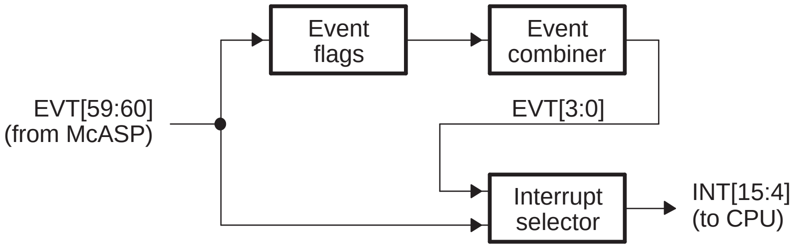
24.3.13 Interrupts

24.3.13.1 Interrupt Multiplexing

The processor includes an interrupt controller (INTC) to manage CPU interrupts. The INTC maps the device events to 12 CPU interrupts. The McASP generates 4 interrupts to the processor.

The event inputs can be routed to 12 maskable interrupts to the CPU (INT[15:4]). The INTC interrupt selector allows the McASP system events to be routed to any of the 12 CPU interrupt inputs. Furthermore, the INTC provides status flags and allows the combination of events, as shown in Figure 24-34. You must properly configure the INTC by enabling, masking, and routing the McASP system events to the desired CPU interrupts.

Figure 24-34. Interrupt Multiplexing



24.3.13.2 Transmit Data Ready Interrupt

The transmit data ready interrupt (XDATA) is generated if XDATA is 1 in the XSTAT register and XDATA is also enabled in XINTCTL. Section 24.3.10.1.1 provides details on when XDATA is set in the XSTAT register.

A transmit start of frame interrupt (XSTAFRM) is triggered by the recognition of transmit frame sync. A transmit last slot interrupt (XLAST) is a qualified version of the data ready interrupt (XDATA). It has the same behavior as the data ready interrupt, but is further qualified by having the data requested belonging to the last slot (the slot that just ended was next-to-last TDM slot, current slot is last slot).

24.3.13.3 Receive Data Ready Interrupt

The receive data ready interrupt (RDATA) is generated if RDATA is 1 in the RSTAT register and RDATA is also enabled in RINTCTL. Section 24.3.10.1.2 provides details on when RDATA is set in the RSTAT register.

A receiver start of frame interrupt (RSTAFRM) is triggered by the recognition of a receiver frame sync. A receiver last slot interrupt (RLAST) is a qualified version of the data ready interrupt (RDATA). It has the same behavior as the data ready interrupt, but is further qualified by having the data in the buffer come from the last TDM time slot (the slot that just ended was last TDM slot).

24.3.13.4 Error Interrupts

Upon detection, the following error conditions generate interrupt flags:

- In the receive status register (RSTAT):
  - Receiver overrun (ROVRN).
  - Unexpected receive frame sync (RSYNCERR).
  - Receive clock failure (RCKFAIL).
  - Receive DMA error (RDMAERR).

- In the transmit status register (XSTAT):
  - Transmit underrun (XUNDRN).
  - Unexpected transmit frame sync (XSYNCERR).
  - Transmit clock failure (XCKFAIL).
  - Transmit DMA error (XDMAERR).

Each interrupt source also has a corresponding enable bit in the receive interrupt control register (RINTCTL) and transmit interrupt control register (XINTCTL). If the enable bit is set in RINTCTL or XINTCTL, an interrupt is requested when the interrupt flag is set in RSTAT or XSTAT. If the enable bit is not set, no interrupt request is generated. However, the interrupt flag may be polled.

24.3.13.5 Audio Mute (AMUTE) Function

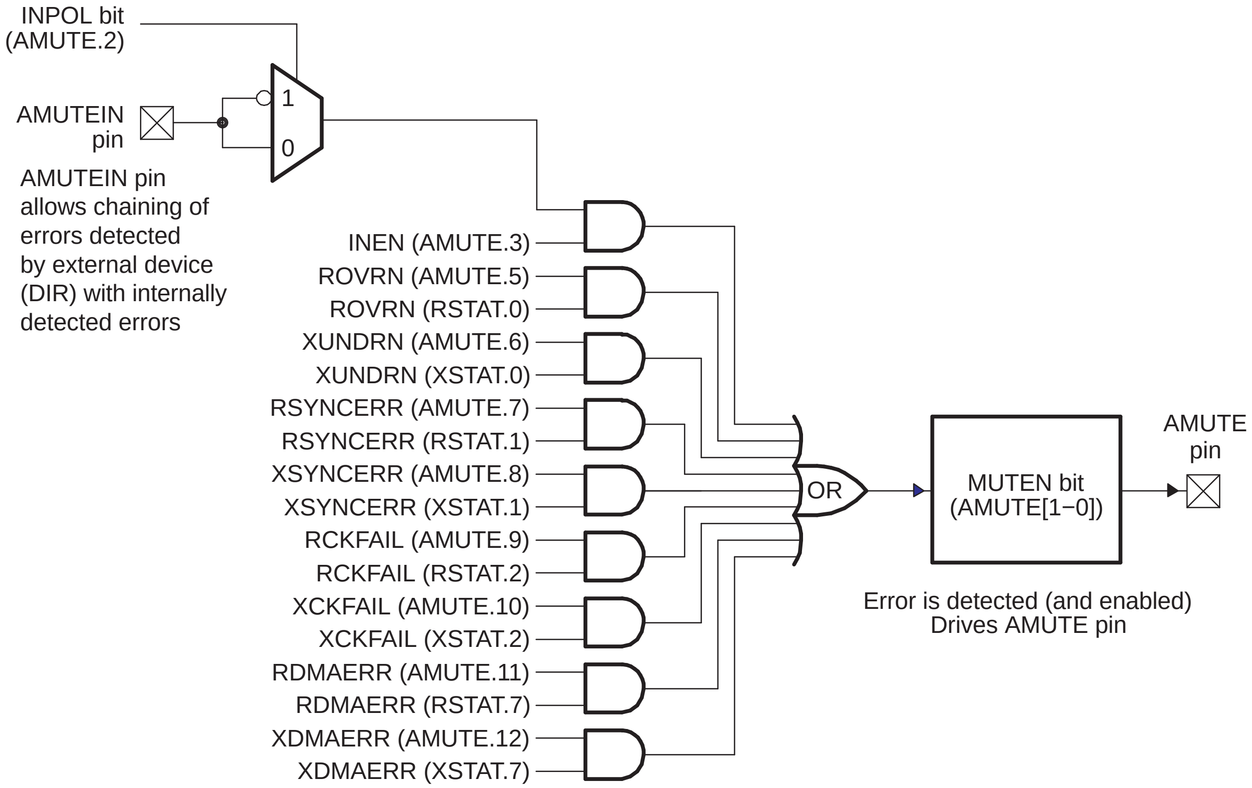
The McASP includes an automatic audio mute function (Figure 24-35) that asserts in hardware the AMUTE pin to a preprogrammed output state, as selected by the MUTEN bit in the audio mute control register (AMUTE). The AMUTE pin is asserted when one of the interrupt flags is set or an external device issues an error signal on the AMUTEIN input. Typically, the AMUTEIN input is shared with a device interrupt pin (for example EXT_INT4).

The AMUTEIN input allows the on-chip logic to consider a mute input from other devices in the system, so that all errors may be considered. The AMUTEIN input has a programmable polarity to allow it to adapt to different devices, as selected by the INPOL bit in AMUTE, and it must be enabled explicitly.

In addition to the external AMUTEIN input, the AMUTE pin output may be asserted when one of the error interrupt flags is set and its mute function is enabled in AMUTE.

When one or more of the errors is detected and enabled, the AMUTE pin is driven to an active state that is selected by MUTEN in AMUTE. The active polarity of the AMUTE pin is programmable by MUTEN (and the inactive polarity is the opposite of the active polarity). The AMUTE pin remains driven active until software clears all the error interrupt flags that are enabled to mute, and until the AMUTEIN is inactive.

Figure 24-35. Audio Mute (AMUTE) Block Diagram



24.3.13.6 Multiple Interrupts

This only applies to interrupts and not to DMA requests. The following terms are defined:

- **Active Interrupt Request:** a flag in RSTAT or XSTAT is set and the interrupt is enabled in RINTCTL or XINTCTL.
- **Outstanding Interrupt Request:** An interrupt request has been issued on one of the McASP transmit/receive interrupt ports, but that request has not yet been serviced.
- **Serviced:** The CPU writes to RSTAT or XSTAT to clear one or more of the active interrupt request flags.

The first interrupt request to become active for the transmitter with the interrupt flag set in XSTAT and the interrupt enabled in XINTCTL generates a request on the McASP transmit interrupt port AXINT.

If more than one interrupt request becomes active in the same cycle, a single interrupt request is generated on the McASP transmit interrupt port. Subsequent interrupt requests that become active while the first interrupt request is outstanding do not immediately generate a new request pulse on the McASP transmit interrupt port.

The transmit interrupt is serviced with the CPU writing to XSTAT. If any interrupt requests are active after the write, a new request is generated on the McASP transmit interrupt port.

The receiver operates in a similar way, but using RSTAT, RINTCTL, and the McASP receive interrupt port ARINT.

One outstanding interrupt request is allowed on each port, so a transmit and a receive interrupt request may both be outstanding at the same time.

24.3.14 EDMA Event Support

24.3.14.1 EDMA Events

There are 6 EDMA events.

24.3.14.2 Using the DMA for McASP Servicing

The most typical scenario is to use the DMA to service the McASP through the data port, although the DMA can also service the McASP through the configuration bus. Two possibilities exist for using the DMA events to service the McASP:

1. Use **AXEVT/AREVT**: Triggered upon each XDATA/RDATA transition from 0 to 1.
2. Use **AXEVTO/AREVTO** and **AXEVTE/AREVTE**: Alternating AXEVT/AREVT events for odd/even slots. Upon AXEVT/AREVT, AXEVTO/AREVTO is triggered if the event is for an odd channel, and AXEVTE/AREVTE is triggered if the event is for an even channel.

NOTE: Check the device-specific data manual to see if AXEVTO/AREVTO and AXEVTE/AREVTE are supported. These are optional.

Figure 24-36 and Figure 24-37 show an example audio system with six audio channels (LF, RF, LS, RS, C, and LFE) transmitted from three AXRn pins on the McASP. Figure 24-36 and Figure 24-37 show when events AXEVT, AXEVTO, and AXEVTE are triggered. Figure 24-36 and Figure 24-37 also apply for the receive audio channels and show when events AREVT, AREVTO, and AREVTE are triggered.

You can either use the DMA to service the McASP upon events AXEVT and AREVT (Figure 24-36) or upon events AXEVTO, AREVTO, AXEVTE, and AREVTE (Figure 24-37).

Figure 24-36. DMA Events in an Audio Example–Two Events (Scenario 1)

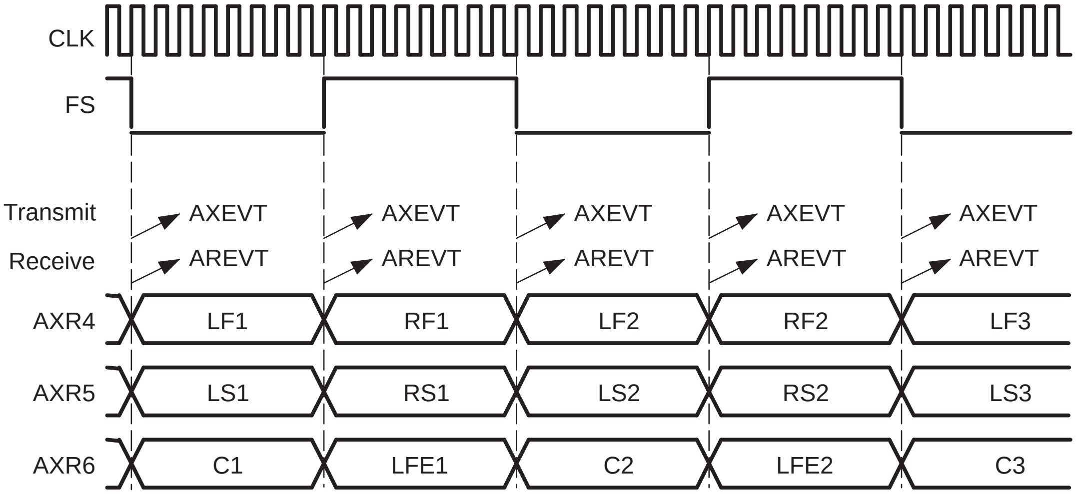
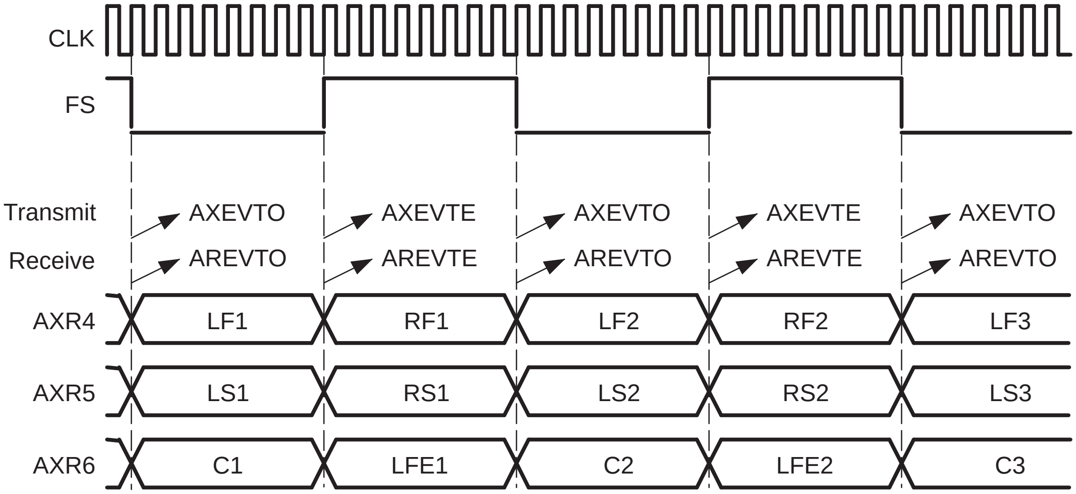


Figure 24-37. DMA Events in an Audio Example–Four Events (Scenario 2)



In scenario 1 (Figure 24-36), a DMA event AXEVT/AREVT is triggered on each time slot. In the example, AXEVT is triggered for each of the transmit audio channel time slot (Time slot for channels LF, LS, and C; and time slot for channels RF, RS, LFE). Similarly, AREVT is triggered for each of the receive audio channel time slot. Scenario 1 allows for the use of a single DMA to transmit all audio channels, and a single DMA to receive all audio channels.

In scenario 2 (Figure 24-37), two alternating DMA events are triggered for each time slot. In the example, AXEVTE (even) is triggered for the time slot for the even audio channels (LF, LS, C) and AXEVTO (odd) is triggered for the time slot for the odd audio channels (RF, RS, LFE). AXEVTO and AXEVTE alternate in time. The same is true in the receive direction with the use of AREVTO and AREVTE. This scenario allows for the use of two DMA channels (odd and even) to transmit all audio channels, and two DMA channels to receive all audio channels.

Here are some guidelines on using the different DMA events:

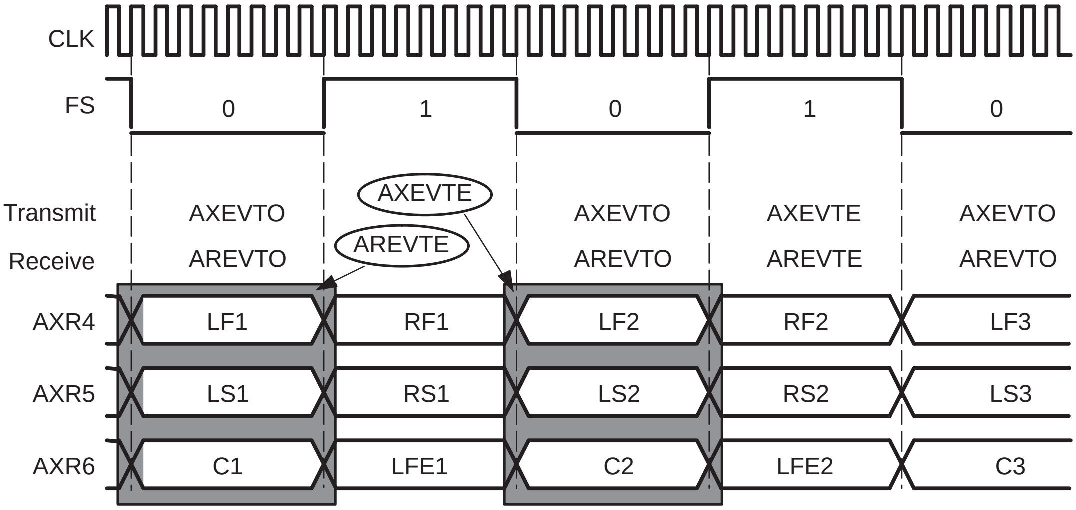
- Either use AXEVT, or the combination of AXEVTO and AXEVTE, to service the McASP. Never use all three at the same time. Similarly for receive, either use AREVT, or the combination of AREVTO and AREVTE.
- The McASP generates transmit DMA events independently from receive DMA events; therefore, separate schemes can be used for transmit and receive DMA. For example, scenario 1 could be used for the transmit data (AXEVT) and scenario 2 could be used for the receive data (AREVTO, AREVTE), and conversely.

Note the difference between DMA event generation and the CPU interrupt generation. DMA events are generated automatically upon data ready; whereas CPU interrupt generation needs to be enabled in the XINTCTL/RINTCTL register.

In Figure 24-37, scenario 2, each transmit DMA request is for data in the next time slot, while each receive DMA request is for data in the previous time slot. For example, Figure 24-38 shows a circled AXEVTE event for an even time slot transmit DMA request. The transmitter always requests a DMA transfer for data it will need to transmit during the next time slot. So, in this example, the circled event AXEVTE is a request for data for samples LF2, LS2, and C2.

On the other hand, the circled AREVTE event is an even time slot receive DMA request. The receiver always requests a DMA transfer for data it received during the previous time slot. In this example, the circled event AREVTE is a request for samples LF1, LS1, and C1.

Figure 24-38. DMA Events in an Audio Example



24.3.15 Power Management

The McASP can be placed in reduced power modes to conserve power during periods of low activity.

24.3.16 Emulation Considerations

NOTE: The receive buffer registers (RBUF*n*) and transmit buffer registers (XBUF*n*) should not be accessed by the emulator when the McASP is running. Such an access will cause the RRDY/XRDY bit in the serializer control register *n* (SRCTL*n*) to be updated.

The McASP does not support the emulation suspend.

24.4 McASP Registers

24.4.1 MCASP Registers

Table 24-11 lists the memory-mapped registers for the MCASP. All register offset addresses not listed in Table 24-11 should be considered as reserved locations and the register contents should not be modified.

Table 24-11. MCASP REGISTERS

Offset	Acronym	Register Name	Section
0h	MCASP_REV	Revision Identification Register	Section 24.4.1.1
4h	MCASP_PWRIDLESYSCONFIG	Power Idle SYSCONFIG Register	Section 24.4.1.2
10h	MCASP_PFUNC	Pin Function Register	Section 24.4.1.3
14h	MCASP_PDIR	Pin Direction Register	Section 24.4.1.4
18h	MCASP_PDOUT	Pin Data Output Register	Section 24.4.1.5
1Ch	MCASP_PDIN	Pin Data Input Register	Section 24.4.1.6
20h	MCASP_PDCLR	Pin Data Clear Register	Section 24.4.1.7
44h	MCASP_GBLCTL	Global Control Register	Section 24.4.1.8
48h	MCASP_AMUTE	Audio Mute Control Register	Section 24.4.1.9
4Ch	MCASP_DLBCTL	Digital Loopback Control Register	Section 24.4.1.10
50h	MCASP_DITCTL	DIT Mode Control Register	Section 24.4.1.11
60h	MCASP_RGBLCTL	Receiver Global Control Register	Section 24.4.1.12
64h	MCASP_RMASK	Receive Format Unit Bit Mask Register	Section 24.4.1.13
68h	MCASP_RFMT	Receive Bit Stream Format Register	Section 24.4.1.14
6Ch	MCASP_AFSRCTL	Receive Frame Sync Control Register	Section 24.4.1.15
70h	MCASP_ACLKRCTL	Receive Clock Control Register	Section 24.4.1.16
74h	MCASP_AHCLKRCTL	Receive High-Frequency Clock Control Register	Section 24.4.1.17
78h	MCASP_RTDM	Receive TDM Time Slot 0-31 Register	Section 24.4.1.18
7Ch	MCASP_RINTCTL	Receiver Interrupt Control Register	Section 24.4.1.19
80h	MCASP_RSTAT	Receiver Status Register	Section 24.4.1.20
84h	MCASP_RSLOT	Current Receive TDM Time Slot Register	Section 24.4.1.21
88h	MCASP_RCLKCHK	Receive Clock Check Control Register	Section 24.4.1.22
8Ch	MCASP_REVTCTL	Receiver DMA Event Control Register	Section 24.4.1.23
A0h	MCASP_XGBLCTL	Transmitter Global Control Register	Section 24.4.1.24
A4h	MCASP_XMASK	Transmit Format Unit Bit Mask Register	Section 24.4.1.25
A8h	MCASP_XFMT	Transmit Bit Stream Format Register	Section 24.4.1.26
ACh	MCASP_AFSXCTL	Transmit Frame Sync Control Register	Section 24.4.1.27
B0h	MCASP_ACLKXCTL	Transmit Clock Control Register	Section 24.4.1.28
B4h	MCASP_AHCLKXCTL	Transmit High-Frequency Clock Control Register	Section 24.4.1.29
B8h	MCASP_XTDM	Transmit TDM Time Slot 0-31 Register	Section 24.4.1.30
BCh	MCASP_XINTCTL	Transmitter Interrupt Control Register	Section 24.4.1.31
C0h	MCASP_XSTAT	Transmitter Status Register	Section 24.4.1.32
C4h	MCASP_XSLOT	Current Transmit TDM Time Slot Register	Section 24.4.1.33
C8h	MCASP_XCLKCHK	Transmit Clock Check Control Register	Section 24.4.1.34
CCh	MCASP_XEVTCTL	Transmitter DMA Event Control Register	Section 24.4.1.35
100h to 114h	MCASP_DITCSRA0 to MCASP_DITCSRA5	Left (Even TDM Time Slot) Channel Status Registers (DIT Mode)	Section 24.4.1.36
118h to 12Ch	MCASP_DITCSRB0 to MCASP_DITCSRB5	Right (Odd TDM Time Slot) Channel Status Registers (DIT Mode)	Section 24.4.1.37
130h to 144h	MCASP_DITUDRA0 to MCASP_DITUDRA5	Left (Even TDM Time Slot) Channel User Data Registers (DIT Mode)	Section 24.4.1.38
148h to 15Ch	MCASP_DITUDRB0 to MCASP_DITUDRB5	Right (Odd TDM Time Slot) Channel User Data Registers (DIT Mode)	Section 24.4.1.39

Table 24-11. MCASP REGISTERS (continued)

Offset	Acronym	Register Name	Section
180h to 194h	MCASP_SRCTL0 to MCASP_SRCTL5	Serializer Control Registers	Section 24.4.1.40
200h to 214h	MCASP_XBUF0 to MCASP_XBUF5	Transmit Buffer Register for Serializers	Section 24.4.1.41
280h to 294h	MCASP_RBUF0 to MCASP_RBUF5	Receive Buffer Register for Serializers	Section 24.4.1.42
1000h	MCASP_WFIFOCTL	Write FIFO Control Register	Section 24.4.1.43
1004h	MCASP_WFIFOSTS	Write FIFO Status Register	Section 24.4.1.44
1008h	MCASP_RFIFOCTL	Read FIFO Control Register	Section 24.4.1.45
100Ch	MCASP_RFIFOSTS	Read FIFO Status Register	Section 24.4.1.46

24.4.1.1 MCASP_REV Register (offset = 0h) [reset = 44300A02h]

MCASP_REV is shown in Figure 24-39 and described in Table 24-12.

The revision identification register (REV) contains identification data for the peripheral.

Figure 24-39. MCASP_REV Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">REV</td></tr>
<tr><td colspan="32">R-44300A02h</td></tr>
</table>

Table 24-12. MCASP_REV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REV	R	44300A02h	Identifies revision of peripheral.

24.4.1.2 MCASP_PWRIDLESYSCONFIG Register (offset = 4h) [reset = 2h]

MCASP_PWRIDLESYSCONFIG is shown in Figure 24-40 and described in Table 24-13.

Figure 24-40. MCASP_PWRIDLESYSCONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED		OTHER				IDLEMODE	
R-0h		R/W-0h				R/W-2h	

Table 24-13. MCASP_PWRIDLESYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5-2	OTHER	R/W	0h	Reserved for future programming.
1-0	IDLEMODE	R/W	2h	Power management Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state. 0h = Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements. Backup mode, for debug only. 1h = No-idle mode: local target never enters idle state. Backup mode, for debug only. 2h = Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wakeup events. 3h = Reserved.

24.4.1.3 MCASP_PFUNC Register (offset = 10h) [reset = 0h]

MCASP_PFUNC is shown in Figure 24-41 and described in Table 24-14.

The pin function register (PFUNC) specifies the function of AXRn, ACLKX, AHCLKX, AFSX, ACLKR, AHCLKR, and AFSR pins as either a McASP pin or a general-purpose input/output (GPIO) pin. CAUTION: Writing a value other than 0 to reserved bits in this register may cause improper device operation.

Figure 24-41. MCASP_PFUNC Register

31	30	29	28	27	26	25	24
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	AMUTE	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED				AXR			
R-0h				R/W-0h			

Table 24-14. MCASP_PFUNC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AFSR	R/W	0h	Determines if AFSR pin functions as McASP or GPIO. 0h = Pin functions as McASP pin. 1h = Pin functions as GPIO pin.
30	AHCLKR	R/W	0h	Determines if AHCLKR pin functions as McASP or GPIO. 0h = Pin functions as McASP pin. 1h = Pin functions as GPIO pin.
29	ACLKR	R/W	0h	Determines if ACLKR pin functions as McASP or GPIO. 0h = Pin functions as McASP pin. 1h = Pin functions as GPIO pin.
28	AFSX	R/W	0h	Determines if AFSX pin functions as McASP or GPIO. 0h = Pin functions as McASP pin. 1h = Pin functions as GPIO pin.
27	AHCLKX	R/W	0h	Determines if AHCLKX pin functions as McASP or GPIO. 0h = Pin functions as McASP pin. 1h = Pin functions as GPIO pin.
26	ACLKX	R/W	0h	Determines if ACLKX pin functions as McASP or GPIO. 0h = Pin functions as McASP pin. 1h = Pin functions as GPIO pin.
25	AMUTE	R/W	0h	Determines if AMUTE pin functions as McASP or GPIO. 0h = Pin functions as McASP pin. 1h = Pin functions as GPIO pin.
24-4	RESERVED	R	0h	
3-0	AXR	R/W	0h	Determines if AXRn pin functions as McASP or GPIO. 0h = Pin functions as McASP pin. 1h = Pin functions as GPIO pin.

24.4.1.4 MCASP_PDIR Register (offset = 14h) [reset = 0h]

MCASP_PDIR is shown in Figure 24-42 and described in Table 24-15.

The pin direction register (PDIR) specifies the direction of AXRn, ACLKX, AHCLKX, AFSX, ACLKR, AHCLKR, and AFSR pins as either an input or an output pin. Regardless of the pin function register (PFUNC) setting, each PDIR bit must be set to 1 for the specified pin to be enabled as an output and each PDIR bit must be cleared to 0 for the specified pin to be an input. For example, if the McASP is configured to use an internally-generated bit clock and the clock is to be driven out to the system, the PFUNC bit must be cleared to 0 (McASP function) and the PDIR bit must be set to 1 (an output). When AXRn is configured to transmit, the PFUNC bit must be cleared to 0 (McASP function) and the PDIR bit must be set to 1 (an output). Similarly, when AXRn is configured to receive, the PFUNC bit must be cleared to 0 (McASP function) and the PDIR bit must be cleared to 0 (an input). CAUTION: Writing a value other than 0 to reserved bits in this register may cause improper device operation.

Figure 24-42. MCASP_PDIR Register

31	30	29	28	27	26	25	24
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	AMUTE	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED				AXR			
R-0h				R/W-0h			

Table 24-15. MCASP_PDIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AFSR	R/W	0h	Determines if AFSR pin functions as an input or output. 0h = Pin functions as input. 1h = Pin functions as output.
30	AHCLKR	R/W	0h	Determines if AHCLKR pin functions as an input or output. 0h = Pin functions as input. 1h = Pin functions as output.
29	ACLKR	R/W	0h	Determines if ACLKR pin functions as an input or output. 0h = Pin functions as input. 1h = Pin functions as output.
28	AFSX	R/W	0h	Determines if AFSX pin functions as an input or output. 0h = Pin functions as input. 1h = Pin functions as output.
27	AHCLKX	R/W	0h	Determines if AHCLKX pin functions as an input or output. 0h = Pin functions as input. 1h = Pin functions as output.
26	ACLKX	R/W	0h	Determines if ACLKX pin functions as an input or output. 0h = Pin functions as input. 1h = Pin functions as output.
25	AMUTE	R/W	0h	Determines if AMUTE pin functions as an input or output. 0h = Pin functions as input. 1h = Pin functions as output.
24-4	RESERVED	R	0h	

Table 24-15. MCASP_PDIR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	AXR	R/W	0h	Determines if AXRn pin functions as an input or output. 0h = Pin functions as input. 1h = Pin functions as output.

24.4.1.5 MCASP_PDOUT Register (offset = 18h) [reset = 0h]

MCASP_PDOUT is shown in Figure 24-43 and described in Table 24-16.

The pin data output register (PDOUT) holds a value for data out at all times, and may be read back at all times. The value held by PDOUT is not affected by writing to PDIR and PFUNC. However, the data value in PDOUT is driven out onto the McASP pin only if the corresponding bit in PFUNC is set to 1 (GPIO function) and the corresponding bit in PDIR is set to 1 (output). When reading data, returns the corresponding bit value in PDOUT[n], does not return input from I/O pin; when writing data, writes to the corresponding PDOUT[n] bit. PDOUT has these aliases or alternate addresses: PDSET When written to at this address, writing a 1 to a bit in PDSET sets the corresponding bit in PDOUT to 1; writing a 0 has no effect and keeps the bits in PDOUT unchanged. PDCLR When written to at this address, writing a 1 to a bit in PDCLR clears the corresponding bit in PDOUT to 0; writing a 0 has no effect and keeps the bits in PDOUT unchanged. There is only one set of data out bits, PDOUT[31-0]. The other registers, PDSET and PDCLR, are just different addresses for the same control bits, with different behaviors during writes. CAUTION: Writing a value other than 0 to reserved bits in this register may cause improper device operation.

Figure 24-43. MCASP_PDOUT Register

31	30	29	28	27	26	25	24
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	AMUTE	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED				AXR			
R-0h				R/W-0h			

Table 24-16. MCASP_PDOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AFSR	R/W	0h	Determines drive on AFSR output pin when the corresponding PFUNC[31] and PDIR[31] bits are set to 1. 0h = Pin drives low. 1h = Pin drives high.
30	AHCLKR	R/W	0h	Determines drive on AHCLKR output pin when the corresponding PFUNC[30] and PDIR[30] bits are set to 1. 0h = Pin drives low. 1h = Pin drives high.
29	ACLKR	R/W	0h	Determines drive on ACLKR output pin when the corresponding PFUNC[29] and PDIR[29] bits are set to 1. 0h = Pin drives low. 1h = Pin drives high.
28	AFSX	R/W	0h	Determines drive on AFSX output pin when the corresponding PFUNC[28] and PDIR[28] bits are set to 1. 0h = Pin drives low. 1h = Pin drives high.
27	AHCLKX	R/W	0h	Determines drive on AHCLKX output pin when the corresponding PFUNC[27] and PDIR[27] bits are set to 1. 0h = Pin drives low. 1h = Pin drives high.

Table 24-16. MCASP_PDOUT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
26	ACLKX	R/W	0h	Determines drive on ACLKX output pin when the corresponding PFUNC[26] and PDIR[26] bits are set to 1. 0h = Pin drives low. 1h = Pin drives high.
25	AMUTE	R/W	0h	Determines drive on AMUTE output pin when the corresponding PFUNC[25] and PDIR[25] bits are set to 1. 0h = Pin drives low. 1h = Pin drives high.
24-4	RESERVED	R	0h	
3-0	AXR	R/W	0h	Determines drive on AXR[n] output pin when the corresponding PFUNC[n] and PDIR[n] bits are set to 1. 0h = Pin drives low. 1h = Pin drives high.

24.4.1.6 MCASP_PDIN Register (offset = 1Ch) [reset = 0h]

MCASP_PDIN is shown in Figure 24-44 and described in Table 24-17.

The pin data input register (PDIN) holds the I/O pin state of each of the McASP pins. PDIN allows the actual value of the pin to be read, regardless of the state of PFUNC and PDIR. The value after reset for registers 1 through 15 and 24 through 31 depends on how the pins are being driven. CAUTION: Writing a value other than 0 to reserved bits in this register may cause improper device operation.

Figure 24-44. MCASP_PDIN Register

31	30	29	28	27	26	25	24
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	AMUTE	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED				AXR			
R-0h				R/W-0h			

Table 24-17. MCASP_PDIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AFSR	R/W	0h	Logic level on AFSR pin. 0h = Pin is logic low. 1h = Pin is logic high.
30	AHCLKR	R/W	0h	Logic level on AHCLKR pin. 0h = Pin is logic low. 1h = Pin is logic high.
29	ACLKR	R/W	0h	Logic level on ACLKR pin. 0h = Pin is logic low. 1h = Pin is logic high.
28	AFSX	R/W	0h	Logic level on AFSX pin. 0h = Pin is logic low. 1h = Pin is logic high.
27	AHCLKX	R/W	0h	Logic level on AHCLKX pin. 0h = Pin is logic low. 1h = Pin is logic high.
26	ACLKX	R/W	0h	Logic level on ACLKX pin. 0h = Pin is logic low. 1h = Pin is logic high.
25	AMUTE	R/W	0h	Logic level on AMUTE pin. 0h = Pin is logic low. 1h = Pin is logic high.
24-4	RESERVED	R	0h	
3-0	AXR	R/W	0h	Logic level on AXR[n] pin. 0h = Pin is logic low. 1h = Pin is logic high.

24.4.1.7 MCASP_PDCLR Register (offset = 20h) [reset = 0h]

MCASP_PDCLR is shown in Figure 24-45 and described in Table 24-18.

The pin data clear register (PDCLR) is an alias of the pin data output register (PDOUT) for writes only. Writing a 1 to the PDCLR bit clears the corresponding bit in PDOUT and, if PFUNC = 1 (GPIO function) and PDIR = 1 (output), drives a logic low on the pin. PDCLR is useful for a multitasking system because it allows you to clear to a logic low only the desired pin(s) within a system without affecting other I/O pins controlled by the same McASP. CAUTION: Writing a value other than 0 to reserved bits in this register may cause improper device operation.

Figure 24-45. MCASP_PDCLR Register

31	30	29	28	27	26	25	24
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	AMUTE	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED				AXR			
R-0h				R/W-0h			

Table 24-18. MCASP_PDCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AFSR	R/W	0h	Allows the corresponding AFSR bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0h = No effect. 1h = PDOUT[31] bit is cleared to 0.
30	AHCLKR	R/W	0h	Allows the corresponding AHCLKR bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0h = No effect. 1h = PDOUT[30] bit is cleared to 0.
29	ACLKR	R/W	0h	Allows the corresponding ACLKR bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0h = No effect. 1h = PDOUT[29] bit is cleared to 0.
28	AFSX	R/W	0h	Allows the corresponding AFSX bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0h = No effect. 1h = PDOUT[28] bit is cleared to 0.
27	AHCLKX	R/W	0h	Allows the corresponding AHCLKX bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0h = No effect. 1h = PDOUT[27] bit is cleared to 0.
26	ACLKX	R/W	0h	Allows the corresponding ACLKX bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0h = No effect. 1h = PDOUT[26] bit is cleared to 0.
25	AMUTE	R/W	0h	Allows the corresponding AMUTE bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0h = No effect. 1h = PDOUT[25] bit is cleared to 0.

Table 24-18. MCASP_PDCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24-4	RESERVED	R	0h	
3-0	AXR	R/W	0h	Allows the corresponding AXR[n] bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0h = No effect. 1h = PDOUT[n] bit is cleared to 0.

24.4.1.8 MCASP_GBLCTL Register (offset = 44h) [reset = 0h]

MCASP_GBLCTL is shown in Figure 24-46 and described in Table 24-19.

The global control register (GBLCTL) provides initialization of the transmit and receive sections. The bit fields in GBLCTL are synchronized and latched by the corresponding clocks (ACLKX for bits 12-8 and ACLKR for bits 4-0). Before GBLCTL is programmed, you must ensure that serial clocks are running. If the corresponding external serial clocks, ACLKX and ACLKR, are not yet running, you should select the internal serial clock source in AHCLKXCTL, AHCLKRCTL, ACLKXCTL, and ACLKRCTL before GBLCTL is programmed. Also, after programming any bits in GBLCTL you should not proceed until you have read back from GBLCTL and verified that the bits are latched in GBLCTL.

Figure 24-46. MCASP_GBLCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED			XFRST	XSMRST	XSRCLR	XHCLKRST	XCLKRST
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
RESERVED			RFRST	RSMRST	RSRCLR	RHCLKRST	RCLKRST
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-19. MCASP_GBLCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12	XFRST	R/W	0h	Transmit frame sync generator reset enable bit. 0h = Transmit frame sync generator is reset. 1h = Transmit frame sync generator is active. When released from reset, the transmit frame sync generator begins counting serial clocks and generating frame sync as programmed.
11	XSMRST	R/W	0h	Transmit state machine reset enable bit. 0h = Transmit state machine is held in reset. AXRn pin state: If PFUNC[n] = 0 and PDIR[n] = 1; then the serializer drives the AXRn pin to the state specified for inactive time slot (as determined by DISMOD bits in SRCTL). 1h = Transmit state machine is released from reset. When released from reset, the transmit state machine immediately transfers data from XRBUF[n] to XRSR[n]. The transmit state machine sets the underrun flag (XUNDRN) in XSTAT, if XRBUF[n] have not been preloaded with data before reset is released. The transmit state machine also immediately begins detecting frame sync and is ready to transmit. Transmit TDM time slot begins at slot 0 after reset is released.
10	XSRCLR	R/W	0h	Transmit serializer clear enable bit. By clearing then setting this bit, the transmit buffer is flushed to an empty state (XDATA = 1). If XSMRST = 1, XSRCLR = 1, XDATA = 1, and XBUF is not loaded with new data before the start of the next active time slot, an underrun will occur. 0h = Transmit serializers are cleared. 1h = Transmit serializers are active. When the transmit serializers are first taken out of reset (XSRCLR changes from 0 to 1), the transmit data ready bit (XDATA) in XSTAT is set to indicate XBUF is ready to be written.

Table 24-19. MCASP_GBLCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	XHCLKRST	R/W	0h	Transmit high-frequency clock divider reset enable bit. 0h = Transmit high-frequency clock divider is held in reset and passes through its input as divide-by-1. 1h = Transmit high-frequency clock divider is running.
8	XCLKRST	R/W	0h	Transmit clock divider reset enable bit. 0h = Transmit clock divider is held in reset. When the clock divider is in reset, it passes through a divide-by-1 of its input. 1h = Transmit clock divider is running.
7-5	RESERVED	R	0h	
4	RFRST	R/W	0h	Receive frame sync generator reset enable bit. 0h = Receive frame sync generator is reset. 1h = Receive frame sync generator is active. When released from reset, the receive frame sync generator begins counting serial clocks and generating frame sync as programmed.
3	RSMRST	R/W	0h	Receive state machine reset enable bit. 0h = Receive state machine is held in reset. 1h = Receive state machine is released from reset. When released from reset, the receive state machine immediately begins detecting frame sync and is ready to receive. Receive TDM time slot begins at slot 0 after reset is released.
2	RSRCLR	R/W	0h	Receive serializer clear enable bit. By clearing then setting this bit, the receive buffer is flushed. 0h = Receive serializers are cleared. 1h = Receive serializers are active.
1	RHCLKRST	R/W	0h	Receive high-frequency clock divider reset enable bit. 0h = Receive high-frequency clock divider is held in reset and passes through its input as divide-by-1. 1h = Receive high-frequency clock divider is running.
0	RCLKRST	R/W	0h	Receive high-frequency clock divider reset enable bit. 0h = Receive clock divider is held in reset. When the clock divider is in reset, it passes through a divide-by-1 of its input. 1h = Receive clock divider is running.

24.4.1.9 MCASP_AMUTE Register (offset = 48h) [reset = 0h]

MCASP_AMUTE is shown in Figure 24-47 and described in Table 24-20.

The audio mute control register (AMUTE) controls the McASP audio mute (AMUTE) output pin. The value after reset for register 4 depends on how the pins are being driven.

Figure 24-47. MCASP_AMUTE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED			XDMAERR	RDMAERR	XCKFAIL	RCKFAIL	XSYNCERR
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
RSYNCERR	XUNDRN	ROVRN	INSTAT	INEN	INPOL	MUTEN	
R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	

Table 24-20. MCASP_AMUTE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12	XDMAERR	R/W	0h	If transmit DMA error (XDMAERR), drive AMUTE active enable bit. 0h = Drive is disabled. Detection of transmit DMA error is ignored by AMUTE. 1h = Drive is enabled (active). Upon detection of transmit DMA error, AMUTE is active and is driven according to MUTEN bit.
11	RDMAERR	R/W	0h	If receive DMA error (RDMAERR), drive AMUTE active enable bit. 0h = Drive is disabled. Detection of receive DMA error is ignored by AMUTE. 1h = Drive is enabled (active). Upon detection of receive DMA error, AMUTE is active and is driven according to MUTEN bit.
10	XCKFAIL	R/W	0h	If transmit clock failure (XCKFAIL), drive AMUTE active enable bit. 0h = Drive is disabled. Detection of transmit clock failure is ignored by AMUTE. 1h = Drive is enabled (active). Upon detection of transmit clock failure, AMUTE is active and is driven according to MUTEN bit
9	RCKFAIL	R/W	0h	If receive clock failure (RCKFAIL), drive AMUTE active enable bit. 0h = Drive is disabled. Detection of receive clock failure is ignored by AMUTE. 1h = Drive is enabled (active). Upon detection of receive clock failure, AMUTE is active and is driven according to MUTEN bit.
8	XSYNCERR	R/W	0h	If unexpected transmit frame sync error (XSYNCERR), drive AMUTE active enable bit. 0h = Drive is disabled. Detection of unexpected transmit frame sync error is ignored by AMUTE. 1h = Drive is enabled (active). Upon detection of unexpected transmit frame sync error, AMUTE is active and is driven according to MUTEN bit.

Table 24-20. MCASP_AMUTE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	RSYNCERR	R/W	0h	If unexpected receive frame sync error (RSYNCERR), drive AMUTE active enable bit. 0h = Drive is disabled. Detection of unexpected receive frame sync error is ignored by AMUTE. 1h = Drive is enabled (active). Upon detection of unexpected receive frame sync error, AMUTE is active and is driven according to MUTEN bit.
6	XUNDRN	R/W	0h	If transmit underrun error (XUNDRN), drive AMUTE active enable bit. 0h = Drive is disabled. Detection of transmit underrun error is ignored by AMUTE. 1h = Drive is enabled (active). Upon detection of transmit underrun error, AMUTE is active and is driven according to MUTEN bit.
5	ROVRN	R/W	0h	If receiver overrun error (ROVRN), drive AMUTE active enable bit. 0h = Drive is disabled. Detection of receiver overrun error is ignored by AMUTE. 1h = Drive is enabled (active). Upon detection of receiver overrun error, AMUTE is active and is driven according to MUTEN bit.
4	INSTAT	R	0h	Determines drive on AXRn pin when PFUNC[n] and PDIR[n] bits are set to 1. 0h = AMUTEIN pin is inactive. 1h = AMUTEIN pin is active. Audio mute in error is detected.
3	INEN	R/W	0h	Drive AMUTE active when AMUTEIN error is active (INSTAT = 1). 0h = Drive is disabled. AMUTEIN is ignored by AMUTE. 1h = Drive is enabled (active). INSTAT = 1 drives AMUTE active.
2	INPOL	R/W	0h	Audio mute in (AMUTEIN) polarity select bit. 0h = Polarity is active high. A high on AMUTEIN sets INSTAT to 1. 1h = Polarity is active low. A low on AM UTEIN sets INSTAT to 1.
1-0	MUTEN	R/W	0h	AMUTE pin enable bit (unless overridden by GPIO registers). 0h = AMUTE pin is disabled, pin goes to tri-state condition. 1h = AMUTE pin is driven high if error is detected. 2h = AMUTE pin is driven low if error is detected. 3h = Reserved

24.4.1.10 MCASP_DLBCTL Register (offset = 4Ch) [reset = 0h]

MCASP_DLBCTL is shown in Figure 24-48 and described in Table 24-21.

The digital loopback control register (DLBCTL) controls the internal loopback settings of the McASP in TDM mode.

Figure 24-48. MCASP_DLBCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED				MODE		ORD	DLBEN
R-0h				R/W-0h		R/W-0h	R/W-0h

Table 24-21. MCASP_DLBCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-2	MODE	R/W	0h	Loopback generator mode bits. Applies only when loopback mode is enabled (DLBEN = 1). 0h = Default and reserved on loopback mode (DLBEN = 1). When in non-loopback mode (DLBEN = 0), MODE should be left at default (00). When in loopback mode (DLBEN = 1), MODE = 00 is reserved and is not applicable. 1h = Transmit clock and frame sync generators used by both transmit and receive sections. When in loopback mode (DLBEN = 1), MODE must be 01. 2h = Reserved. 3h = Reserved.
1	ORD	R/W	0h	Loopback order bit when loopback mode is enabled (DLBEN = 1). 0h = Odd serializers N + 1 transmit to even serializers N that receive. The corresponding serializers must be programmed properly. 1h = Even serializers N transmit to odd serializers N + 1 that receive. The corresponding serializers must be programmed properly.
0	DLBEN	R/W	0h	Loopback mode enable bit. 0h = Loopback mode is disabled. 1h = Loopback mode is enabled.

24.4.1.11 MCASP_DITCTL Register (offset = 50h) [reset = 0h]

MCASP_DITCTL is shown in Figure 24-49 and described in Table 24-22.

The DIT mode control register (DITCTL) controls DIT operations of the McASP.

Figure 24-49. MCASP_DITCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED				VB	VA	RESERVED	DITEN
R-0h				R/W-0h	R/W-0h	R-0h	R/W-0h

Table 24-22. MCASP_DITCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	VB	R/W	0h	Valid bit for odd time slots (DIT right subframe). 0h = V bit is 0 during odd DIT subframes. 1h = V bit is 1 during odd DIT subframes.
2	VA	R/W	0h	Valid bit for even time slots (DIT left subframe). 0h = V bit is 0 during even DIT subframes. 1h = V bit is 1 during even DIT subframes.
1	RESERVED	R	0h	
0	DITEN	R/W	0h	DIT mode enable bit. DITEN should only be changed while the XSMRST bit in GBLCTL is in reset (and for startup, XSRCLR also in reset). However, it is not necessary to reset the XCLKRST or XHCLKRST bits in GBLCTL to change DITEN. 0h = DIT mode is disabled. Transmitter operates in TDM or burst mode. 1h = DIT mode is enabled. Transmitter operates in DIT encoded mode.

24.4.1.12 MCASP_RGBLCTL Register (offset = 60h) [reset = 0h]

MCASP_RGBLCTL is shown in Figure 24-50 and described in Table 24-23.

Alias of the global control register (GBLCTL). Writing to the receiver global control register (RGBLCTL) affects only the receive bits of GBLCTL (bits 4-0). Reads from RGBLCTL return the value of GBLCTL. RGBLCTL allows the receiver to be reset independently from the transmitter.

Figure 24-50. MCASP_RGBLCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED			XFRST	XSMRST	XSRCLR	XHCLKRST	XCLKRST
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h

7	6	5	4	3	2	1	0
RESERVED			RFRST	RSMRST	RSRCLR	RHCLKRST	RCLKRST
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-23. MCASP_RGBLCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12	XFRST	R	0h	Transmit frame sync generator reset enable bit. A read of this bit returns the XFRST bit value of GBLCTL. Writes have no effect.
11	XSMRST	R	0h	Transmit state machine reset enable bit. A read of this bit returns the XSMRST bit value of GBLCTL. Writes have no effect.
10	XSRCLR	R	0h	Transmit serializer clear enable bit. A read of this bit returns the XSRCLR bit value of GBLCTL. Writes have no effect.
9	XHCLKRST	R	0h	Transmit high-frequency clock divider reset enable bit. A read of this bit returns the XHCLKRST bit value of GBLCTL. Writes have no effect.
8	XCLKRST	R	0h	Transmit clock divider reset enable bit. A read of this bit returns the XCLKRST bit value of GBLCTL. Writes have no effect.
7-5	RESERVED	R	0h	
4	RFRST	R/W	0h	Receive frame sync generator reset enable bit. A write to this bit affects the RFRST bit of GBLCTL. 0h = Receive frame sync generator is reset. 1h = Receive frame sync generator is active.
3	RSMRST	R/W	0h	Receive state machine reset enable bit. A write to this bit affects the RSMRST bit of GBLCTL. 0h = Receive state machine is held in reset. 1h = Receive state machine is released from reset.
2	RSRCLR	R/W	0h	Receive serializer clear enable bit. A write to this bit affects the RSRCLR bit of GBLCTL. 0h = Receive serializers are cleared. 1h = Receive serializers are active.

Table 24-23. MCASP_RGBLCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	RHCLKRST	R/W	0h	Receive high-frequency clock divider reset enable bit. A write to this bit affects the RHCLKRST bit of GBLCTL. 0h = Receive high-frequency clock divider is held in reset and passes through its input as divide-by-1. 1h = Receive high-frequency clock divider is running.
0	RCLKRST	R/W	0h	Receive clock divider reset enable bit. A write to this bit affects the RCLKRST bit of GBLCTL. 0h = Receive clock divider is held in reset. 1h = Receive clock divider is running.

24.4.1.13 MCASP_RMASK Register (offset = 64h) [reset = 0h]

MCASP_RMASK is shown in Figure 24-51 and described in Table 24-24.

The receive format unit bit mask register (RMASK) determines which bits of the received data are masked off and padded with a known value before being read by the CPU or DMA.

Figure 24-51. MCASP_RMASK Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">RMASK</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 24-24. MCASP_RMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RMASK	R/W	0h	Receive data mask n enable bit. 0h = Corresponding bit of receive data (after passing through reverse and rotate units) is masked out and then padded with the selected bit pad value (RPAD and RPBIT bits in RFMT). 1h = Corresponding bit of receive data (after passing through reverse and rotate units) is returned to CPU or DMA.

24.4.1.14 MCASP_RFMT Register (offset = 68h) [reset = 0h]

MCASP_RFMT is shown in Figure 24-52 and described in Table 24-25.

The receive bit stream format register (RFMT) configures the receive data format.

Figure 24-52. MCASP_RFMT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED						RDATDLY	
R-0h						R/W-0h	

15	14	13	12	11	10	9	8
RRVRS	RPAD		RPBIT				
R/W-0h	R/W-0h		R/W-0h				

7	6	5	4	3	2	1	0
RSSZ				RBUSEL	RROT		
R/W-0h				R/W-0h	R/W-0h		

Table 24-25. MCASP_RFMT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	
17-16	RDATDLY	R/W	0h	Receive bit delay. 0h = 0-bit delay. The first receive data bit, AXRn, occurs in same ACLKR cycle as the receive frame sync (AFSR). 1h = 1-bit delay. The first receive data bit, AXRn, occurs one ACLKR cycle after the receive frame sync (AFSR). 2h = 2-bit delay. The first receive data bit, AXRn, occurs two ACLKR cycles after the receive frame sync (AFSR). 3h = Reserved.
15	RRVRS	R/W	0h	Receive serial bitstream order. 0h = Bitstream is LSB first. No bit reversal is performed in receive format bit reverse unit. 1h = Bitstream is MSB first. Bit reversal is performed in receive format bit reverse unit.
14-13	RPAD	R/W	0h	Pad value for extra bits in slot not belonging to the word. This field only applies to bits when RMASK[n] = 0. 0h = Pad extra bits with 0. 1h = Pad extra bits with 1. 2h = Pad extra bits with one of the bits from the word as specified by RPBIT bits. 3h = Reserved.
12-8	RPBIT	R/W	0h	RPBIT value determines which bit (as read by the CPU or DMA from RBUF[n]) is used to pad the extra bits. This field only applies when RPAD = 2h. 0h = Pad with bit 0 value. 1h = Pad with bit 1 to bit 31 value from 1h to 1Fh.

Table 24-25. MCASP_RFMT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-4	RSSZ	R/W	0h	Receive slot size. 0h = Reseved. 1h = Reserved. 2h = Reserved. 3h = Slot size is 8 bits. 4h = Reserved 5h = Slot size is 12 bits. 6h = Reserved 7h = Slot size is 16 bits. 8h = Reserved 9h = Slot size is 20 bits. Ah = Reserved 0xB = Slot size is 24 bits Ch = Reserved Dh = Slot size is 28 bits. Eh = Reserved Fh = Slot size is 32 bits.
3	RBUSEL	R/W	0h	Selects whether reads from serializer buffer XRBUF[n] originate from the configuration bus (CFG) or the data (DAT) port. 0h = Reads from XRBUF[n] originate on data port. Reads from XRBUF[n] on configuration bus are ignored. 1h = Reads from XRBUF[n] originate on configuration bus. Reads from XRBUF[n] on data port are ignored.
2-0	RROT	R/W	0h	Right-rotation value for receive rotate right format unit. 0h = Rotate right by 0 (no rotation). 1h = Rotate right by 4 bit positions. 2h = Rotate right by 8 bit positions. 3h = Rotate right by 12 bit positions. 4h = Rotate right by 16 bit positions. 5h = Rotate right by 20 bit positions. 6h = Rotate right by 24 bit positions. 7h = Rotate right by 28 bit positions.

24.4.1.15 MCASP_AFSRCTL Register (offset = 6Ch) [reset = 0h]

MCASP_AFSRCTL is shown in Figure 24-53 and described in Table 24-26.

The receive frame sync control register (AFSRCTL) configures the receive frame sync (AFSR).

Figure 24-53. MCASP_AFSRCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RMOD							
R/W-0h							

7	6	5	4	3	2	1	0
RMOD	RESERVED		FRWID	RESERVED		FSRM	FSRP
R/W-0h	R-0h		R/W-0h	R-0h		R/W-0h	R/W-0h

Table 24-26. MCASP_AFSRCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-7	RMOD	R/W	0h	Receive frame sync mode select bits. 1FFh = Reserved from 181h to 1FFh. 0h = Burst mode. 1h = Reserved. 2h = 2-slot TDM (I2S mode) to 32-slot TDM from 2h to 20h. 21h = Reserved from 21h to 17Fh. 180h = 384-slot TDM (external DIR IC inputting 384-slot DIR frames to McASP over I2S interface). 181h = Reserved from 181h to 1FFh.
6-5	RESERVED	R	0h	
4	FRWID	R/W	0h	Receive frame sync width select bit indicates the width of the receive frame sync (AFSR) during its active period. 0h = Single bit. 1h = Single word.
3-2	RESERVED	R	0h	
1	FSRM	R/W	0h	Receive frame sync generation select bit. 0h = Externally-generated receive frame sync. 1h = Internally-generated receive frame sync.
0	FSRP	R/W	0h	Receive frame sync polarity select bit. 0h = A rising edge on receive frame sync (AFSR) indicates the beginning of a frame. 1h = A falling edge on receive frame sync (AFSR) indicates the beginning of a frame.

24.4.1.16 MCASP_ACLKRCTL Register (offset = 70h) [reset = 0h]

MCASP_ACLKRCTL is shown in Figure 24-54 and described in Table 24-27.

The receive clock control register (ACLKRCTL) configures the receive bit clock (ACLKR) and the receive clock generator.

Figure 24-54. MCASP_ACLKRCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
CLKRP	RESERVED	CLKRM	CLKRDIV				
R/W-0h	R-0h	R/W-0h	R/W-0h				

Table 24-27. MCASP_ACLKRCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	CLKRP	R/W	0h	Receive bitstream clock polarity select bit. 0h = Falling edge. Receiver samples data on the falling edge of the serial clock, so the external transmitter driving this receiver must shift data out on the rising edge of the serial clock. 1h = Rising edge. Receiver samples data on the rising edge of the serial clock, so the external transmitter driving this receiver must shift data out on the falling edge of the serial clock.
6	RESERVED	R	0h	
5	CLKRM	R/W	0h	Receive bit clock source bit. Note that this bit does not have any effect, if ACLKXCTL.ASYNC = 0. 0h = External receive clock source from ACLKR pin. 1h = Internal receive clock source from output of programmable bit clock divider.
4-0	CLKRDIV	R/W	0h	Receive bit clock divide ratio bits determine the divide-down ratio from AHCLKR to ACLKR. Note that this bit does not have any effect, if ACLKXCTL.ASYNC = 0. 0h = Divide-by-1. 1h = Divide-by-2. 2h = Divide-by-3 to divide-by-32 from 2h to 1Fh.

24.4.1.17 MCASP_AHCLKRCTL Register (offset = 74h) [reset = 0h]

MCASP_AHCLKRCTL is shown in Figure 24-55 and described in Table 24-28.

The receive high-frequency clock control register (AHCLKRCTL) configures the receive high-frequency master clock (AHCLKR) and the receive clock generator.

Figure 24-55. MCASP_AHCLKRCTL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>HCLKRM</td><td>HCLKRP</td><td colspan="2">RESERVED</td><td colspan="4">HCLKRDIV</td></tr>
<tr><td>R/W-0h</td><td>R/W-0h</td><td colspan="2">R-0h</td><td colspan="4">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">HCLKRDIV</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 24-28. MCASP_AHCLKRCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	HCLKRM	R/W	0h	Receive high-frequency clock source bit. 0h = External receive high-frequency clock source from AHCLKR pin. 1h = Internal receive high-frequency clock source from output of programmable high clock divider.
14	HCLKRP	R/W	0h	Receive bitstream high-frequency clock polarity select bit. 0h = AHCLKR is not inverted before programmable bit clock divider. In the special case where the receive bit clock (ACLKR) is internally generated and the programmable bit clock divider is set to divide-by-1 (CLKRDIV = 0 in ACLKRCTL), AHCLKR is directly passed through to the ACLKR pin. 1h = AHCLKR is inverted before programmable bit clock divider. In the special case where the receive bit clock (ACLKR) is internally generated and the programmable bit clock divider is set to divide-by-1 (CLKRDIV = 0 in ACLKRCTL), AHCLKR is directly passed through to the ACLKR pin.
13-12	RESERVED	R	0h	
11-0	HCLKRDIV	R/W	0h	Receive high-frequency clock divide ratio bits determine the divide-down ratio from AUXCLK to AHCLKR. 0h = Divide-by-1. 1h = Divide-by-2. 2h = Divide-by-3 to divide-by-4096 from 2h to FFFh.

24.4.1.18 MCASP_RTDM Register (offset = 78h) [reset = 0h]

MCASP_RTDM is shown in Figure 24-56 and described in Table 24-29.

The receive TDM time slot register (RTDM) specifies which TDM time slot the receiver is active.

Figure 24-56. MCASP_RTDM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTDMS																															
R/W-0h																															

Table 24-29. MCASP_RTDM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RTDMS	R/W	0h	Receiver mode during TDM time slot n. 0h = Receive TDM time slot n is inactive. The receive serializer does not shift in data during this slot. 1h = Receive TDM time slot n is active. The receive serializer shifts in data during this slot.

24.4.1.19 MCASP_RINTCTL Register (offset = 7Ch) [reset = 0h]

MCASP_RINTCTL is shown in Figure 24-57 and described in Table 24-30.

The receiver interrupt control register (RINTCTL) controls generation of the McASP receive interrupt (RINT). When the register bit(s) is set to 1, the occurrence of the enabled McASP condition(s) generates RINT. See the RSTAT register for a description of the interrupt conditions.

Figure 24-57. MCASP_RINTCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RSTAFRM	RESERVED	RDATA	RLAST	RDMAERR	RCKFAIL	RSYNCERR	ROVRN
R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-30. MCASP_RINTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	RSTAFRM	R/W	0h	Receive start of frame interrupt enable bit. 0h = Interrupt is disabled. A receive start of frame interrupt does not generate a McASP receive interrupt (RINT). 1h = Interrupt is enabled. A receive start of frame interrupt generates a McASP receive interrupt (RINT).
6	RESERVED	R	0h	
5	RDATA	R/W	0h	Receive data ready interrupt enable bit. 0h = Interrupt is disabled. A receive data ready interrupt does not generate a McASP receive interrupt (RINT). 1h = Interrupt is enabled. A receive data ready interrupt generates a McASP receive interrupt (RINT).
4	RLAST	R/W	0h	Receive last slot interrupt enable bit. 0h = Interrupt is disabled. A receive last slot interrupt does not generate a McASP receive interrupt (RINT). 1h = Interrupt is enabled. A receive last slot interrupt generates a McASP receive interrupt (RINT).
3	RDMAERR	R/W	0h	Receive DMA error interrupt enable bit. 0h = Interrupt is disabled. A receive DMA error interrupt does not generate a McASP receive interrupt (RINT). 1h = Interrupt is enabled. A receive DMA error interrupt generates a McASP receive interrupt (RINT).
2	RCKFAIL	R/W	0h	Receive clock failure interrupt enable bit. 0h = Interrupt is disabled. A receive clock failure interrupt does not generate a McASP receive interrupt (RINT). 1h = Interrupt is enabled. A receive clock failure interrupt generates a McASP receive interrupt (RINT).
1	RSYNCERR	R/W	0h	Unexpected receive frame sync interrupt enable bit. 0h = Interrupt is disabled. An unexpected receive frame sync interrupt does not generate a McASP receive interrupt (RINT). 1h = Interrupt is enabled. An unexpected receive frame sync interrupt generates a McASP receive interrupt (RINT).

Table 24-30. MCASP_RINTCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	ROVRN	R/W	0h	Receiver overrun interrupt enable bit. 0h = Interrupt is disabled. A receiver overrun interrupt does not generate a McASP receive interrupt (RINT). 1h = Interrupt is enabled. A receiver overrun interrupt generates a McASP receive interrupt (RINT).

24.4.1.20 MCASP_RSTAT Register (offset = 80h) [reset = 0h]

MCASP_RSTAT is shown in Figure 24-58 and described in Table 24-31.

The receiver status register (RSTAT) provides the receiver status and receive TDM time slot number. If the McASP logic attempts to set an interrupt flag in the same cycle that the CPU writes to the flag to clear it, the McASP logic has priority and the flag remains set. This also causes a new interrupt request to be generated.

Figure 24-58. MCASP_RSTAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							RERR
R-0h							R/W-0h

7	6	5	4	3	2	1	0
RDMAERR	RSTAFRM	RDATA	RLAST	RTDMSLOT	RCKFAIL	RSYNCERR	ROVRN
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

Table 24-31. MCASP_RSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	RERR	R/W	0h	RERR bit always returns a logic-OR of: ROVRN OR RSYNCERR OR RCKFAIL OR RDMAERR. Allows a single bit to be checked to determine if a receiver error interrupt has occurred. 0h = No errors have occurred. 1h = An error has occurred.
7	RDMAERR	R/W1C	0h	Receive DMA error flag. RDMAERR is set when the CPU or DMA reads more serializers through the data port in a given time slot than were programmed as receivers. Causes a receive interrupt (RINT), if this bit is set and RDMAERR in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0h = Receive DMA error did not occur. 1h = Receive DMA error did occur.
6	RSTAFRM	R/W1C	0h	Receive start of frame flag. Causes a receive interrupt (RINT), if this bit is set and RSTAFRM in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0h = No new receive frame sync (AFSR) is detected. 1h = A new receive frame sync (AFSR) is detected.
5	RDATA	R/W1C	0h	Receive data ready flag. Causes a receive interrupt (RINT), if this bit is set and RDATA in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0h = No new data in RBUF. 1h = Data is transferred from XRSR to RBUF and ready to be serviced by the CPU or DMA. When RDATA is set, it always causes a DMA event (AREVT).

Table 24-31. MCASP_RSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	RLAST	R/W1C	0h	Receive last slot flag. RLAST is set along with RDATA, if the current slot is the last slot in a frame. Causes a receive interrupt (RINT), if this bit is set and RLAST in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0h = Current slot is not the last slot in a frame. 1h = Current slot is the last slot in a frame. RDATA is also set.
3	RTDMSLOT	R	0h	Returns the LSB of RSLOT. Allows a single read of RSTAT to determine whether the current TDM time slot is even or odd. 0h = Current TDM time slot is odd. 1h = Current TDM time slot is even.
2	RCKFAIL	R/W1C	0h	Receive clock failure flag. RCKFAIL is set when the receive clock failure detection circuit reports an error. Causes a receive interrupt (RINT), if this bit is set and RCKFAIL in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0h = Receive clock failure did not occur. 1h = Receive clock failure did occur.
1	RSYNCERR	R/W1C	0h	Unexpected receive frame sync flag. RSYNCERR is set when a new receive frame sync (AFSR) occurs before it is expected. Causes a receive interrupt (RINT), if this bit is set and RSYNCERR in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0h = Unexpected receive frame sync did not occur. 1h = Unexpected receive frame sync did occur.
0	ROVRN	R/W1C	0h	Receiver overrun flag. ROVRN is set when the receive serializer is instructed to transfer data from XRSR to RBUF, but the former data in RBUF has not yet been read by the CPU or DMA. Causes a receive interrupt (RINT), if this bit is set and ROVRN in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0h = Receiver overrun did not occur. 1h = Receiver overrun did occur.

24.4.1.21 MCASP_RSLOT Register (offset = 84h) [reset = 0h]

MCASP_RSLOT is shown in Figure 24-59 and described in Table 24-32.

The current receive TDM time slot register (RSLOT) indicates the current time slot for the receive data frame.

Figure 24-59. MCASP_RSLOT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="23">RESERVED</td><td colspan="9">RSLOTCNT</td></tr>
<tr><td colspan="23">R-0h</td><td colspan="9">R-0h</td></tr>
</table>

Table 24-32. MCASP_RSLOT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-0	RSLOTCNT	R	0h	0-17Fh = Current receive time slot count. Legal values: 0 to 383 (17Fh). TDM function is not supported for > 32 time slots. However, TDM time slot counter may count to 383 when used to receive a DIR block (transferred over TDM format).

24.4.1.22 MCASP_RCLKCHK Register (offset = 88h) [reset = 0h]

MCASP_RCLKCHK is shown in Figure 24-60 and described in Table 24-33.

The receive clock check control register (RCLKCHK) configures the receive clock failure detection circuit.

Figure 24-60. MCASP_RCLKCHK Register

31-24	23-16
RCNT	RMAX
R-0h	R/W-0h

15-8	7-4	3-0
RMIN	RESERVED	RPS
R/W-0h	R-0h	R/W-0h

Table 24-33. MCASP_RCLKCHK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RCNT	R	0h	Receive clock count value (from previous measurement). The clock circuit continually counts the number of system clocks for every 32 receive high-frequency master clock (AHCLKR) signals, and stores the count in RCNT until the next measurement is taken.
23-16	RMAX	R/W	0h	Receive clock maximum boundary. This 8 bit unsigned value sets the maximum allowed boundary for the clock check counter after 32 receive high-frequency master clock (AHCLKR) signals have been received. If the current counter value is greater than RMAX after counting 32 AHCLKR signals, RCKFAIL in RSTAT is set. The comparison is performed using unsigned arithmetic.
15-8	RMIN	R/W	0h	Receive clock minimum boundary. This 8 bit unsigned value sets the minimum allowed boundary for the clock check counter after 32 receive high-frequency master clock (AHCLKR) signals have been received. If RCNT is less than RMIN after counting 32 AHCLKR signals, RCKFAIL in RSTAT is set. The comparison is performed using unsigned arithmetic.
7-4	RESERVED	R	0h	
3-0	RPS	R/W	0h	Receive clock check prescaler value. 0h = McASP system clock divided by 1. 1h = McASP system clock divided by 2. 2h = McASP system clock divided by 4. 3h = McASP system clock divided by 8. 4h = McASP system clock divided by 16. 5h = McASP system clock divided by 32. 6h = McASP system clock divided by 64. 7h = McASP system clock divided by 128. 8h = McASP system clock divided by 256. 9h = Reserved from 9h to Fh.

24.4.1.23 MCASP_REVTCTL Register (offset = 8Ch) [reset = 0h]

MCASP_REVTCTL is shown in Figure 24-61 and described in Table 24-34.

The receiver DMA event control register (REVTCTL) contains a disable bit for the receiver DMA event. Note for device-specific registers: Accessing REVTCTL not implemented on a specific device may cause improper operation.

Figure 24-61. MCASP_REVTCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							RDATDMA
R-0h							R/W-0h

Table 24-34. MCASP_REVTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	RDATDMA	R/W	0h	Receive data DMA request enable bit. If writing to this bit, always write the default value of 0. 0h = Receive data DMA request is enabled. 1h = Reserved

24.4.1.24 MCASP_XGBLCTL Register (offset = A0h) [reset = 0h]

MCASP_XGBLCTL is shown in Figure 24-62 and described in Table 24-35.

Alias of the global control register (GBLCTL). Writing to the transmitter global control register (XGBLCTL) affects only the transmit bits of GBLCTL (bits 12-8). Reads from XGBLCTL return the value of GBLCTL. XGBLCTL allows the transmitter to be reset independently from the receiver. See the GBLCTL register for a detailed description of GBLCTL.

Figure 24-62. MCASP_XGBLCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED			XFRST	XSMRST	XSRCLR	XHCLKRST	XCLKRST
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
RESERVED			RFRST	RSMRST	RSRCLR	RHCLKRST	RCLKRST
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h

Table 24-35. MCASP_XGBLCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12	XFRST	R/W	0h	Transmit frame sync generator reset enable bit. A write to this bit affects the XFRST bit of GBLCTL. 0h = Transmit frame sync generator is reset. 1h = Transmit frame sync generator is active.
11	XSMRST	R/W	0h	Transmit state machine reset enable bit. A write to this bit affects the XSMRST bit of GBLCTL. 0h = Transmit state machine is held in reset. 1h = Transmit state machine is released from reset.
10	XSRCLR	R/W	0h	Transmit serializer clear enable bit. A write to this bit affects the XSRCLR bit of GBLCTL. 0h = Transmit serializers are cleared. 1h = Transmit serializers are active.
9	XHCLKRST	R/W	0h	Transmit high-frequency clock divider reset enable bit. A write to this bit affects the XHCLKRST bit of GBLCTL. 0h = Transmit high-frequency clock divider is held in reset. 1h = Transmit high-frequency clock divider is running.
8	XCLKRST	R/W	0h	Transmit clock divider reset enable bit. A write to this bit affects the XCLKRST bit of GBLCTL. 0h = Transmit clock divider is held in reset. 1h = Transmit clock divider is running.
7-5	RESERVED	R	0h	
4	RFRST	R	0h	Receive frame sync generator reset enable bit. A read of this bit returns the RFRST bit value of GBLCTL. Writes have no effect.
3	RSMRST	R	0h	Receive state machine reset enable bit. A read of this bit returns the RSMRST bit value of GBLCTL. Writes have no effect.
2	RSRCLR	R	0h	Receive serializer clear enable bit. A read of this bit returns the RSRSCLR bit value of GBLCTL. Writes have no effect.

Table 24-35. MCASP_XGBLCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	RHCLKRST	R	0h	Receive high-frequency clock divider reset enable bit. A read of this bit returns the RHCLKRST bit value of GBLCTL. Writes have no effect.
0	RCLKRST	R	0h	Receive clock divider reset enable bit. A read of this bit returns the RCLKRST bit value of GBLCTL. Writes have no effect.

24.4.1.25 MCASP_XMASK Register (offset = A4h) [reset = 0h]

MCASP_XMASK is shown in Figure 24-63 and described in Table 24-36.

The transmit format unit bit mask register (XMASK) determines which bits of the transmitted data are masked off and padded with a known value before being shifted out the McASP.

Figure 24-63. MCASP_XMASK Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">XMASK</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 24-36. MCASP_XMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	XMASK	R/W	0h	Transmit data mask n enable bit. 0h = Corresponding bit of transmit data (before passing through reverse and rotate units) is masked out and then padded with the selected bit pad value (XPAD and XPBIT bits in XFMT), which is transmitted out the McASP in place of the original bit. 1h = Corresponding bit of transmit data (before passing through reverse and rotate units) is transmitted out the McASP.

24.4.1.26 MCASP_XFMT Register (offset = A8h) [reset = 0h]

MCASP_XFMT is shown in Figure 24-64 and described in Table 24-37.

The transmit bit stream format register (XFMT) configures the transmit data format.

Figure 24-64. MCASP_XFMT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">XDATDLY</td></tr>
<tr><td colspan="6">R-0h</td><td colspan="2">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>XRVRS</td><td colspan="2">XPAD</td><td colspan="5">XPBIT</td></tr>
<tr><td>R/W-0h</td><td colspan="2">R/W-0h</td><td colspan="5">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">XSSZ</td><td>XBUSEL</td><td colspan="3">XROT</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R/W-0h</td><td colspan="3">R/W-0h</td></tr>
</table>

Table 24-37. MCASP_XFMT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	
17-16	XDATDLY	R/W	0h	Transmit sync bit delay. 0h = 0-bit delay. The first transmit data bit, AXRn, occurs in same ACLKX cycle as the transmit frame sync (AFSX). 1h = 1-bit delay. The first transmit data bit, AXRn, occurs one ACLKX cycle after the transmit frame sync (AFSX). 2h = 2-bit delay. The first transmit data bit, AXRn, occurs two ACLKX cycles after the transmit frame sync (AFSX). 3h = Reserved.
15	XRVRS	R/W	0h	Transmit serial bitstream order. 0h = Bitstream is LSB first. No bit reversal is performed in transmit format bit reverse unit. 1h = Bitstream is MSB first. Bit reversal is performed in transmit format bit reverse unit.
14-13	XPAD	R/W	0h	Pad value for extra bits in slot not belonging to word defined by XMASK. This field only applies to bits when XMASK[n] = 0. 0h = Pad extra bits with 0. 1h = Pad extra bits with 1. 2h = Pad extra bits with one of the bits from the word as specified by XPBIT bits. 3h = Reserved.
12-8	XPBIT	R/W	0h	XPBIT value determines which bit (as written by the CPU or DMA to XBUF[n]) is used to pad the extra bits before shifting. This field only applies when XPAD = 2h. 0h = Pad with bit 0 value. 1h = Pad with bit 1 to bit 31 value from 1h to 1Fh.

Table 24-37. MCASP_XFMT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-4	XSSZ	R/W	0h	Transmit slot size. 0h = Reserved. 1h = Reserved. 2h = Reserved. 3h = Slot size is 8 bits. 4h = Reserved. 5h = Slot size is 12 bits. 6h = Reserved. 7h = Slot size is 16 bits. 8h = Reserved. 9h = Slot size is 20 bits. Ah = Reserved. 0xB = Slot size is 24 bits. Ch = Reserved. Dh = Slot size is 28 bits. Eh = Reserved. Fh = Slot size is 32 bits.
3	XBUSEL	R/W	0h	Selects whether writes to serializer buffer XRBUF[n] originate from the configuration bus (CFG) or the data (DAT) port. 0h = Writes to XRBUF[n] originate from the data port. Writes to XRBUF[n] from the configuration bus are ignored with no effect to the McASP. 1h = Writes to XRBUF[n] originate from the configuration bus. Writes to XRBUF[n] from the data port are ignored with no effect to the McASP.
2-0	XROT	R/W	0h	Right-rotation value for transmit rotate right format unit. 0h = Rotate right by 0 (no rotation). 1h = Rotate right by 4 bit positions. 2h = Rotate right by 8 bit positions. 3h = Rotate right by 12 bit positions. 4h = Rotate right by 16 bit positions. 5h = Rotate right by 20 bit positions. 6h = Rotate right by 24 bit positions. 7h = Rotate right by 28 bit positions.

24.4.1.27 MCASP_AFSXCTL Register (offset = ACh) [reset = 0h]

MCASP_AFSXCTL is shown in Figure 24-65 and described in Table 24-38.

The transmit frame sync control register (AFSXCTL) configures the transmit frame sync (AFSX).

Figure 24-65. MCASP_AFSXCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
XMOD							
R/W-0h							

7	6	5	4	3	2	1	0
XMOD	RESERVED		FXWID	RESERVED		FSXM	FSXP
R/W-0h	R-0h		R/W-0h	R-0h		R/W-0h	R/W-0h

Table 24-38. MCASP_AFSXCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-7	XMOD	R/W	0h	Transmit frame sync mode select bits. 1FFh = Reserved from 181h to 1FFh. 0h = Burst mode. 1h = Reserved. 2h = 2-slot TDM (I2S mode) to 32-slot TDM from 2h to 20h. 21h = Reserved from 21h to 17Fh. 180h = 384-slot DIT mode. 181h = Reserved from 181h to 1FFh.
6-5	RESERVED	R	0h	
4	FXWID	R/W	0h	Transmit frame sync width select bit indicates the width of the transmit frame sync (AFSX) during its active period. 0h = Single bit. 1h = Single word.
3-2	RESERVED	R	0h	
1	FSXM	R/W	0h	Transmit frame sync generation select bit. 0h = Externally-generated transmit frame sync. 1h = Internally-generated transmit frame sync.
0	FSXP	R/W	0h	Transmit frame sync polarity select bit. 0h = A rising edge on transmit frame sync (AFSX) indicates the beginning of a frame. 1h = A falling edge on transmit frame sync (AFSX) indicates the beginning of a frame.

24.4.1.28 MCASP_ACLKXCTL Register (offset = B0h) [reset = 60h]

MCASP_ACLKXCTL is shown in Figure 24-66 and described in Table 24-39.

The transmit clock control register (ACLKXCTL) configures the transmit bit clock (ACLKX) and the transmit clock generator.

Figure 24-66. MCASP_ACLKXCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CLKXP	ASYNC	CLKXM	CLKXDIV				
R/W-0h	R/W-1h	R/W-1h	R/W-0h				

Table 24-39. MCASP_ACLKXCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	CLKXP	R/W	0h	Transmit bitstream clock polarity select bit. 0h = Rising edge. External receiver samples data on the falling edge of the serial clock, so the transmitter must shift data out on the rising edge of the serial clock. 1h = Falling edge. External receiver samples data on the rising edge of the serial clock, so the transmitter must shift data out on the falling edge of the serial clock.
6	ASYNC	R/W	1h	Transmit/receive operation asynchronous enable bit. 0h = Synchronous. Transmit clock and frame sync provides the source for both the transmit and receive sections. 1h = Asynchronous. Separate clock and frame sync used by transmit and receive sections.
5	CLKXM	R/W	1h	Transmit bit clock source bit. 0h = External transmit clock source from ACLKX pin. 1h = Internal transmit clock source from output of programmable bit clock divider.
4-0	CLKXDIV	R/W	0h	Transmit bit clock divide ratio bits determine the divide-down ratio from AHCLKX to ACLKX. 0h = Divide-by-1. 1h = Divide-by-2. 2h = Divide-by-3 to divide-by-32 from 2h to 1Fh.

24.4.1.29 MCASP_AHCLKXCTL Register (offset = B4h) [reset = 0h]

MCASP_AHCLKXCTL is shown in Figure 24-67 and described in Table 24-40.

The transmit high-frequency clock control register (AHCLKXCTL) configures the transmit high-frequency master clock (AHCLKX) and the transmit clock generator.

Figure 24-67. MCASP_AHCLKXCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
HCLKXM	HCLKXP	RESERVED		HCLKXDIV			
R/W-0h	R/W-0h	R-0h		R/W-0h			

7	6	5	4	3	2	1	0
HCLKXDIV							
R/W-0h							

Table 24-40. MCASP_AHCLKXCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	HCLKXM	R/W	0h	Transmit high-frequency clock source bit. 0h = External transmit high-frequency clock source from AHCLKX pin. 1h = Internal transmit high-frequency clock source from output of programmable high clock divider.
14	HCLKXP	R/W	0h	Transmit bitstream high-frequency clock polarity select bit. 0h = AHCLKX is not inverted before programmable bit clock divider. In the special case where the transmit bit clock (ACLKX) is internally generated and the programmable bit clock divider is set to divide-by-1 (CLKXDIV = 0 in ACLKXCTL), AHCLKX is directly passed through to the ACLKX pin. 1h = AHCLKX is inverted before programmable bit clock divider. In the special case where the transmit bit clock (ACLKX) is internally generated and the programmable bit clock divider is set to divide-by-1 (CLKXDIV = 0 in ACLKXCTL), AHCLKX is directly passed through to the ACLKX pin.
13-12	RESERVED	R	0h	
11-0	HCLKXDIV	R/W	0h	Transmit high-frequency clock divide ratio bits determine the divide-down ratio from AUXCLK to AHCLKX. 0h = Divide-by-1. 1h = Divide-by-2. 2h = Divide-by-3 to divide-by-4096 from 2h to FFFh.

24.4.1.30 MCASP_XTDM Register (offset = B8h) [reset = 0h]

MCASP_XTDM is shown in Figure 24-68 and described in Table 24-41.

The transmit TDM time slot register (XTDM) specifies in which TDM time slot the transmitter is active. TDM time slot counter range is extended to 384 slots (to support SPDIF blocks of 384 subframes). XTDM operates modulo 32, that is, XTDMS specifies the TDM activity for time slots 0, 32, 64, 96, 128, and so on.

Figure 24-68. MCASP_XTDM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XTDMS																															
R/W-0h																															

Table 24-41. MCASP_XTDM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	XTDMS	R/W	0h	Transmitter mode during TDM time slot n. 0h = Transmit TDM time slot n is inactive. The transmit serializer does not shift out data during this slot. 1h = Transmit TDM time slot n is active. The transmit serializer shifts out data during this slot according to the serializer control register (SRCTL).

24.4.1.31 MCASP_XINTCTL Register (offset = BCh) [reset = 0h]

MCASP_XINTCTL is shown in Figure 24-69 and described in Table 24-42.

The transmitter interrupt control register (XINTCTL) controls generation of the McASP transmit interrupt (XINT). When the register bit(s) is set to 1, the occurrence of the enabled McASP condition(s) generates XINT. See the XSTAT register for a description of the interrupt conditions.

Figure 24-69. MCASP_XINTCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
XSTAFRM	RESERVED	XDATA	XLAST	XDMAERR	XCKFAIL	XSYNCERR	XUNDRN
R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-42. MCASP_XINTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	XSTAFRM	R/W	0h	Transmit start of frame interrupt enable bit. 0h = Interrupt is disabled. A transmit start of frame interrupt does not generate a McASP transmit interrupt (XINT). 1h = Interrupt is enabled. A transmit start of frame interrupt generates a McASP transmit interrupt (XINT).
6	RESERVED	R	0h	
5	XDATA	R/W	0h	Transmit data ready interrupt enable bit. 0h = Interrupt is disabled. A transmit data ready interrupt does not generate a McASP transmit interrupt (XINT). 1h = Interrupt is enabled. A transmit data ready interrupt generates a McASP transmit interrupt (XINT).
4	XLAST	R/W	0h	Transmit last slot interrupt enable bit. 0h = Interrupt is disabled. A transmit last slot interrupt does not generate a McASP transmit interrupt (XINT). 1h = Interrupt is enabled. A transmit last slot interrupt generates a McASP transmit interrupt (XINT).
3	XDMAERR	R/W	0h	Transmit DMA error interrupt enable bit. 0h = Interrupt is disabled. A transmit DMA error interrupt does not generate a McASP transmit interrupt (XINT). 1h = Interrupt is enabled. A transmit DMA error interrupt generates a McASP transmit interrupt (XINT).
2	XCKFAIL	R/W	0h	Transmit clock failure interrupt enable bit. 0h = Interrupt is disabled. A transmit clock failure interrupt does not generate a McASP transmit interrupt (XINT). 1h = Interrupt is enabled. A transmit clock failure interrupt generates a McASP transmit interrupt (XINT).
1	XSYNCERR	R/W	0h	Unexpected transmit frame sync interrupt enable bit. 0h = Interrupt is disabled. An unexpected transmit frame sync interrupt does not generate a McASP transmit interrupt (XINT). 1h = Interrupt is enabled. An unexpected transmit frame sync interrupt generates a McASP transmit interrupt (XINT).

Table 24-42. MCASP_XINTCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	XUNDRN	R/W	0h	Transmitter underrun interrupt enable bit. 0h = Interrupt is disabled. A transmitter underrun interrupt does not generate a McASP transmit interrupt (XINT). 1h = Interrupt is enabled. A transmitter underrun interrupt generates a McASP transmit interrupt (XINT).

24.4.1.32 MCASP_XSTAT Register (offset = C0h) [reset = 0h]

MCASP_XSTAT is shown in Figure 24-70 and described in Table 24-43.

The transmitter status register (XSTAT) provides the transmitter status and transmit TDM time slot number. If the McASP logic attempts to set an interrupt flag in the same cycle that the CPU writes to the flag to clear it, the McASP logic has priority and the flag remains set. This also causes a new interrupt request to be generated.

Figure 24-70. MCASP_XSTAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							XERR
R-0h							R/W-0h

7	6	5	4	3	2	1	0
XDMAERR	XSTAFRM	XDATA	XLAST	XTDMSLOT	XCKFAIL	XSYNCERR	XUNDRN
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

Table 24-43. MCASP_XSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	XERR	R/W	0h	XERR bit always returns a logic-OR of: XUNDRN OR XSYNCERR OR XCKFAIL OR XDMAERR. Allows a single bit to be checked to determine if a transmitter error interrupt has occurred. 0h = No errors have occurred. 1h = An error has occurred.
7	XDMAERR	R/W1C	0h	Transmit DMA error flag. XDMAERR is set when the CPU or DMA writes more serializers through the data port in a given time slot than were programmed as transmitters. Causes a transmit interrupt (XINT), if this bit is set and XDMAERR in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect. 0h = Transmit DMA error did not occur. 1h = Transmit DMA error did occur.
6	XSTAFRM	R/W1C	0h	Transmit start of frame flag. Causes a transmit interrupt (XINT), if this bit is set and XSTAFRM in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect. 0h = No new transmit frame sync (AFSX) is detected. 1h = A new transmit frame sync (AFSX) is detected.
5	XDATA	R/W1C	0h	Transmit data ready flag. Causes a transmit interrupt (XINT), if this bit is set and XDATA in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect. 0h = XBUF is written and is full. 1h = Data is copied from XBUF to XRSR. XBUF is empty and ready to be written. XDATA is also set when the transmit serializers are taken out of reset. When XDATA is set, it always causes a DMA event (AXEVT).

Table 24-43. MCASP_XSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	XLAST	R/W1C	0h	Transmit last slot flag. XLAST is set along with XDATA, if the current slot is the last slot in a frame. Causes a transmit interrupt (XINT), if this bit is set and XLAST in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect. 0h = Current slot is not the last slot in a frame. 1h = Current slot is the last slot in a frame. XDATA is also set.
3	XTDMSLOT	R	0h	Returns the LSB of XSLOT. Allows a single read of XSTAT to determine whether the current TDM time slot is even or odd. 0h = Current TDM time slot is odd. 1h = Current TDM time slot is even.
2	XCKFAIL	R/W1C	0h	Transmit clock failure flag. XCKFAIL is set when the transmit clock failure detection circuit reports an error. Causes a transmit interrupt (XINT), if this bit is set and XCKFAIL in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect. 0h = Transmit clock failure did not occur. 1h = Transmit clock failure did occur.
1	XSYNCERR	R/W1C	0h	Unexpected transmit frame sync flag. XSYNCERR is set when a new transmit frame sync (AFSX) occurs before it is expected. Causes a transmit interrupt (XINT), if this bit is set and XSYNCERR in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect. 0h = Unexpected transmit frame sync did not occur. 1h = Unexpected transmit frame sync did occur.
0	XUNDRN	R/W1C	0h	Transmitter underrun flag. XUNDRN is set when the transmit serializer is instructed to transfer data from XBUF to XRSR, but XBUF has not yet been serviced with new data since the last transfer. Causes a transmit interrupt (XINT), if this bit is set and XUNDRN in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect. 0h = Transmitter underrun did not occur. 1h = Transmitter underrun did occur. For details on McASP action upon underrun conditions, see Buffer Underrun Error - Transmitter.

24.4.1.33 MCASP_XSLOT Register (offset = C4h) [reset = 0h]

MCASP_XSLOT is shown in Figure 24-71 and described in Table 24-44.

The current transmit TDM time slot register (XSLOT) indicates the current time slot for the transmit data frame.

Figure 24-71. MCASP_XSLOT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="22">RESERVED</td><td colspan="10">XSLOTCNT</td></tr>
<tr><td colspan="22">R-0h</td><td colspan="10">R-0h</td></tr>
</table>

Table 24-44. MCASP_XSLOT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9-0	XSLOTCNT	R	0h	Current transmit time slot count. Legal values: 0 to 383 (17Fh). During reset, this counter value is 383 so the next count value, which is used to encode the first DIT group of data, will be 0 and encodes the B preamble. TDM function is not supported for >32 time slots. However, TDM time slot counter may count to 383 when used to transmit a DIT block.

24.4.1.34 MCASP_XCLKCHK Register (offset = C8h) [reset = 0h]

MCASP_XCLKCHK is shown in Figure 24-72 and described in Table 24-45.

The transmit clock check control register (XCLKCHK) configures the transmit clock failure detection circuit.

Figure 24-72. MCASP_XCLKCHK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
XCNT								XMAX							
R-0h								R/W-0h							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XMIN								RESERVED				XPS			
R/W-0h								R-0h				R/W-0h			

Table 24-45. MCASP_XCLKCHK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	XCNT	R	0h	Transmit clock count value (from previous measurement). The clock circuit continually counts the number of system clocks for every 32 transmit high-frequency master clock (AHCLKX) signals, and stores the count in XCNT until the next measurement is taken.
23-16	XMAX	R/W	0h	Transmit clock maximum boundary. This 8 bit unsigned value sets the maximum allowed boundary for the clock check counter after 32 transmit high-frequency master clock (AHCLKX) signals have been received. If the current counter value is greater than XMAX after counting 32 AHCLKX signals, XCKFAIL in XSTAT is set. The comparison is performed using unsigned arithmetic.
15-8	XMIN	R/W	0h	Transmit clock minimum boundary. This 8 bit unsigned value sets the minimum allowed boundary for the clock check counter after 32 transmit high-frequency master clock (AHCLKX) signals have been received. If XCNT is less than XMIN after counting 32 AHCLKX signals, XCKFAIL in XSTAT is set. The comparison is performed using unsigned arithmetic.
7-4	RESERVED	R	0h	
3-0	XPS	R/W	0h	Transmit clock check prescaler value. Fh = Reserved from 9h to Fh. 0h = McASP system clock divided by 1. 1h = McASP system clock divided by 2. 2h = McASP system clock divided by 4. 3h = McASP system clock divided by 8. 4h = McASP system clock divided by 16. 5h = McASP system clock divided by 32. 6h = McASP system clock divided by 64. 7h = McASP system clock divided by 128. 8h = McASP system clock divided by 256. 9h = Reserved from 9h to Fh.

24.4.1.35 MCASP_XEVTCTL Register (offset = CCh) [reset = 0h]

MCASP_XEVTCTL is shown in Figure 24-73 and described in Table 24-46.

The transmitter DMA event control register (XEVTCTL) contains a disable bit for the transmit DMA event. Note for device-specific registers: Accessing XEVTCTL not implemented on a specific device may cause improper device operation.

Figure 24-73. MCASP_XEVTCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							XDATDMA
R-0h							R/W-0h

Table 24-46. MCASP_XEVTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	XDATDMA	R/W	0h	Transmit data DMA request enable bit. If writing to this bit, always write the default value of 0. 0h = Transmit data DMA request is enabled. 1h = Reserved

24.4.1.36 MCASP_DITCSRA0 to MCASP_DITCSRA5 Register (offset = 100h to 114h) [reset = 0h]

MCASP_DITCSRA0 to MCASP_DITCSRA5 is shown in Figure 24-74 and described in Table 24-47.

The DIT left channel status registers (DITCSRA) provide the status of each left channel (even TDM time slot). Each of the six 32-bit registers can store 192 bits of channel status data for a complete block of transmission. The DIT reuses the same data for the next block. It is your responsibility to update the register file in time, if a different set of data need to be sent.

Figure 24-74. MCASP_DITCSRA0 to MCASP_DITCSRA5 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">DITCSRA</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 24-47. MCASP_DITCSRA0 to MCASP_DITCSRA5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DITCSRA	R/W	0h	DIT left channel status registers.

24.4.1.37 MCASP_DITCSRB0 to MCASP_DITCSRB5 Register (offset = 118h to 12Ch) [reset = 0h]

MCASP_DITCSRB0 to MCASP_DITCSRB5 is shown in Figure 24-75 and described in Table 24-48.

The DIT right channel status registers (DITCSRB) provide the status of each right channel (odd TDM time slot). Each of the six 32-bit registers can store 192 bits of channel status data for a complete block of transmission. The DIT reuses the same data for the next block. It is your responsibility to update the register file in time, if a different set of data need to be sent.

Figure 24-75. MCASP_DITCSRB0 to MCASP_DITCSRB5 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">DITCSRB</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 24-48. MCASP_DITCSRB0 to MCASP_DITCSRB5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DITCSRB	R/W	0h	DIT right channel status registers.

24.4.1.38 MCASP_DITUDRA0 to MCASP_DITUDRA5 Register (offset = 130h to 144h) [reset = 0h]

MCASP_DITUDRA0 to MCASP_DITUDRA5 is shown in Figure 24-76 and described in Table 24-49.

The DIT left channel user data registers (DITUDRA) provides the user data of each left channel (even TDM time slot). Each of the six 32-bit registers can store 192 bits of user data for a complete block of transmission. The DIT reuses the same data for the next block. It is your responsibility to update the register in time, if a different set of data need to be sent.

Figure 24-76. MCASP_DITUDRA0 to MCASP_DITUDRA5 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">DITUDRA</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 24-49. MCASP_DITUDRA0 to MCASP_DITUDRA5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DITUDRA	R/W	0h	DIT left channel user data registers.

24.4.1.39 MCASP_DITUDRB0 to MCASP_DITUDRB5 Register (offset = 148h to 15Ch) [reset = 0h]

MCASP_DITUDRB0 to MCASP_DITUDRB5 is shown in Figure 24-77 and described in Table 24-50.

The DIT right channel user data registers (DITUDRB) provides the user data of each right channel (odd TDM time slot). Each of the six 32-bit registers can store 192 bits of user data for a complete block of transmission. The DIT reuses the same data for the next block. It is your responsibility to update the register in time, if a different set of data need to be sent.

Figure 24-77. MCASP_DITUDRB0 to MCASP_DITUDRB5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DITUDRB																															
R/W-0h																															

Table 24-50. MCASP_DITUDRB0 to MCASP_DITUDRB5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DITUDRB	R/W	0h	DIT right channel user data registers.

24.4.1.40 MCASP_SRCTL0 to MCASP_SRCTL5 Register (offset = 180h to 194h) [reset = 0h]

MCASP_SRCTL0 to MCASP_SRCTL5 is shown in Figure 24-78 and described in Table 24-51.

Each serializer on the McASP has a serializer control register (SRCTL). There are up to 16 serializers per McASP. Note for device-specific registers: Accessing SRCTLn not implemented on a specific device may cause improper device operation.

Figure 24-78. MCASP_SRCTL0 to MCASP_SRCTL5 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED		RRDY	XRDY	DISMOD		SRMOD	
R-0h		R-0h	R-0h	R/W-0h		R/W-0h	

Table 24-51. MCASP_SRCTL0 to MCASP_SRCTL5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5	RRDY	R	0h	Receive buffer ready bit. RRDY indicates the current receive buffer state. Always reads 0 when programmed as a transmitter or as inactive. If SRMOD bit is set to receive (2h), RRDY switches from 0 to 1 whenever data is transferred from XRSR to RBUF. 0h = Receive buffer (RBUF) is empty. 1h = Receive buffer (RBUF) contains data and needs to be read before the start of the next time slot or a receiver overrun occurs.
4	XRDY	R	0h	Transmit buffer ready bit. XRDY indicates the current transmit buffer state. Always reads 0 when programmed as a receiver or as inactive. If SRMOD bit is set to transmit (1h), XRDY switches from 0 to 1 when XSRCLR in GBLCTL is switched from 0 to 1 to indicate an empty transmitter. XRDY remains set until XSRCLR is forced to 0, data is written to the corresponding transmit buffer, or SRMOD bit is changed to receive (2h) or inactive (0). 0h = Transmit buffer (XBUF) contains data. 1h = Transmit buffer (XBUF) is empty and needs to be written before the start of the next time slot or a transmit underrun occurs.
3-2	DISMOD	R/W	0h	Serializer pin drive mode bit. Drive on pin when in inactive TDM slot of transmit mode or when serializer is inactive. This field only applies if the pin is configured as a McASP pin (PFUNC = 0). 0h = Drive on pin is 3-state. 1h = Reserved. 2h = Drive on pin is logic low. 3h = Drive on pin is logic high.

Table 24-51. MCASP_SRCTL0 to MCASP_SRCTL5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	SRMOD	R/W	0h	Serializer mode bit. 0h = Serializer is inactive. 1h = Serializer is transmitter. 2h = Serializer is receiver. 3h = Reserved.

24.4.1.41 MCASP_XBUF0 to MCASP_XBUF5 Register (offset = 200h to 214h) [reset = 0h]

MCASP_XBUF0 to MCASP_XBUF5 is shown in Figure 24-79 and described in Table 24-52.

The transmit buffers for the serializers (XBUF) hold data from the transmit format unit. For transmit operations, the XBUF is an alias of the XRBUF in the serializer. Accessing XBUF registers not implemented on a specific device may cause improper device operation.

Figure 24-79. MCASP_XBUF0 to MCASP_XBUF5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XBUF																															
R/W-0h																															

Table 24-52. MCASP_XBUF0 to MCASP_XBUF5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	XBUF	R/W	0h	Transmit buffers for serializers.

24.4.1.42 MCASP_RBUF0 to MCASP_RBUF5 Register (offset = 280h to 294h) [reset = 0h]

MCASP_RBUF0 to MCASP_RBUF5 is shown in Figure 24-80 and described in Table 24-53.

The receive buffers for the serializers (RBUF) hold data from the serializer before the data goes to the receive format unit. For receive operations, the RBUF is an alias of the XRBUF in the serializer. Accessing XBUF registers not implemented on a specific device may cause improper device operation.

Figure 24-80. MCASP_RBUF0 to MCASP_RBUF5 Register

31–0
RBUF
R/W-0h

Table 24-53. MCASP_RBUF0 to MCASP_RBUF5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RBUF	R/W	0h	Receive buffers for serializers.

24.4.1.43 MCASP_WFIFOCTL Register (offset = 1000h) [reset = 0h]

MCASP_WFIFOCTL is shown in Figure 24-81 and described in Table 24-54.

The WNUMEVT and WNUMDMA values must be set prior to enabling the Write FIFO. If the Write FIFO is to be enabled, it must be enabled prior to taking the McASP out of reset

Figure 24-81. MCASP_WFIFOCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							WENA
R-0h							R/W-0h

15	14	13	12	11	10	9	8
WNUMEVT							
R/W-0h							

7	6	5	4	3	2	1	0
WNUMDMA							
R/W-0h							

Table 24-54. MCASP_WFIFOCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	WENA	R/W	0h	Write FIFO enable bit. 0h = Write FIFO is disabled. The WLVL bit in the Write FIFO status register (WFIFOSTS) is reset to 0 and pointers are initialized, that is, the Write FIFO is flushed. 1h = Write FIFO is enabled. If Write FIFO is to be enabled, it must be enabled prior to taking McASP out of reset.
15-8	WNUMEVT	R/W	0h	Write word count per DMA event (32 bit). When the Write FIFO has space for at least WNUMEVT words of data, then an AXEVT (transmit DMA event) is generated to the host/DMA controller. This value should be set to a non-zero integer multiple of the number of serializers enabled as transmitters. This value must be set prior to enabling the Write FIFO. 40h = 3 to 64 words from 3h to 40h. FFh = Reserved from 41h to FFh. 0h = 0 words 1h = 1 word 2h = 2 words 3h = 3 to 64 words from 3h to 40h. 41h = Reserved from 41h to FFh.
7-0	WNUMDMA	R/W	0h	Write word count per transfer (32 bit words). Upon a transmit DMA event from the McASP, WNUMDMA words are transferred from the Write FIFO to the McASP. This value must equal the number of McASP serializers used as transmitters. This value must be set prior to enabling the Write FIFO. FFh = Reserved from 11h to FFh. 0h = 0 words 1h = 1 word 2h = 2 words 3h = 3-16 words from 3h to 10h. 11h = Reserved from 11h to FFh.

24.4.1.44 MCASP_WFIFOSTS Register (offset = 1004h) [reset = 0h]

MCASP_WFIFOSTS is shown in Figure 24-82 and described in Table 24-55.

Figure 24-82. MCASP_WFIFOSTS Register

31-8	7-0
RESERVED	WLVL
R-0h	R-0h

Table 24-55. MCASP_WFIFOSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	WLVL	R	0h	Write level (read-only). Number of 32 bit words currently in the Write FIFO. 40h = 3 to 64 words currently in Write FIFO from 3h to 40h. FFh = Reserved from 41h to FFh. 0h = 0 words currently in Write FIFO. 1h = 1 word currently in Write FIFO. 2h = 2 words currently in Write FIFO. 3h = 3 to 64 words currently in Write FIFO from 3h to 40h. 41h = Reserved from 41h to FFh.

24.4.1.45 MCASP_RFIFOCTL Register (offset = 1008h) [reset = 0h]

MCASP_RFIFOCTL is shown in Figure 24-83 and described in Table 24-56.

The RNUMEVT and RNUMDMA values must be set prior to enabling the Read FIFO. If the Read FIFO is to be enabled, it must be enabled prior to taking the McASP out of reset

Figure 24-83. MCASP_RFIFOCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							RENA
R-0h							R/W-0h

15	14	13	12	11	10	9	8
RNUMEVT							
R/W-0h							

7	6	5	4	3	2	1	0
RNUMDMA							
R/W-0h							

Table 24-56. MCASP_RFIFOCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RENA	R/W	0h	Read FIFO enable bit. 0h = Read FIFO is disabled. The RLVL bit in the Read FIFO status register (RFIFOSTS) is reset to 0 and pointers are initialized, that is, the Read FIFO is flushed. 1h = Read FIFO is enabled. If Read FIFO is to be enabled, it must be enabled prior to taking McASP out of reset.
15-8	RNUMEVT	R/W	0h	Read word count per DMA event (32 bit). When the Read FIFO contains at least RNUMEVT words of data, then an AREVT (receive DMA event) is generated to the host/DMA controller. This value should be set to a non-zero integer multiple of the number of serializers enabled as receivers. This value must be set prior to enabling the Read FIFO. 40h = 3 to 64 words from 3h to 40h. FFh = Reserved from 41h = FFh. 0h = 0 words 1h = 1 word 2h = 2 words 3h = 3 to 64 words from 3h to 40h. 41h = Reserved from 41h to FFh.
7-0	RNUMDMA	R/W	0h	Read word count per transfer (32 bit words). Upon a receive DMA event from the McASP, the Read FIFO reads RNUMDMA words from the McASP. This value must equal the number of McASP serializers used as receivers. This value must be set prior to enabling the Read FIFO. 10h = 3 to 16 words from 3h to 10h. FFh = Reserved from 11h to FFh. 0h = 0 words 1h = 1 word 2h = 2 words 3h = 3 to 16 words from 3h to 10h. 11h = Reserved from 11h to FFh.

24.4.1.46 MCASP_RFIFOSTS Register (offset = 100Ch) [reset = 0h]

MCASP_RFIFOSTS is shown in Figure 24-84 and described in Table 24-57.

Figure 24-84. MCASP_RFIFOSTS Register

31-8	7-0
RESERVED	RLVL
R-0h	R-0h

Table 24-57. MCASP_RFIFOSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	RLVL	R	0h	Read level (read-only). Number of 32 bit words currently in the Read FIFO. 40h = 3 to 64 words currently in Read FIFO from 3h to 40h. FFh = Reserved from 41h to FFh. 0h = 0 words currently in Read FIFO. 1h = 1 word currently in Read FIFO. 2h = 2 words currently in Read FIFO. 3h = 3 to 64 words currently in Read FIFO from 3h to 40h. 41h = Reserved from 41h to FFh.

Controller Area Network (CAN)

This chapter describes the controller area network (CAN) for the device.

Topic **Page**

25.1 Introduction

25.1.1 DCAN Features

The general features of the DCAN controller are:

- Supports CAN protocol version 2.0 part A, B (ISO 11898-1)
- Bit rates up to 1 MBit/s
- Dual clock source
- 16, 32, 64 or 128 message objects (instantiated as 64 on this device)
- Individual identifier mask for each message object
- Programmable FIFO mode for message objects
- Programmable loop-back modes for self-test operation
- Suspend mode for debug support
- Software module reset
- Automatic bus on after Bus-Off state by a programmable 32-bit timer
- Message RAM parity check mechanism
- Direct access to Message RAM during test mode
- CAN Rx / Tx pins configurable as general purpose IO pins
- Two interrupt lines (plus additional parity-error interrupt line)
- RAM initialization
- DMA support

25.1.2 Unsupported DCAN Features

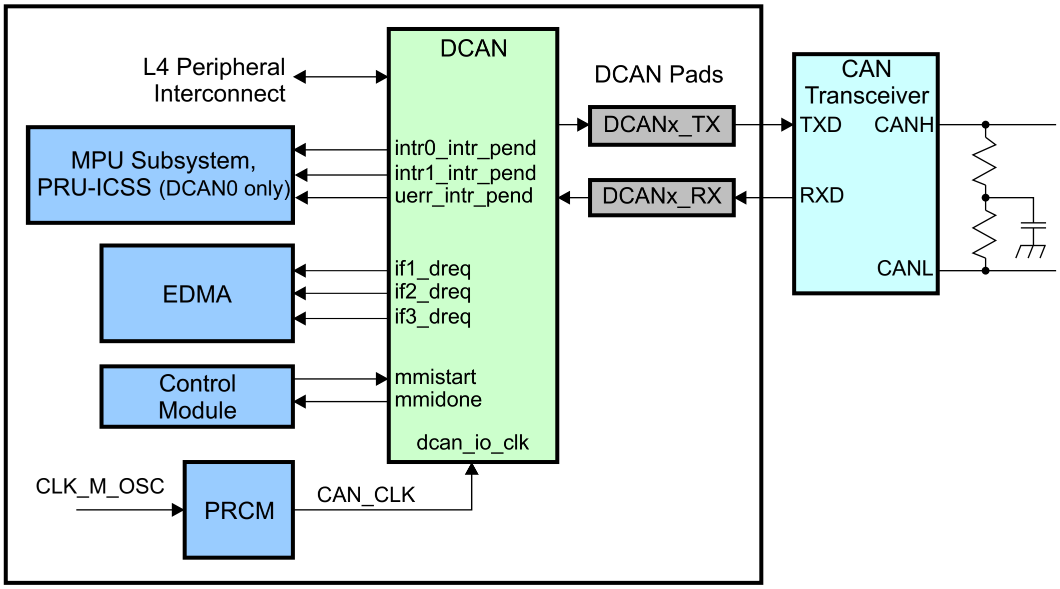
The DCAN module in this device does not support GPIO pin mode. All GPIO functionality is mapped through the GPIO modules and muxed at the pins. GPIO pin control signals from the DCAN modules are not connected.

25.2 Integration

The Controller Area Network is a serial communications protocol which efficiently supports distributed realtime control with a high level of security. The DCAN module supports bitrates up to 1 Mbit/s and is compliant to the CAN 2.0B protocol specification. The core IP within DCAN is provided by Bosch.

This device includes two instantiations of the DCAN controller: DCAN0 and DCAN1. Figure 25-1 shows the DCAN module integration.

Figure 25-1. DCAN Integration



25.2.1 DCAN Connectivity Attributes

The general connectivity attributes for the DCAN module are shown in Table 25-1.

Table 25-1. DCAN Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L4LS_GCLK (OCP) PD_PER_CAN_CLK (Func)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	3 Interrupts per instance Intr0 (DCANx_INT0) – Error, Status, Msg Object interrupt Intr1 (DCANx_INT1) – Msg Object interrupt Uerr (DCANx_PARITY) – Parity error interrupt All DCAN0 interrupts to MPU Subsystem and PRU-ICSS All DCAN1 interrupts to only MPU Subsystem
DMA Requests	3 DMA requests per instance to EDMA (CAN_IFxDMA)
Physical Address	L4 Peripheral slave port

25.2.2 DCAN Clock and Reset Management

The DCAN controllers have separate bus interface and functional clocks.

Table 25-2. DCAN Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
DCAN_ocp_clk Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk from PRCM
DCAN_io_clk Functional clock	26 MHz	CLK_M_OSC	pd_per_can_clk from PRCM

25.2.3 DCAN Pin List

The external signals for the DCAN module are shown in the following table.

Table 25-3. DCAN Pin List

Pin	Type	Description
DCAN*x*_TX	O	DCAN transmit line
DCAN*x*_RX	I	DCAN receive line

25.3 Functional Description

The DCAN module performs CAN protocol communication according to ISO 11898-1. The bit rate can be programmed to values up to 1 MBit/s. Additional transceiver hardware is required for the connection to the physical layer (CAN bus).

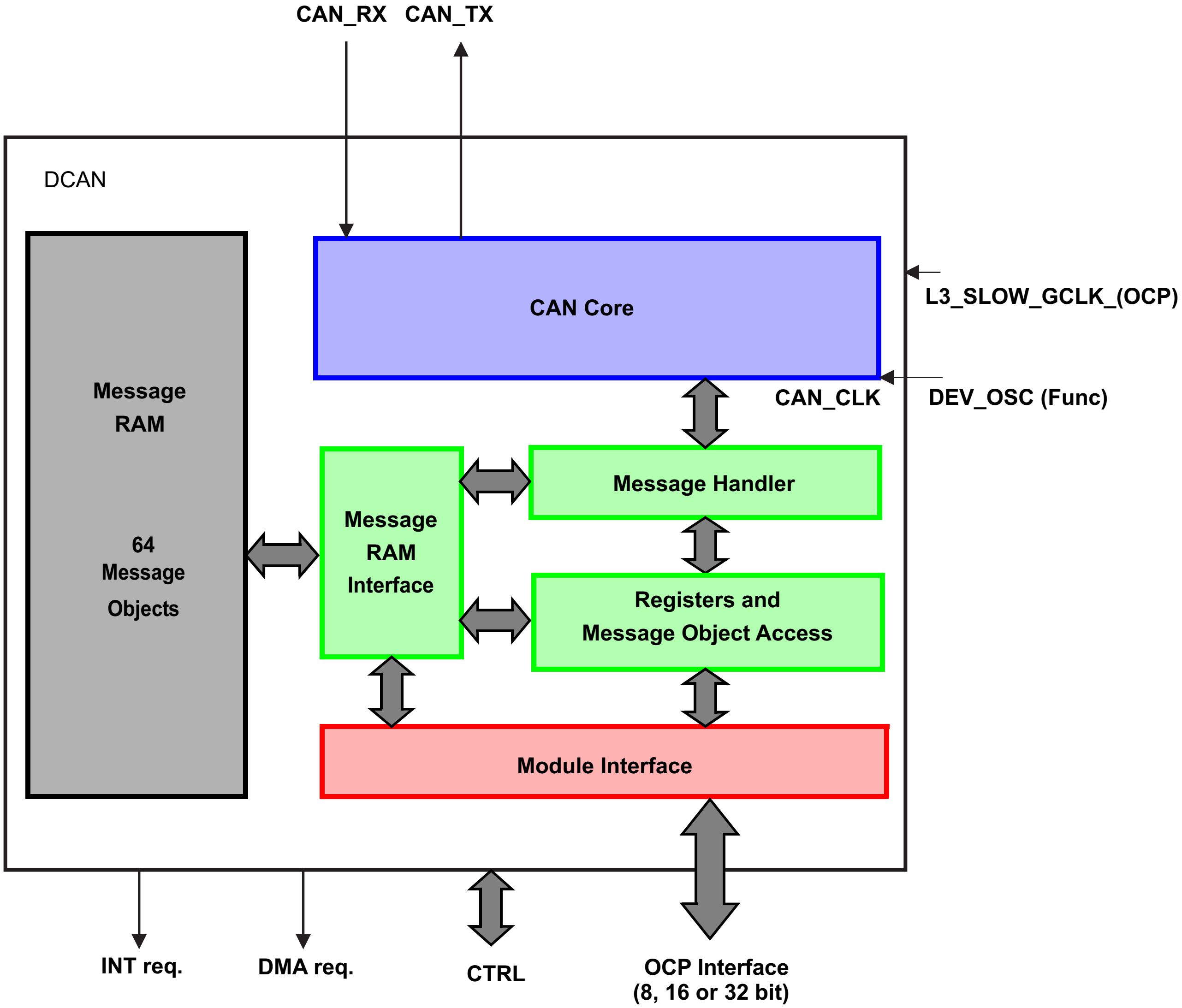
For communication on a CAN network, individual message objects can be configured. The message objects and identifier masks are stored in the message RAM.

All functions concerning the handling of messages are implemented in the message handler. Those functions are acceptance filtering, the transfer of messages between the CAN core and the message RAM, and the handling of transmission requests, as well as the generation of interrupts or DMA requests.

The register set of the DCAN module can be accessed directly by the CPU via the module interface. These registers are used to control and configure the CAN core and the message handler, and to access the message RAM.

Figure 25-2 shows the DCAN block diagram and its features are described below.

Figure 25-2. DCAN Block Diagram



25.3.1 CAN Core

The CAN core consists of the CAN protocol controller and the Rx/Tx shift register. It handles all ISO 11898-1 protocol functions.

25.3.2 Message Handler

The message handler is a state machine that controls the data transfer between the single-ported message RAM and the CAN core’s Rx/Tx shift register. It also handles acceptance filtering and the interrupt/DMA request generation as programmed in the control registers.

25.3.3 Message RAM

The DCAN0 and DCAN1 enables a storage of 64 CAN messages.

25.3.4 Message RAM Interface

Three interface register sets control the CPU read and write accesses to the message RAM. There are two interface registers sets for read and write access, IF1 and IF2, and one interface register set for read access only, IF3. Additional information can be found in Section 25.3.15.12.

The interface registers have the same word-length as the message RAM.

25.3.5 Registers and Message Object Access

Data consistency is ensured by indirect accesses to the message objects. During normal operation, all CPU and DMA accesses to the message RAM are done through interface registers. In a dedicated test mode, the message RAM is memory mapped and can be directly accessed by either CPU or DMA.

25.3.6 Module Interface

The DCAN module registers are accessed by the CPU or user software through a 32-bit peripheral bus interface.

25.3.7 Dual Clock Source

Two clock domains are provided to the DCAN module: the peripheral synchronous clock domain (L3_SLOW_GCLK) and the peripheral asynchronous clock source domain (CLK_M_OSC) for CAN_CLK.

25.3.8 CAN Operation

After hardware reset, the Init bit in the CAN control register (CTL) is set and all CAN protocol functions are disabled. The CAN module must be initialized before operating it. Figure 25-3 illustrates the basic initialization flow for the CAN module.

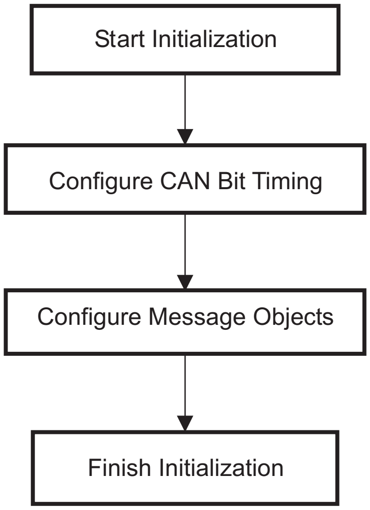
25.3.8.1 CAN Module Initialization

A general CAN module initialization would mean the following two critical steps:

- Configuration of the CAN bit timing
- Configuration of message objects

To initialize the CAN controller, the CPU has to set up the CAN bit timing and those message objects that have to be used for CAN communication. Message objects that are not needed, can be deactivated.

Figure 25-3. CAN Module General Initialization Flow



25.3.8.1.1 Configuration of CAN Bit Timing

The CAN module must be in initialization mode to configure the CAN bit timing.

For CAN bit timing software configuration flow, see Figure 25-4.

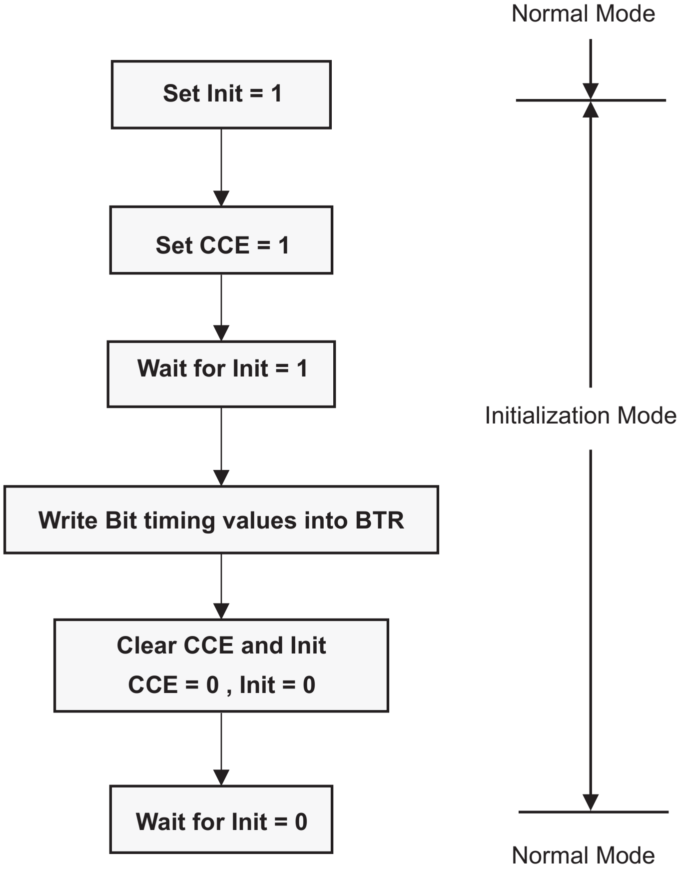
Step 1: Enter *initialization mode* by setting the Init (Initialization) bit in the CAN control register.

While the Init bit is set, the message transfer from and to the CAN bus is stopped, and the status of the CAN_TX output is recessive (high).

The CAN error counters are not updated. Setting the Init bit does not change any other configuration register.

Also, note that the CAN module is in initialization mode on hardware reset and during Bus-Off.

Figure 25-4. CAN Bit-Timing Configuration



Step 2: Set the Configure Change Enable (CCE) bit in the CAN control register.

The access to the Bit Timing register (BTR) for the configuration of the bit timing is enabled when both the Init and CCE bits in the CAN Control register are set.

Step 3: Wait for the Init bit to get set. This would make sure that the module has entered *Initialization mode*.

Step 4: Write the bit timing values into the bit timing register. See Section 25.3.16.2 for the BTR value calculation for a given bit timing.

Step 5: Clear the CCE and Init bit.

Step 6: Wait for the Init bit to clear. This would ensure that the module has come out of *initialization mode*.

Following these steps, the module comes to operation by synchronizing itself to the CAN bus, provided the BTR is configured as per the CAN bus baud rate, although the message objects have to be configured before carrying out any communication.

NOTE: The module will not come out of the *initialization mode* if any incorrect BTR values are written in step 4.

NOTE: The required message objects should be configured as transmit or receive objects before the start of data transfer as explained in Section 25.3.8.1.

25.3.8.1.2 Configuration of Message Objects

The message objects can be configured only through the interface registers; the CPU does not have direct access to the message object (message RAM) . Familiarize yourself with the interface register set (IFx) usage (see Section 25.3.17) and the message object structure (see Section 25.3.18) before configuring the message objects.

For more information regarding the procedure to configure the message objects, see Section 25.3.14. All the message objects should be configured to particular identifiers or set to not valid before the message transfer is started. It is possible to change the configuration of message objects during normal operation (that is between data transfers).

NOTE: The message objects initialization is independent of the bit-timing configuration.

25.3.8.1.3 DCAN RAM Hardware Initialization

The memory hardware initialization for the DCAN module is enabled in the device control register (DCAN_RAMINIT) which initializes the RAM with zeros and sets parity bits accordingly. Wait for the RAMINIT_DONE bit to be set to ensure successful RAM initialization. Ensure the clock to the DCAN module is enabled before starting this initialization.

For more details on RAM hardware initialization, see Chapter 7, *Control Module*.

25.3.8.2 CAN Message Transfer (Normal Operation)

Once the DCAN is initialized and the Init bit is reset to zero, the CAN core synchronizes itself to the CAN bus and is ready for message transfer as per the configured message objects.

The CPU may enable the interrupt lines (setting IE0 and IE1 to ‘1’) at the same time when it clears Init and CCE. The status interrupts EIE and SIE may be enabled simultaneously.

The CAN communication can be carried out in any of the following two modes: interrupt and polling.

The interrupt register points to those message objects with IntPnd = ‘1’. It is updated even if the interrupt lines to the CPU are disabled (IE0/IE1 are zero).

The CPU may poll all MessageObject’s NewDat and TxRqst bits in parallel from the NewData X registers and the Transmission Request X Registers (TXRQ X). Polling can be made easier if all transmit objects are grouped at the low numbers and all receive objects are grouped at the high numbers.

Received messages are stored into their appropriate message objects if they pass acceptance filtering.

The whole message (including all arbitration bits, DLC and up to eight data bytes) is stored into the message object. As a consequence (e.g., when the identifier mask is used), the arbitration bits which are masked to “don’t care” may change in the message object when a received message is stored.

The CPU may read or write each message at any time via the interface registers, as the message handler guarantees data consistency in case of concurrent accesses.

If a permanent message object (arbitration and control bits set up during configuration and leaving unchanged for multiple CAN transfers) exists for the message, it is possible to only update the data bytes.

If several transmit messages should be assigned to one message object, the whole message object has to be configured before the transmission of this message is requested.

The transmission of multiple message objects may be requested at the same time. They are subsequently transmitted, according to their internal priority.

Messages may be updated or set to not valid at any time, even if a requested transmission is still pending. However, the data bytes will be discarded if a message is updated before a pending transmission has started.

Depending on the configuration of the message object, a transmission may be automatically requested by the reception of a remote frame with a matching identifier.

25.3.8.2.1 Automatic Retransmission

According to the CAN Specification (ISO11898), the DCAN provides a mechanism to automatically retransmit frames which have lost arbitration or have been disturbed by errors during transmission. The frame transmission service will not be confirmed to the user before the transmission is successfully completed.

By default, this automatic retransmission is enabled. It can be disabled by setting the disable automatic retransmission (DAR) bit in the CTL register. Further details to this mode are provided in Section 25.3.15.3.

25.3.8.2.2 Auto-Bus-On

By default, after the DCAN has entered Bus-Off state, the CPU can start a Bus-Off-Recovery sequence by resetting the Init bit. If this is not done, the module will stay in Bus-Off state.

The DCAN provides an automatic Auto-Bus-On feature which is enabled by bit ABO in the CTL register. If set, the DCAN module will automatically start the Bus-Off-Recovery sequence. The sequence can be delayed by a user-defined number of L3_SLOW_GCLK cycles which can be defined in the Auto-Bus-On Time register (ABOTR).

NOTE: If the DCAN goes to Bus-Off state due to a massive occurrence of CAN bus errors, it stops all bus activities and automatically sets the Init bit. Once the Init bit has been reset by the CPU or due to the Auto-Bus-On feature, the device will wait for 129 occurrences of bus Idle (equal to 129 * 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus-Off recovery sequence, the error counters will be reset.

25.3.8.3 Test Modes

The DCAN module provides several test modes which are mainly intended for production tests or self test.

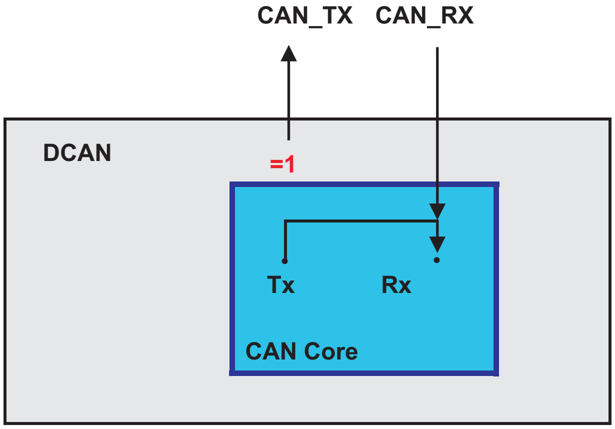
For all test modes, the Test bit in the CTL register needs to be set to one. This enables write access to the test register (TEST).

25.3.8.3.1 Silent Mode

The silent mode may be used to analyze the traffic on the CAN bus without affecting it by sending dominant bits (e.g., acknowledge bit, overload flag, active error flag). The DCAN is still able to receive valid data frames and valid remote frames, but it will not send any dominant bits. However, these are internally routed to the CAN core.

Figure 25-5 shows the connection of signals CAN_TX and CAN_RX to the CAN core in silent mode. Silent mode can be activated by setting the Silent bit in the TEST register to one. In ISO 11898-1, the silent mode is called the bus monitoring mode.

Figure 25-5. CAN Core in Silent Mode



25.3.8.3.2 Loopback Mode

The loopback mode is mainly intended for hardware self-test functions. In this mode, the CAN core uses internal feedback from Tx output to Rx input. Transmitted messages are treated as received messages, and can be stored into message objects if they pass acceptance filtering. The actual value of the CAN_RX input pin is disregarded by the CAN core. Transmitted messages can still be monitored at the CAN_TX pin.

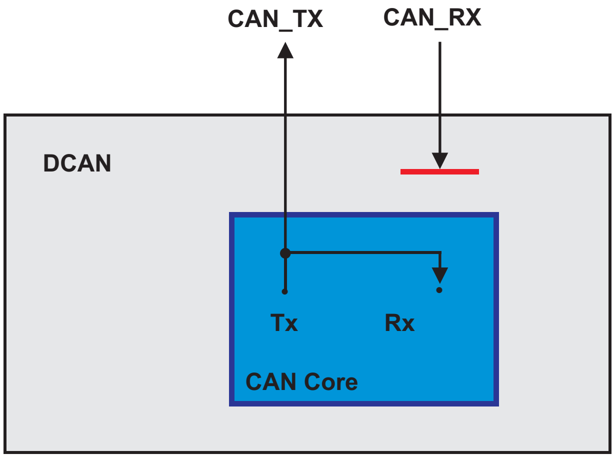
In order to be independent from external stimulation, the CAN core ignores acknowledge sampled in the acknowledge slot of a data/remote frame.

Figure 25-6 shows the connection of signals CAN_TX and CAN_RX to the CAN core in loopback mode.

Loopback mode can be activated by setting bit LBack in the TEST register to one.

NOTE: In loopback mode, the signal path from CAN core to Tx pin, the Tx pin itself, and the signal path from Tx pin back to CAN core are disregarded. For including these into the testing, see Section 25.3.8.3.3.

Figure 25-6. CAN Core in Loopback Mode



25.3.8.3.3 External Loopback Mode

The external loopback mode is similar to the loopback mode; however, it includes the signal path from CAN core to Tx pin, the Tx pin itself, and the signal path from Tx pin back to CAN core. When external loopback mode is selected, the input of the CAN core is connected to the input buffer of the Tx pin.

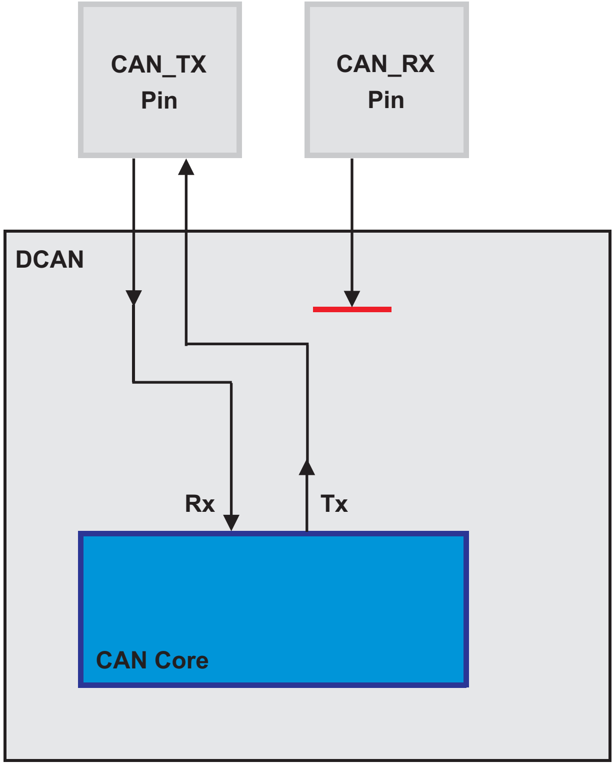
With this configuration, the Tx pin IO circuit can be tested.

External loopback mode can be activated by setting bit **EXL** in the TEST register to one.

Figure 25-7 shows the connection of signals CAN_TX and CAN_RX to the CAN core in external loopback mode.

NOTE: When loopback mode is active (LBack bit set), the ExL bit will be ignored.

Figure 25-7. CAN Core in External Loopback Mode

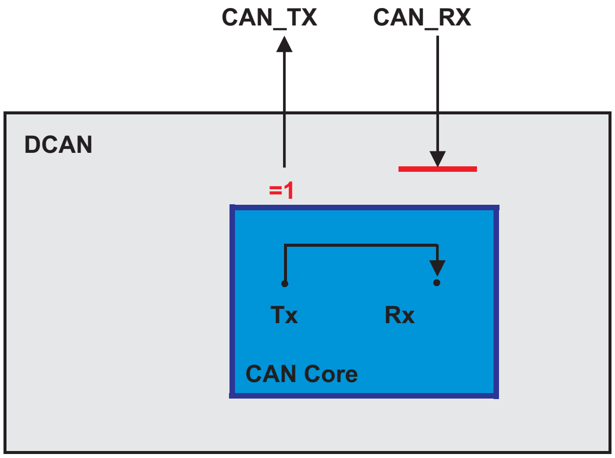


25.3.8.3.4 Loopback Mode Combined With Silent Mode

It is also possible to combine loopback mode and silent mode by setting bits LBack and Silent at the same time. This mode can be used for a “Hot Selftest,” that is, the DCAN hardware can be tested without affecting the CAN network. In this mode, the CAN_RX pin is disconnected from the CAN core and no dominant bits will be sent on the CAN_TX pin.

Figure 25-8 shows the connection of the signals CAN_TX and CAN_RX to the CAN core in case of the combination of loopback mode with silent mode.

Figure 25-8. CAN Core in Loop Back Combined With Silent Mode



25.3.8.3.5 Software Control of CAN_TX pin

Four output functions are available for the CAN transmit pin CAN_TX. In addition to its default function (serial data output), the CAN_TX pin can drive constant dominant or recessive values, or it can drive the CAN sample point signal to monitor the CAN core's bit timing.

Combined with the readable value of the CAN_RX pin, this function can be used to check the physical layer of the CAN bus.

The output mode of pin CAN_TX is selected by programming the TEST register bits Tx[1:0].

NOTE: The software control for the CAN_TX pin interferes with CAN protocol functions. For CAN message transfer or any of the test modes (loopback mode, external loopback mode or silent mode), the CAN_TX pin should operate in its default functionality.

25.3.9 Dual Clock Source

Two clock domains are provided to the DCAN module: the peripheral synchronous clock domain (L3_SLOW_GCLK) as the general module clock source, and the peripheral asynchronous clock source domain (CLK_M_OSC) provided to the CAN core (as clock source CAN_CLK) for generating the CAN bit timing.

Both clock domains can be derived from the same clock source (so that L3_SLOW_GCLK = CLK_M_OSC).

For more information on how to configure the relevant clock source registers in the system module, see Chapter 6, Power and Clock Management.

Between the two clock domains, a synchronization mechanism is implemented in the DCAN module in order to ensure correct data transfer.

NOTE: If the dual clock functionality is used, then L3_SLOW_GCLK must always be higher or equal to CAN_CLK (CLK_M_OSC) (derived from the asynchronous clock source), in order to achieve a stable functionality of the DCAN. Here also the frequency shift of the modulated L3_SLOW_GCLK has to be considered:

$f_{0,\ \mathrm{L3\_SLOW\_GCLK}}(\mathrm{OCP}) \pm \Delta f_{\mathrm{FM},\ \mathrm{L3\_SLOW\_GCLK}}(\mathrm{OCP}) \geq f_{\mathrm{CANCLK}}$

NOTE: The CAN core has to be programmed to at least 8 clock cycles per bit time. To achieve a transfer rate of 1 MBaud when using the asynchronous clock domain as the clock source for CAN_CLK (CLK_M_OSC), an oscillator frequency of 8 MHz or higher has to be used.

25.3.10 Interrupt Functionality

Interrupts can be generated on two interrupt lines: DCANINT0 and DCANINT1. These lines can be enabled by setting the IE0 and IE1 bits, respectively, in the CTL register. The interrupts are level triggered at the chip level.

The DCAN provides three groups of interrupt sources: message object interrupts, status change interrupts, and error interrupts (see Figure 25-9 and Figure 25-10).

The source of an interrupt can be determined by the interrupt identifiers Int0ID/Int1ID in the interrupt register (see INT). When no interrupt is pending, the register will hold the value zero.

Each interrupt line remains active until the dedicated field (Int0ID or Int1ID) in the interrupt register (INT) again reach zero, meaning the cause of the interrupt is reset, or until IE0 or IE1 are reset.

The value 0x8000 in the Int0ID field indicates that an interrupt is pending because the CAN core has updated (not necessarily changed) the Error and Status register (ES). This interrupt has the highest priority. The CPU can update (reset) the status bits WakeUpPnd, RxOk, TxOk and LEC by reading the ES register, but a write access of the CPU will never generate or reset an interrupt.

Values between 1 and the number of the last message object indicates that the source of the interrupt is one of the message objects, Int0ID or Int1ID will point to the pending message interrupt with the highest priority. The Message Object 1 has the highest priority; the last message object has the lowest priority.

An interrupt service routine that reads the message that is the source of the interrupt may read the message and reset the message object’s IntPnd at the same time (ClrIntPnd bit in the IF1CMD or IF2CMD register). When IntPnd is cleared, the interrupt register will point to the next message object with a pending interrupt.

25.3.10.1 Message Object Interrupts

Message object interrupts are generated by events from the message objects. They are controlled by the flags IntPND, TxIE and RxIE that are described in Section 25.3.18.1.

Message object interrupts can be routed to either DCANINT0 or DCANINT1 line, controlled by the interrupt multiplexer register (INTMUX12 to INTMUX78).

25.3.10.2 Status Change Interrupts

The events WakeUpPnd, RxOk, TxOk and LEC in the ES register belong to the status change interrupts. The status change interrupt group can be enabled by bit in CTL register.

If SIE is set, a status change interrupt will be generated at each CAN frame, independent of bus errors or valid CAN communication, and also independent of the message RAM configuration.

Status change interrupts can only be routed to interrupt line DCAN0INT, which has to be enabled by setting the IE0 bit in the CTL register.

NOTE: Reading the error and status register will clear the WakeUpPnd flag. If in global power-down mode, the WakeUpPnd flag is cleared by such a read access before the DCAN module has been waken up by the system, the DCAN may re-assert the WakeUpPnd flag, and a second interrupt may occur.

25.3.10.3 Error Interrupts

The events PER, BOff and EWarn, monitored in the ES register, belong to the error interrupts. The error interrupt group can be enabled by setting the EIE bit in the CTL register.

Error interrupts can only be routed to interrupt line DCAN0INT, which has to be enabled by setting the IE0 bit in the CTL register.

Figure 25-9. CAN Interrupt Topology 1

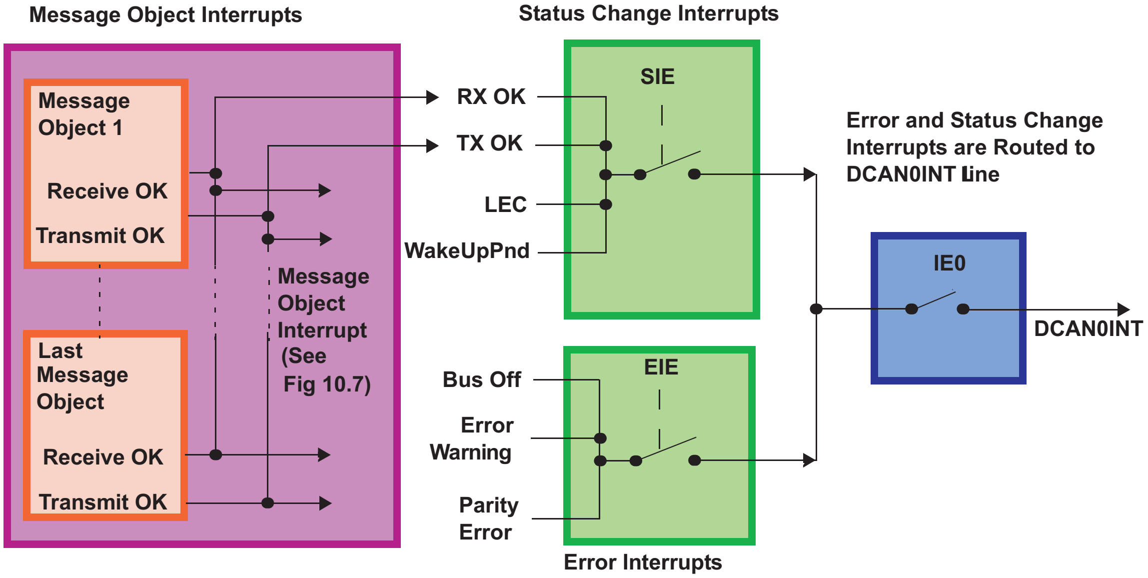
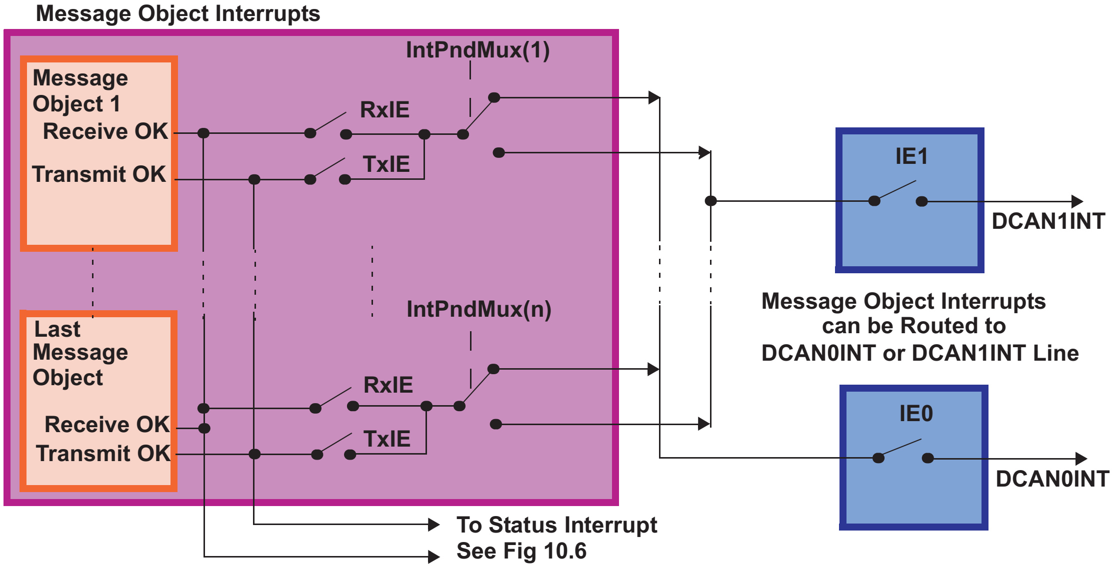


Figure 25-10. CAN Interrupt Topology 2



25.3.11 Local Power-Down Mode

The DCAN supports a local power-down mode, which can be controlled within the CTL register.

25.3.11.1 Entering Local Power-Down Mode

The local power-down mode is requested by setting the PDR bit in the CTL register.

The DCAN then finishes all transmit requests of the message objects. When all requests are done, the DCAN module waits until a bus idle state is recognized. The module will automatically set the Init bit in the CTL register to prevent any further CAN transfers, and it will also set the PDA bit in the CAN error and status register (ES). With setting the PDA bits, the DCAN module indicates that the local power-down mode has been entered.

During local power-down mode, the internal clocks of the DCAN module are turned off, but there is wakeup logic (see Section 25.3.11.2) that can be active, if enabled. Also, the actual contents of the control registers can be read back.

NOTE: In local low-power mode, the application should not clear the Init bit while PDR is set. If there are any messages in the message RAM which are configured as transmit messages and the application resets the init bit, these messages may get sent.

25.3.11.2 Wakeup From Local Power Down

There are two ways to wake up the DCAN from local power-down mode:

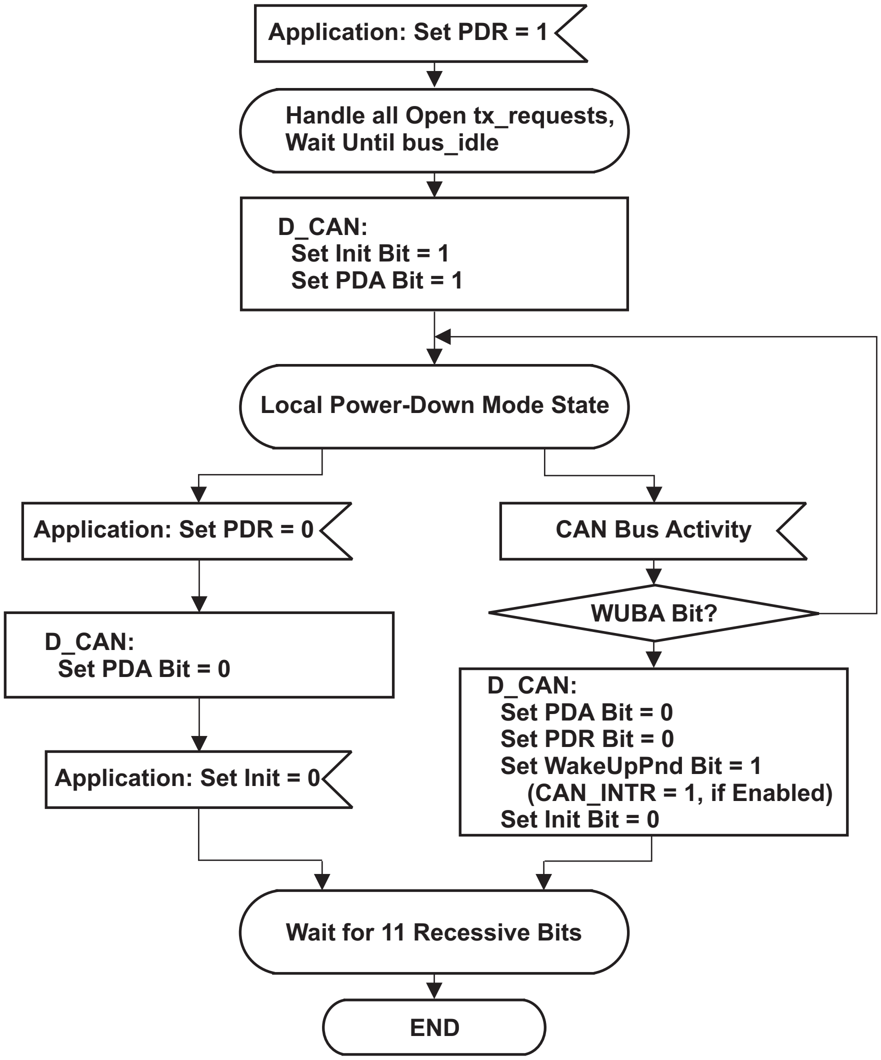
- The application could wake up the DCAN module manually by clearing the PDR bit and then clearing the Init bit in the CTL register.
- Alternatively, a CAN bus activity detection circuit can be activated by setting the wakeup on bus activity (WUBA) bit in the CTL register. If this circuit is active, on occurrence of a dominant CAN bus level, the DCAN will automatically start the wakeup sequence. It will clear the PDR bit in the CTL register and also clear the PDA bit in the error and status register. The WakeUpPnd bit in the ES register will be set. If status interrupts are enabled, also an interrupt will be generated. Finally the Init bit in the CTL register will be cleared.

After the Init bit has been cleared, the module waits until it detects 11 consecutive recessive bits on the CAN_RX pin and then goes bus-active again.

NOTE: The CAN transceiver circuit has to stay active in order to detect any CAN bus activity while the DCAN is in local power down mode. The first CAN message, which initiates the bus activity, cannot be received. This means that the first message received in power-down and automatic wake-up mode, is lost.

Figure 25-11 shows a flow diagram about entering and leaving local power-down mode.

Figure 25-11. Local Power-Down Mode Flow Diagram



25.3.12 Parity Check Mechanism

The DCAN provides a parity check mechanism to ensure data integrity of message RAM data. For each word (32 bits) in message RAM, one parity bit will be calculated. The formation of the different words is according to the message RAM representation in RDA mode, see Section 25.3.18.4.

Parity information is stored in the message RAM on write accesses and will be checked against the stored parity bit from message RAM on read accesses.

The Parity check functionality can be enabled or disabled by PMD bit field in the CTL register.

In case of disabled parity check, the parity bits in message RAM will be left unchanged on write access to data area and no check will be done on read access.

If parity checking is enabled, parity bits will be automatically generated and checked by the DCAN. The parity bits could be read in debug/suspend mode (see Section 25.3.18.3) or in RDA mode (see Section 25.3.18.4). However, direct write access to the parity bits is only possible in these two modes with parity check disabled.

A parity bit will be set, if the modulo-2-sum of the data bits is 1. This definition is equivalent to: The parity bit will be set, if the number of 1 bits in the data is odd.

NOTE: The parity scheme is tied to even parity at the device level.

25.3.12.1 Behavior on Parity Error

On any read access to message RAM (e.g., during start of a CAN frame transmission), the parity of the message object will be checked. If a parity error is detected, the PER bit in the ES register will be set. If error interrupts are enabled, an interrupt would also be generated. In order to avoid the transmission of invalid data over the CAN bus, the D bit of the message object will be reset.

The message object data can be read by the host CPU, independently of parity errors. Thus, the application has to ensure that the read data is valid, for example, by immediately checking the parity error code register (PERR) on parity error interrupt.

NOTE: During RAM initialization, no parity check will be done.

25.3.12.2 Parity Testing

Testing the parity mechanism can be done by enabling the bit RamDirectAccess (RDA) and manually writing the parity bits directly to the dedicated RAM locations. With this, data and parity bits could be checked when reading directly from RAM.

NOTE: If parity check was disabled, the application has to ensure correct parity bit handling in order to prevent parity errors later on when parity check is enabled.

25.3.13 Debug/Suspend Mode

The module supports the usage of an external debug unit by providing functions like pausing DCAN activities and making message RAM content accessible via OCP interface.

Before entering debug/suspend mode, the circuit will either wait until a started transmission or reception will be finished and bus idle state is recognized, or immediately interrupt a current transmission or reception. This is depending on bit IDS in the CTL register.

Afterwards, the DCAN enters debug/suspend mode, indicated by InitDbg flag in the CTL register.

During debug/suspend mode, all DCAN registers can be accessed. Reading reserved bits will return ‘0’. Writing to reserved bits will have no effect.

Also, the message RAM will be memory mapped. This allows the external debug unit to read the message RAM. For the memory organization, see Section 25.3.18.3).

NOTE: During debug/suspend mode, the message RAM cannot be accessed via the IFx register sets.

Writing to control registers in debug/suspend mode may influence the CAN state machine and further message handling.

For debug support, the auto clear functionality of the following DCAN registers is disabled:

- ES register (clear of status flags by read)
- IF1CMD and IF2CMD (clear of DMAActive flag by read/write)

25.3.14 Configuration of Message Objects

The whole message RAM should be configured before the end of the initialization, however it is also possible to change the configuration of message objects during CAN communication.

The CAN software driver library should offer subroutines that:

- Transfer a complete message structure into a message object. (Configuration)
- Transfer the data bytes of a message into a message object and set TxRqst and NewDat. (Start a new transmission)
- Get the data bytes of a message from a message object and clear NewDat (and IntPnd). (Read received data)
- Get the complete message from a message object and clear NewDat (and IntPnd). (Read a received message, including identifier, from a message object with UMask = ‘1’)

Parameters of the subroutines are the Message Number and a pointer to a complete message structure or to the data bytes of a message structure.

Two examples of assigning the IFx interface register sets to these subroutines are shown here:

In the first method, the tasks of the application program that may access the module are divided into two groups. Each group is restricted to the use of one of the interface register sets. The tasks of one group may interrupt tasks of the other group, but not of the same group.

In the second method, which may be a special case of the first method, there are only two tasks in the application program that access the module. A Read_Message task that uses IF2 or IF3 to get received messages from the message RAM and a Write_Message task that uses IF1 to write messages to be transmitted (or to be configured) into the message RAM. Both tasks may interrupt each other.

25.3.14.1 Configuration of a Transmit Object for Data Frames

Table 25-4 shows how a transmit object can be initialized.

Table 25-4. Initialization of a Transmit Object

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	1	0	0	0	appl.	0	appl.	0

The arbitration bits (ID[28:0] and Xtd bit) are given by the application. They define the identifier and type of the outgoing message. If an 11-bit identifier (standard frame) is used (Xtd = ‘0’), it is programmed to ID[28:18]. In this case, ID[17:0] can be ignored.

The data registers (DLC[3:0] and Data0-7) are given by the application. TxRqst and RmtEn should not be set before the data is valid.

If the TxIE bit is set, the IntPnd bit will be set after a successful transmission of the message object.

If the RmtEn bit is set, a matching received remote frame will cause the TxRqst bit to be set; the remote frame will autonomously be answered by a data frame.

The mask bits (Msk[28:0], UMask, MXtd, and MDir bits) may be used (UMask=’1’) to allow groups of remote frames with similar identifiers to set the TxRqst bit. The Dir bit should not be masked. For details, see Section 25.3.15.8. Identifier masking must be disabled (UMask = ‘0’) if no remote frames are allowed to set the TxRqst bit (RmtEn = ‘0’).

25.3.14.2 Configuration of a Transmit Object for Remote Frames

It is not necessary to configure transmit objects for the transmission of remote frames. Setting TxRqst for a receive object causes the transmission of a remote frame with the same identifier as the data frame for which this receive object is configured.

25.3.14.3 Configuration of a Single Receive Object for Data Frames

Table 25-5 shows how a receive object for data frames can be initialized.

Table 25-5. Initialization of a single Receive Object for Data Frames

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	0	0	0	appl.	0	0	0	0

The arbitration bits (ID[28:0] and Xtd bit) are given by the application. They define the identifier and type of accepted received messages. If an 11-bit Identifier (standard frame) is used (Xtd = ‘0’), it is programmed to ID[28:18]. In this case, ID[17:0] can be ignored. When a data frame with an 11-bit Identifier is received, ID[17:0] is set to ‘0’.

The data length code (DLC[3:0]) is given by the application. When the message handler stores a data frame in the message object, it will store the received data length code and eight data bytes. If the data length code is less than 8, the remaining bytes of the message object may be overwritten by non specified values.

The mask bits (Msk[28:0], UMask, MXtd, and MDir bits) may be used (UMask = ’1’) to allow groups of data frames with similar identifiers to be accepted. The Dir bit should not be masked in typical applications. If some bits of the mask bits are set to “don’t care,” the corresponding bits of the arbitration register will be overwritten by the bits of the stored data frame.

If the RxIE bit is set, the IntPnd bit will be set when a received data frame is accepted and stored in the message object.

If the TxRqst bit is set, the transmission of a remote frame with the same identifier as actually stored in the arbitration bits will be triggered. The content of the arbitration bits may change if the mask bits are used (UMask = ’1’) for acceptance filtering.

25.3.14.4 Configuration of a Single Receive Object for Remote Frames

Table 25-6 shows how a receive object for remote frames can be initialized.

Table 25-6. Initialization of a Single Receive Object for Remote Frames

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	1	0	0	appl.	0	0	0	0

A receive object for remote frames may be used to monitor remote frames on the CAN bus. The remote frame stored in the receive object will not trigger the transmission of a data frame. Receive objects for remote frames may be expanded to a FIFO buffer (see Section 25.3.14.5).

UMask must be set to ‘1.’ The mask bits (Msk[28:0], UMask, MXtd, and MDir bits) may be set to “must-match” or to “don’t care,” to allow groups of remote frames with similar identifiers to be accepted. The Dir bit should not be masked in typical applications. For details, see Section 25.3.15.8.

The arbitration bits (ID[28:0] and Xtd bit) may be given by the application. They define the identifier and type of accepted received remote frames. If some bits of the mask bits are set to “don’t care,” the corresponding bits of the arbitration bits will be overwritten by the bits of the stored remote frame. If an 11-bit Identifier (standard frame) is used (Xtd = ‘0’), it is programmed to ID[28:18]. In this case, ID[17:0] can be ignored. When a remote frame with an 11-bit Identifier is received, ID[17:0] will be set to ‘0.’

The data length code (DLC[3:0]) may be given by the application. When the message handler stores a remote frame in the message object, it will store the received data length code. The data bytes of the message object will remain unchanged.

If the RxIE bit is set, the IntPnd bit will be set when a received remote frame is accepted and stored in the message object.

25.3.14.5 Configuration of a FIFO Buffer

With the exception of the EoB bit, the configuration of receive objects belonging to a FIFO buffer is the same as the configuration of a single receive object.

To concatenate multiple message objects to a FIFO buffer, the identifiers and masks (if used) of these message objects have to be programmed to matching values. Due to the implicit priority of the message objects, the message object with the lowest number will be the first message object of the FIFO buffer. The EoB bit of all message objects of a FIFO buffer except the last one have to be programmed to zero. The EoB bits of the last message object of a FIFO Buffer is set to one, configuring it as the end of the block.

25.3.15 Message Handling

When initialization is finished, the DCAN module synchronizes itself to the traffic on the CAN bus. It does acceptance filtering on received messages and stores those frames that are accepted into the designated message objects. The application has to update the data of the messages to be transmitted and to enable and request their transmission. The transmission is requested automatically when a matching remote frame is received.

The application may read messages which are received and accepted. Messages that are not read before the next messages is accepted for the same message object will be overwritten.

Messages may be read interrupt-driven or after polling of NewDat.

25.3.15.1 Message Handler Overview

The message handler state machine controls the data transfer between the Rx/Tx shift register of the CAN core and the message RAM. It performs the following tasks:

- Data transfer from message RAM to CAN core (messages to be transmitted)
- Data transfer from CAN core to the message RAM (received messages)
- Data transfer from CAN core to the acceptance filtering unit
- Scanning of message RAM for a matching message object (acceptance filtering)
- Scanning the same message object after being changed by IF1/IF2 registers when priority is the same or higher as the message the object found by last scanning
- Handling of TxRqst flags
- Handling of interrupt flags

The message handler registers contains status flags of all message objects grouped into the following topics:

- Transmission Request Flags
- New Data Flags
- Interrupt Pending Flags
- Message Valid Registers

Instead of collecting above listed status information of each message object via IFx registers separately, these message handler registers provides a fast and easy way to get an overview, for example, about all pending transmission requests.

All message handler registers are read-only.

25.3.15.2 Receive/Transmit Priority

The receive/transmit priority for the message objects is attached to the message number, not to the CAN identifier. Message object 1 has the highest priority, while the last implemented message object has the lowest priority. If more than one transmission request is pending, they are serviced due to the priority of the corresponding message object so messages with the highest priority, for example, can be placed in the message objects with the lowest numbers.

The acceptance filtering for received data frames or remote frames is also done in ascending order of message objects, so a frame that has been accepted by a message object cannot be accepted by another message object with a higher message number. The last message object may be configured to accept any data frame or remote frame that was not accepted by any other message object, for nodes that need to log the complete message traffic on the CAN bus.

25.3.15.3 Transmission of Messages in Event-Driven CAN Communication

If the shift register of the CAN core is ready for loading and if there is no data transfer between the IFx registers and message RAM, the D bits in the Message Valid register (MSGVAL12 to MSGVAL78) and the TxRqst bits in the transmission request register are evaluated. The valid message object with the highest priority pending transmission request is loaded into the shift register by the message handler and the transmission is started. The message object’s NewDat bit is reset.

After a successful transmission and if no new data was written to the message object (NewDat = ‘0’) since the start of the transmission, the TxRqst bit will be reset. If TxIE is set, IntPnd will be set after a successful transmission. If the DCAN module has lost the arbitration or if an error occurred during the transmission, the message will be retransmitted as soon as the CAN bus is free again. If meanwhile the transmission of a message with higher priority has been requested, the messages will be transmitted in the order of their priority.

If automatic retransmission mode is disabled by setting the DAR bit in the CTL register, the behavior of bits TxRqst and NewDat in the Message Control register of the interface register set is as follows:

- When a transmission starts, the TxRqst bit of the respective interface register set is reset, while bit NewDat remains set.
- When the transmission has been successfully completed, the NewDat bit is reset.

When a transmission failed (lost arbitration or error) bit NewDat remains set. To restart the transmission, the application has to set TxRqst again.

Received remote frames do not require a receive object. They will automatically trigger the transmission of a data frame, if in the matching transmit object the RmtEn bit is set.

25.3.15.4 Updating a Transmit Object

The CPU may update the data bytes of a transmit object any time via the IF1 and IF2 interface registers, neither D nor TxRqst have to be reset before the update.

Even if only part of the data bytes is to be updated, all four bytes in the corresponding IF1 or IF2 Data A register (IF1DATA or IF2DATA) or IF1 or IF2 Data B register (IF1DATB or IF2DATB) have to be valid before the content of that register is transferred to the message object. Either the CPU has to write all four bytes into the IF1/IF2 data register or the message object is transferred to the IF1/IF2 data register before the CPU writes the new data bytes.

When only the data bytes are updated, first 0x87 can be written to bits [23:16] of the IF1 and IF2 Command register (IF1CMD and IF2CMD) and then the number of the message object is written to bits [7:0] of the command register, concurrently updating the data bytes and setting TxRqst with NewDat.

To prevent the reset of TxRqst at the end of a transmission that may already be in progress while the data is updated, NewDat has to be set together with TxRqst in event driven CAN communication. For details, see Section 25.3.15.3.

When NewDat is set together with TxRqst, NewDat will be reset as soon as the new transmission has started.

25.3.15.5 Changing a Transmit Object

If the number of implemented message objects is not sufficient to be used as permanent message objects only, the transmit objects may be managed dynamically. The CPU can write the whole message (arbitration, control, and data) into the interface register. The bits [23:16] of the command register can be set to 0xB7 for the transfer of the whole message object content into the message object. Neither D nor TxRqst have to be reset before this operation.

If a previously requested transmission of this message object is not completed but already in progress, it will be continued; however, it will not be repeated if it is disturbed.

To only update the data bytes of a message to be transmitted, bits [23:16] of the command register should be set to 0x87.

NOTE: After the update of the transmit object, the interface register set will contain a copy of the actual contents of the object, including the part that had not been updated.

25.3.15.6 Acceptance Filtering of Received Messages

When the arbitration and control bits (Identifier + IDE + RTR + DLC) of an incoming message are completely shifted into the shift register of the CAN core, the message handler starts the scan of the message RAM for a matching valid message object:

- The acceptance filtering unit is loaded with the arbitration bits from the CAN core shift register.
- Then the arbitration and mask bits (including MsgVal, UMask, NewDat, and EoB) of message object 1 are loaded into the acceptance filtering unit and are compared with the arbitration bits from the shift register. This is repeated for all following message objects until a matching message object is found, or until the end of the message RAM is reached.
- If a match occurs, the scanning is stopped and the message handler proceeds depending on the type of the frame (data frame or remote frame) received.

25.3.15.7 Reception of Data Frames

The message handler stores the message from the CAN core shift register into the respective message object in the message RAM. Not only the data bytes, but all arbitration bits and the data length code are stored into the corresponding message object. This ensures that the data bytes stay associated to the identifier even if arbitration mask registers are used.

The NewDat bit is set to indicate that new data (not yet seen by the CPU) has been received. The CPU should reset the NewDat bit when it reads the message object. If at the time of the reception the NewDat bit was already set, MsgLst is set to indicate that the previous data (supposedly not seen by the CPU) is lost. If the RxIE bit is set, the IntPnd bit is set, causing the interrupt register (INT) to point to this message object.

The TxRqst bit of this message object is reset to prevent the transmission of a remote frame, while the requested data frame has just been received.

25.3.15.8 Reception of Remote Frames

When a remote frame is received, three different configurations of the matching message object have to be considered:

- Dir = ‘1’ (direction = transmit), RmtEn = ‘1’, UMask = ‘1’ or ‘0’

  The TxRqst bit of this message object is set at the reception of a matching remote frame. The rest of the message object remains unchanged.
- Dir = ‘1’ (direction = transmit), RmtEn = ‘0’, UMask = ‘0’

  The remote frame is ignored, this message object remains unchanged.
- Dir = ‘1’ (direction = transmit), RmtEn = ‘0’, UMask = ‘1’

  The remote frame is treated similar to a received data frame. At the reception of a matching Remote Message Frame, the TxRqst bit of this message object is reset. The arbitration and control bits (Identifier + IDE + RTR + DLC) from the shift register are stored in the message object in the message RAM and the NewDat bit of this message object is set. The data bytes of the message object remain unchanged.

25.3.15.9 Reading Received Messages

The CPU may read a received message any time via the IFx interface register. The data consistency is guaranteed by the message handler state machine. Typically the CPU will write first 0x7F to bits [23:16] and then the number of the message object to bits [7:0] of the command register. That combination will transfer the entire received message from the message RAM into the interface register set. Additionally, the bits NewDat and IntPnd are cleared in the message RAM (not in the interface register set). The values of these bits in the message control register always reflect the status before resetting the bits. If the message object uses masks for acceptance filtering, the arbitration bits show which of the different matching messages has been received.

The actual value of NewDat shows whether a new message has been received since last time when this message object was read. The actual value of MsgLst shows whether more than one message has been received since the last time when this message object was read. MsgLst will not be automatically reset.

25.3.15.10 Requesting New Data for a Receive Object

By means of a remote frame, the CPU may request another CAN node to provide new data for a receive object. Setting the TxRqst bit of a receive object will cause the transmission of a remote frame with the identifier of the receive object. This remote frame triggers the other CAN node to start the transmission of the matching data frame. If the matching data frame is received before the remote frame could be transmitted, the TxRqst bit is automatically reset. Setting the TxRqst bit without changing the contents of a message object requires the value 0x84 in bits [23:16] of the IFxCMD register.

25.3.15.11 Storing Received Messages in FIFO Buffers

Several message objects may be grouped to form one or more FIFO buffers. Each FIFO buffer configured to store received messages with a particular (group of) identifier(s). Arbitration and mask registers of the FIFO buffer's message objects are identical. The end of buffer (EoB) bits of all but the last of the FIFO buffer's message objects are '0'; in the last one the EoB bit is '1.'

Received messages with identifiers matching to a FIFO buffer are stored into a message object of this FIFO buffer, starting with the message object with the lowest message number. When a message is stored into a message object of a FIFO buffer, the NewDat bit of this message object is set. By setting NewDat while EoB is '0', the message object is locked for further write accesses by the message handler until the CPU has cleared the NewDat bit.

Messages are stored into a FIFO buffer until the last message object of this FIFO buffer is reached. If none of the preceding message objects is released by writing NewDat to '0,' all further messages for this FIFO buffer will be written into the last message object of the FIFO buffer (EoB = '1') and therefore overwrite previous messages in this message object.

25.3.15.12 Reading From a FIFO Buffer

Several messages may be accumulated in a set of message objects which are concatenated to form a FIFO buffer before the application program is required (in order to avoid the loss of data) to empty the buffer.

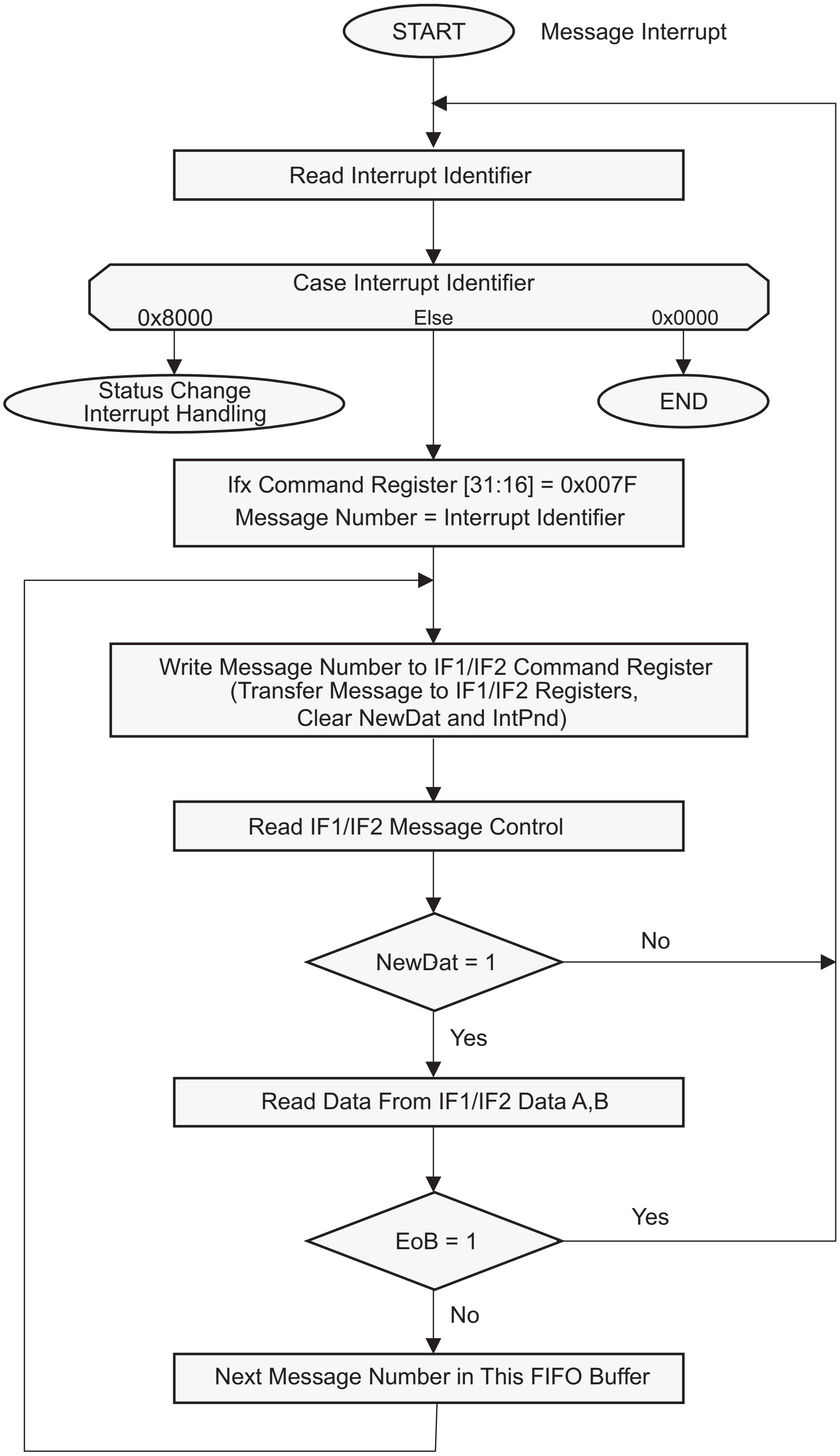
A FIFO buffer of length N will store –1 plus the last received message since last time it was cleared.

A FIFO buffer is cleared by reading and resetting the NewDat bits of all its message objects, starting at the FIFO Object with the lowest message number. This should be done in a subroutine following the example shown in Figure 25-12.

NOTE: All message objects of a FIFO buffer need to be read and cleared before the next batch of messages can be stored. Otherwise, true FIFO functionality can not be guaranteed, since the message objects of a partly read buffer will be re-filled according to the normal (descending) priority.

Reading from a FIFO buffer message object and resetting its NewDat bit is handled the same way as reading from a single message object.

Figure 25-12. CPU Handling of a FIFO Buffer (Interrupt Driven)



25.3.16 CAN Bit Timing

The DCAN supports bit rates between less than 1 kBit/s and 1000 kBit/s.

Each member of the CAN network has its own clock generator, typically derived from a crystal oscillator. The bit timing parameters can be configured individually for each CAN node, creating a common bit rate even though the CAN nodes' oscillator periods (f_{osc}) may be different.

The frequencies of these oscillators are not absolutely stable. Small variations are caused by changes in temperature or voltage and by deteriorating components. As long as the variations remain inside a specific oscillator tolerance range (df), the CAN nodes are able to compensate for the different bit rates by resynchronizing to the bit stream.

In many cases, the CAN bit synchronization will amend a faulty configuration of the CAN bit timing to such a degree that only occasionally an error frame is generated. In the case of arbitration however, when two or more CAN nodes simultaneously try to transmit a frame, a misplaced sample point may cause one of the transmitters to become error passive.

The analysis of such sporadic errors requires a detailed knowledge of the CAN bit synchronization inside a CAN node and of the CAN nodes' interaction on the CAN bus.

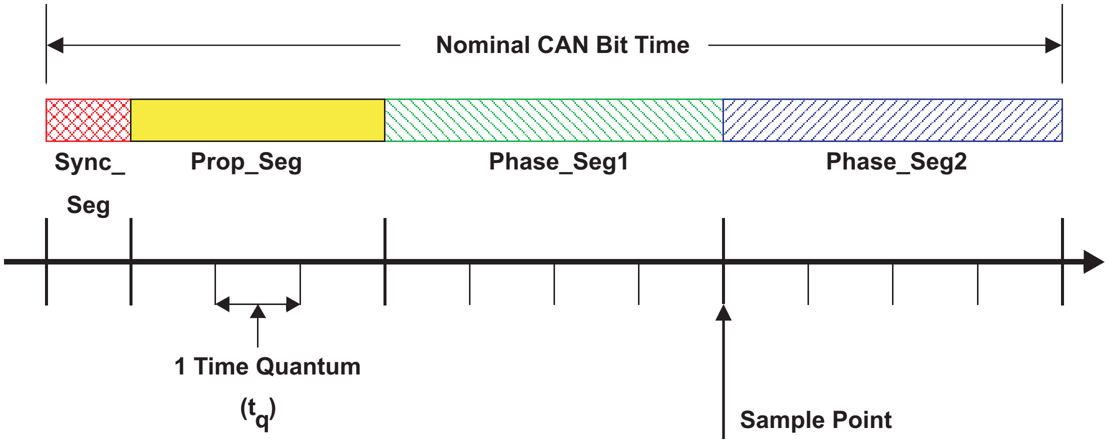
Even if minor errors in the configuration of the CAN bit timing do not result in immediate failure, the performance of a CAN network can be reduced significantly.

25.3.16.1 Bit Time and Bit Rate

According to the CAN specification, the bit time is divided into four segments (see Figure 25-13):

- Synchronization segment (Sync_Seg)
- Propagation time segment (Prop_Seg)
- Phase buffer segment 1 (Phase_Seg1)
- Phase buffer segment 2 (Phase_Seg2)

Figure 25-13. Bit Timing



Each segment consists of a specific number of time quanta. The length of one time quantum (t_q), which is the basic time unit of the bit time, is given by the CAN_CLK and the baud rate prescalers (BRPE and BRP). With these two baud rate prescalers combined, divider values from 1 to 1024 can be programmed:

t_q = Baud Rate Prescaler / CAN_CLK

Apart from the fixed length of the synchronization segment, these numbers are programmable. Table 25-7 describes the minimum programmable ranges required by the CAN protocol.

A given bit rate may be met by different bit time configurations.

Table 25-7. Parameters of the CAN Bit Time

Parameter	Range	Remark
Sync_Seg	1 t_q (fixed)	Synchronization of bus input to CAN_CLK
Prop_Seg	[1 … 8] t_q	Compensates for the physical delay times
Phase_Seg1	[1 … 8] t_q	May be lengthened temporarily by synchronization
Phase_Seg2	[1 … 8] t_q	May be shortened temporarily by synchronization
Synchronization Jump Width (SJW)	[1 … 4] t_q	May not be longer than either Phase Buffer Segment

NOTE: For proper functionality of the CAN network, the physical delay times and the oscillator's tolerance range have to be considered.

25.3.16.1.1 Synchronization Segment

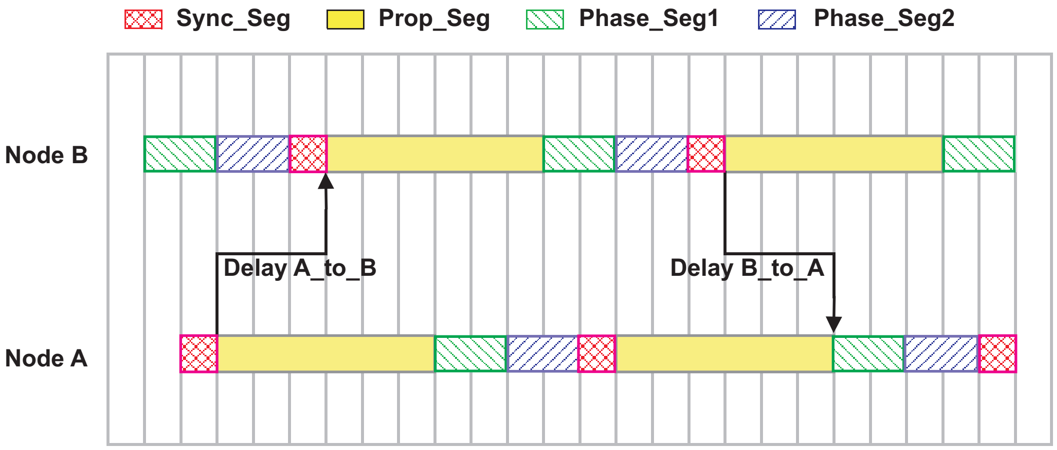
The synchronization segment (Sync_Seg) is the part of the bit time where edges of the CAN bus level are expected to occur. If an edge occurs outside of Sync_Seg, its distance to the Sync_Seg is called the phase error of this edge.

25.3.16.1.2 Propagation Time Segment

This part of the bit time is used to compensate physical delay times within the CAN network. These delay times consist of the signal propagation time on the bus and the internal delay time of the CAN nodes.

Any CAN node synchronized to the bit stream on the CAN bus can be out of phase with the transmitter of the bit stream, caused by the signal propagation time between the two nodes. The CAN protocol's nondestructive bitwise arbitration and the dominant acknowledge bit provided by receivers of CAN messages require that a CAN node transmitting a bit stream must also be able to receive dominant bits transmitted by other CAN nodes that are synchronized to that bit stream. The example in Figure 25-14 shows the phase shift and propagation times between two CAN nodes.

Figure 25-14. The Propagation Time Segment



Delay A_to_B >= node output delay(A) + bus line delay(AÆB) + node input delay(B)

Prop_Seg >= Delay A_to_B + Delay B_to_A

Prop_Seg >= 2 • [max(node output delay+bus line delay + node input delay)]

In this example, both nodes A and B are transmitters performing an arbitration for the CAN bus. The node A has sent its start of frame bit less than one bit time earlier than node B, therefore node B has synchronized itself to the received edge from recessive to dominant. Since node B has received this edge delay(A_to_B) after it has been transmitted, node B's bit timing segments are shifted with regard to node A. Node B sends an identifier with higher priority and so it will win the arbitration at a specific identifier bit when it transmits a dominant bit while node A transmits a recessive bit. The dominant bit transmitted by node B will arrive at node A after the delay (B_to_A).

Due to oscillator tolerances, the actual position of node A's sample point can be anywhere inside the nominal range of node A's Phase Buffer Segments, so the bit transmitted by node B must arrive at node A before the start of Phase_Seg1. This condition defines the length of Prop_Seg.

If the edge from recessive to dominant transmitted by node B would arrive at node A after the start of Phase_Seg1, it could happen that node A samples a recessive bit instead of a dominant bit, resulting in a bit error and the destruction of the current frame by an error flag.

This error only occurs when two nodes arbitrate for the CAN bus which have oscillators of opposite ends of the tolerance range and are separated by a long bus line; this is an example of a minor error in the bit timing configuration (Prop_Seg too short) that causes sporadic bus errors.

Some CAN implementations provide an optional 3 Sample Mode. The DCAN does not. In this mode, the CAN bus input signal passes a digital low-pass filter, using three samples and a majority logic to determine the valid bit value. This results in an additional input delay of 1 t_q, requiring a longer Prop_Seg.

25.3.16.1.3 Phase Buffer Segments and Synchronization

The phase buffer segments (Phase_Seg1 and Phase_Seg2) and the synchronization jump width (SJW) are used to compensate for the oscillator tolerance.

The phase buffer segments surround the sample point and may be lengthened or shortened by synchronization.

The synchronization jump width (SJW) defines how far the resynchronizing mechanism may move the sample point inside the limits defined by the phase buffer segments to compensate for edge phase errors.

Synchronizations occur on edges from recessive to dominant. Their purpose is to control the distance between edges and sample points.

Edges are detected by sampling the actual bus level in each time quantum and comparing it with the bus level at the previous sample point. A synchronization may be done only if a recessive bit was sampled at the previous sample point and if the actual time quantum's bus level is dominant.

An edge is synchronous if it occurs inside of Sync_Seg; otherwise, its distance to the Sync_Seg is the edge phase error, measured in time quanta. If the edge occurs before Sync_Seg, the phase error is negative, else it is positive.

Two types of synchronization exist: hard synchronization and resynchronizing. A hard synchronization is done once at the start of a frame; inside a frame, only resynchronization is possible.

- Hard Synchronization

  After a hard synchronization, the bit time is restarted with the end of Sync_Seg, regardless of the edge phase error. Thus hard synchronization forces the edge which has caused the hard synchronization, to lie within the synchronization segment of the restarted bit time.

- Bit Resynchronizations

  Resynchronization leads to a shortening or lengthening of the Bit time such that the position of the sample point is shifted with regard to the edge.

  When the phase error of the edge which causes resynchronization is positive, Phase_Seg1 is lengthened. If the magnitude of the phase error is less than SJW, Phase_Seg1 is lengthened by the magnitude of the phase error, else it is lengthened by SJW.

  When the phase error of the edge which causes Resynchronization is negative, Phase_Seg2 is shortened. If the magnitude of the phase error is less than SJW, Phase_Seg2 is shortened by the magnitude of the phase error, else it is shortened by SJW.

If the magnitude of the phase error of the edge is less than or equal to the programmed value of SJW, the results of hard synchronization and resynchronization are the same. If the magnitude of the phase error is larger than SJW, the resynchronization cannot compensate the phase error completely, and an error of (phase error - SJW) remains.

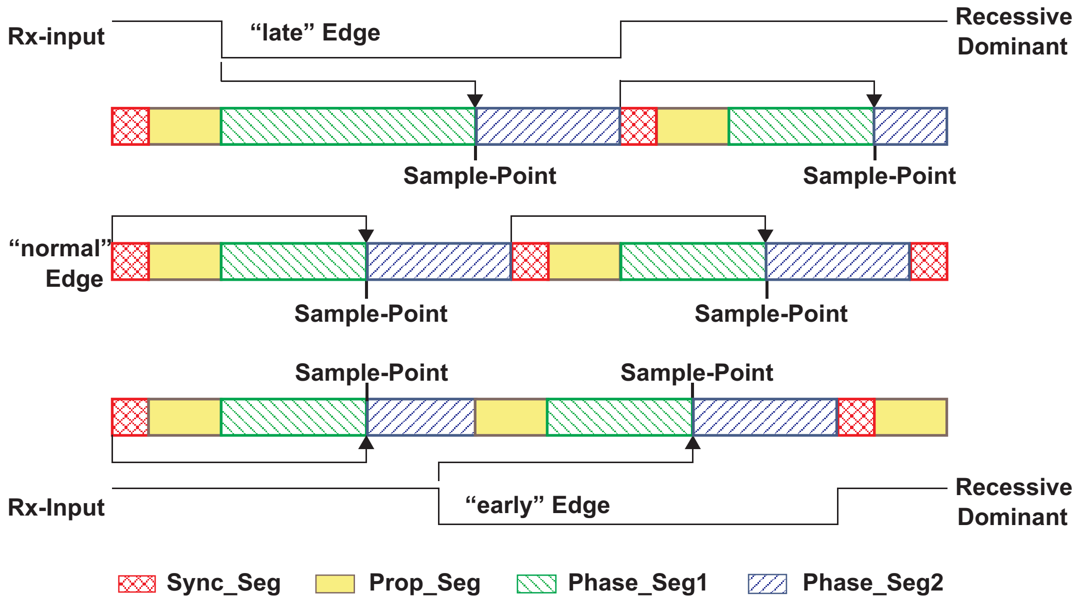
Only one synchronization may be done between two sample points. The synchronizations maintain a minimum distance between edges and sample points, giving the bus level time to stabilize and filtering out spikes that are shorter than (Prop_Seg + Phase_Seg1).

Apart from noise spikes, most synchronizations are caused by arbitration. All nodes synchronize “hard” on the edge transmitted by the “leading” transceiver that started transmitting first, but due to propagation delay times, they cannot become ideally synchronized. The leading transmitter does not necessarily win the arbitration; therefore, the receivers have to synchronize themselves to different transmitters that subsequently take the lead and that are differently synchronized to the previously leading transmitter. The same happens at the acknowledge field, where the transmitter and some of the receivers will have to synchronize to that receiver that takes the lead in the transmission of the dominant acknowledge bit.

Synchronizations after the end of the arbitration will be caused by oscillator tolerance, when the differences in the oscillator’s clock periods of transmitter and receivers sum up during the time between synchronizations (at most ten bits). These summarized differences may not be longer than the SJW, limiting the oscillator’s tolerance range.

Figure 25-15 shows how the phase buffer segments are used to compensate for phase errors. There are three drawings of each two consecutive bit timings. The upper drawing shows the synchronization on a “late” edge, the lower drawing shows the synchronization on an “early” edge, and the middle drawing is the reference without synchronization.

Figure 25-15. Synchronization on Late and Early Edges



In the first example, an edge from recessive to dominant occurs at the end of Prop_Seg. The edge is "late" since it occurs after the Sync_Seg. Reacting to the late edge, Phase_Seg1 is lengthened so that the distance from the edge to the sample point is the same as it would have been from the Sync_Seg to the sample point if no edge had occurred. The phase error of this late edge is less than SJW, so it is fully compensated and the edge from dominant to recessive at the end of the bit, which is one nominal bit time long, occurs in the Sync_Seg.

In the second example, an edge from recessive to dominant occurs during Phase_Seg2. The edge is "early" since it occurs before a Sync_Seg. Reacting to the early edge, Phase_Seg2 is shortened and Sync_Seg is omitted, so that the distance from the edge to the sample point is the same as it would have been from a Sync_Seg to the sample point if no edge had occurred. As in the previous example, the magnitude of this early edge’s phase error is less than SJW, so it is fully compensated.

The phase buffer segments are lengthened or shortened temporarily only; at the next bit time, the segments return to their nominal programmed values.

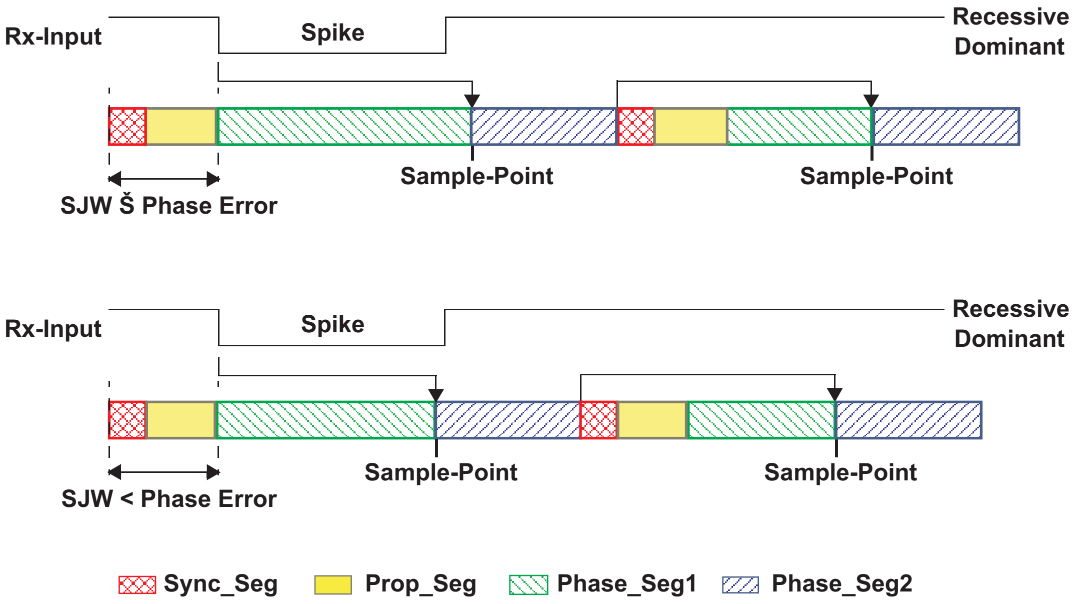
In these examples, the bit timing is seen from the point of view of the CAN implementation's state machine, where the bit time starts and ends at the sample points. The state machine omits Sync_Seg when synchronizing on an early edge because it cannot subsequently redefine that time quantum of Phase_Seg2 where the edge occurs to be the Sync_Seg.

Figure 25-16 shows how short dominant noise spikes are filtered by synchronizations. In both examples, the spike starts at the end of Prop_Seg and has the length of (Prop_Seg + Phase_Seg1).

In the first example, the synchronization jump width is greater than or equal to the phase error of the spike's edge from recessive to dominant. Therefore the sample point is shifted after the end of the spike; a recessive bus level is sampled.

In the second example, SJW is shorter than the phase error, so the sample point cannot be shifted far enough; the dominant spike is sampled as actual bus level.

Figure 25-16. Filtering of Short Dominant Spikes



25.3.16.1.4 *Oscillator Tolerance Range*

With the introduction of CAN protocol version 1.2, the option to synchronize on edges from dominant to recessive became obsolete. Only edges from recessive to dominant are considered for synchronization. The protocol update to version 2.0 (A and B) had no influence on the oscillator tolerance.

The tolerance range df for an oscillator's frequency f_{osc} around the nominal frequency f_{nom} with:

$$(1 - df) \bullet f_{nom} \leq f_{osc} \leq (1 + df) \bullet f_{nom}$$

depends on the proportions of Phase_Seg1, Phase_Seg2, SJW, and the bit time. The maximum tolerance df is the defined by two conditions (both shall be met):

$$\text{I:} \quad df \leq \frac{\min(TSeg1, TSeg2)}{2x(13x(bit\_time - TSeg2))}$$

$$\text{II:} \quad df \leq \frac{SJW}{20xbit\_time}$$

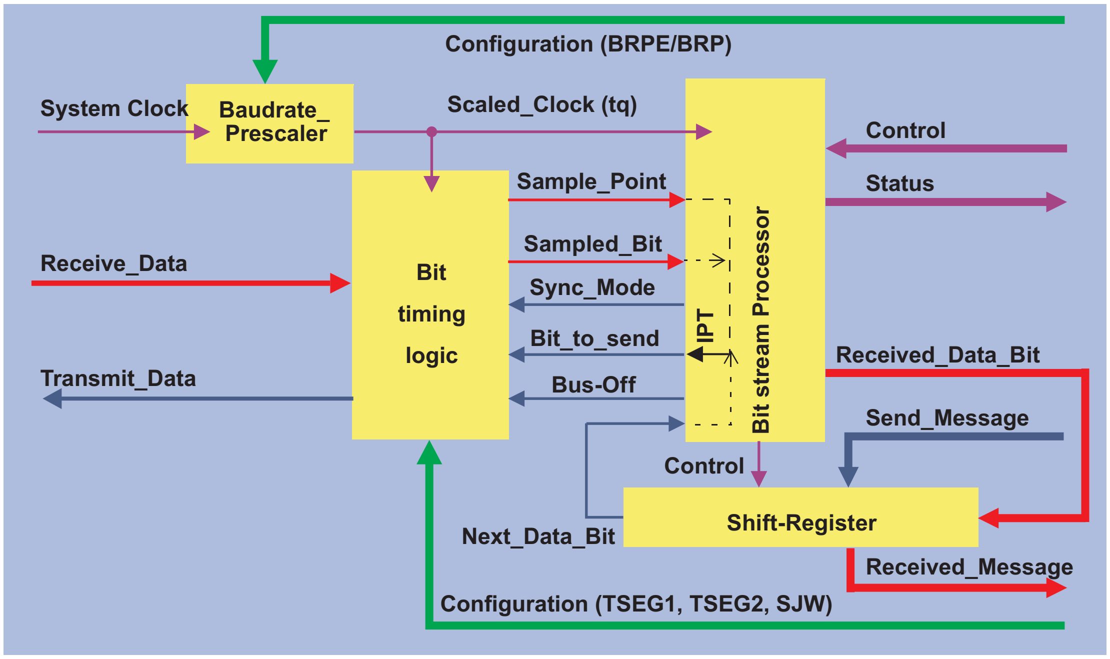
It has to be considered that SJW may not be larger than the smaller of the phase buffer segments and that the propagation time segment limits that part of the bit time that may be used for the phase buffer segments.

The combination Prop_Seg = 1 and Phase_Seg1 = Phase_Seg2 = SJW = 4 allows the largest possible oscillator tolerance of 1.58%. This combination with a propagation time segment of only 10% of the bit time is not suitable for short bit times; it can be used for bit rates of up to 125 kBit/s (bit time = 8 μs) with a bus length of 40 m.

25.3.16.2 DCAN Bit Timing Registers

In the DCAN, the bit timing configuration is programmed in two register bytes, additionally a third byte for a baud rate prescaler extension of four bits (BREP) is provided. The sum of Prop_Seg and Phase_Seg1 (as TSEG1) is combined with Phase_Seg2 (as TSEG2) in one byte, SJW and BRP (plus BRPE in third byte) are combined in the other byte (see Figure 25-17).

Figure 25-17. Structure of the CAN Core's CAN Protocol Controller



In this bit timing register, the components TSEG1, TSEG2, SJW and BRP have to be programmed to a numerical value that is one less than its functional value; so instead of values in the range of [1…n], values in the range of [0…–1] are programmed. That way, e.g., SJW (functional range of [1…4]) is represented by only two bits.

Therefore, the length of the bit time is (programmed values) [TSEG1 + TSEG2 + 3] t_q or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q.

The data in the bit timing register (BTR) is the configuration input of the CAN protocol controller. The baud rate prescaler (configured by BRPE/BRP) defines the length of the time quantum (the basic time unit of the bit time); the bit timing logic (configured by TSEG1, TSEG2, and SJW) defines the number of time quanta in the bit time.

The processing of the bit time, the calculation of the position of the sample point, and occasional synchronizations are controlled by the bit timing state machine, which is evaluated once each time quantum. The rest of the CAN protocol controller, the bit stream processor (BSP) state machine, is evaluated once each bit time, at the sample point.

The shift register serializes the messages to be sent and parallelizes received messages. Its loading and shifting is controlled by the BSP. The BSP translates messages into frames and vice versa. It generates and discards the enclosing fixed format bits, inserts and extracts stuff bits, calculates and checks the CRC code, performs the error management, and decides which type of synchronization is to be used. It is evaluated at the sample point and processes the sampled bus input bit. The time after the sample point that is needed to calculate the next bit to be sent (e.g., data bit, CRC bit, stuff bit, error flag, or idle) is called the information processing time (IPT), which is 0 tq for the DCAN.

Generally, the IPT is CAN controller-specific, but may not be longer than 2 tq. The IPC length is the lower limit of the programmed length of Phase_Seg2. In case of a synchronization, Phase_Seg2 may be shortened to a value less than IPT, which does not affect bus timing.

25.3.16.2.1 Calculation of the Bit Timing Parameters

Usually, the calculation of the bit timing configuration starts with a desired bit rate or bit time. The resulting bit time (1 / Bit rate) must be an integer multiple of the CAN clock period.

NOTE: 8 MHz is the minimum CAN clock frequency required to operate the DCAN at a bit rate of 1 MBit/s.

The bit time may consist of 8 to 25 time quanta. The length of the time quantum t_q is defined by the baud rate prescaler with t_q = (Baud Rate Prescaler) / CAN_CLK. Several combinations may lead to the desired bit time, allowing iterations of the following steps.

First part of the bit time to be defined is the Prop_Seg. Its length depends on the delay times measured in the system. A maximum bus length as well as a maximum node delay has to be defined for expandible CAN bus systems. The resulting time for Prop_Seg is converted into time quanta (rounded up to the nearest integer multiple of t_q).

The Sync_Seg is 1 t_q long (fixed), leaving (bit time – Prop_Seg – 1) t_q for the two Phase Buffer Segments. If the number of remaining t_q is even, the Phase Buffer Segments have the same length, Phase_Seg2 = Phase_Seg1, else Phase_Seg2 = Phase_Seg1 + 1.

The minimum nominal length of Phase_Seg2 has to be regarded as well. Phase_Seg2 may not be shorter than any CAN controller’s Information Processing Time in the network, which is device dependent and can be in the range of [0…2] t_q.

The length of the synchronization jump width is set to its maximum value, which is the minimum of four (4) and Phase_Seg1.

The oscillator tolerance range necessary for the resulting configuration is calculated by the formulas given in Section 25.3.16.2.3.

If more than one configurations are possible to reach a certain Bit rate, it is recommended to choose the configuration which allows the highest oscillator tolerance range.

CAN nodes with different clocks require different configurations to come to the same bit rate. The calculation of the propagation time in the CAN network, based on the nodes with the longest delay times, is done once for the whole network.

The CAN system’s oscillator tolerance range is limited by the node with the lowest tolerance range.

The calculation may show that bus length or bit rate have to be decreased or that the oscillator frequencies’ stability has to be increased in order to find a protocol compliant configuration of the CAN bit timing.

The resulting configuration is written into the bit timing register:

Tseg2 = Phase_Seg2-1

Tseg1 = Phase_Seg1+ Prop_Seg-1

SJW = SynchronizationJumpWidth-1

BRP = Prescaler-1

25.3.16.2.2 Example for Bit Timing at High Baud Rate

In this example, the frequency of CAN_CLK is 10 MHz, BRP is 0, the bit rate is 1 MBit/s.

t_q	100 ns	=	t_{CAN_CLK}
delay of bus driver	60 ns	=	
delay of receiver circuit	40 ns	=	
delay of bus line (40 m)	220 ns	=	
t_{Prop}	700 ns	=	INT (2*delays + 1) = 7 • t_q
t_{SJW}	100 ns	=	1 • t_q
t_{TSeg1}	800 ns	=	$t_{Prop} + t_{SJW}$
t_{TSeg2}	100 ns	=	Information Processing Time + 1 • t_q
$t_{Sync\text{-}Seg}$	100 ns	=	1 • t_q
bit time	1000 ns	=	$t_{Sync\text{-}Seg} + t_{TSeg1} + t_{TSeg2}$
tolerance for CAN_CLK	0.43 %	=	$\frac{(\min(TSeg1, TSeg2))}{2x(13x(bit\_time - TSeg2))}$
		=	$\frac{0.1\,\mu s}{2x(13x(1\,\mu s - 0.1\,\mu s))}$

In this example, the concatenated bit time parameters are $(1\text{-}1)_3\&(8\text{-}1)_4\&(1\text{-}1)_2\&(1\text{-}1)_6$, so the bit timing register is programmed to = 00000700.

25.3.16.2.3 Example for Bit Timing at Low Baud Rate

In this example, the frequency of CAN_CLK is 2 MHz, BRP is 1, the bit rate is 100 KBit/s.

t_q	1 µs	=	t_{CAN_CLK}
Delay of bus driver	200 ns	=	
Delay of receiver circuit	80 ns	=	
Delay of bus line (40 m)	220 ns	=	
t_{Prop}	1 µs	=	1 • t_q
t_{SJW}	4 µs	=	4 • t_q
t_{TSeg1}	5 µs	=	$t_{Prop} + t_{SJW}$
t_{TSeg2}	3 µs	=	Information Processing Time + 3 • t_q
$t_{Sync\text{-}Seg}$	1 µs	=	1 • t_q
Bit time	9 µs	=	$t_{Sync\text{-}Seg} + t_{TSeg1} + t_{TSeg2}$
Tolerance for CAN_CLK		=	$\frac{(\min(TSeg1, TSeg2))}{2x(13x(bit\_time - TSeg2))}$
		=	$\frac{0.1\,\mu s}{2x(13x(1\,\mu s - 0.1\,\mu s))}$

In this example, the concatenated bit time parameters are $(3\text{-}1)_3\&(5\text{-}1)_4\&(4\text{-}1)_2\&(2\text{-}1)_6$, so the bit timing register is programmed to = 0x000024C1.

25.3.17 Message Interface Register Sets

The interface register sets control the CPU read and write accesses to the message RAM. There are two interface registers sets for read/write access, IF1 and IF2 and one interface register set for read access only, IF3.

Due to the structure of the message RAM, it is not possible to change single bits or bytes of a message object. Instead, always a complete message object in the message RAM is accessed. Therefore the data transfer from the IF1/IF2 registers to the message RAM requires the message handler to perform a read-modify-write cycle: First those parts of the message object that are not to be changed are read from the message RAM into the interface register set, and after the update the whole content of the interface register set is written into the message object.

After the partial write of a message object, those parts of the interface register set that are not selected in the command register, will be set to the actual contents of the selected message object.

After the partial read of a message object, those parts of the interface register set that are not selected in the command register, will be left unchanged.

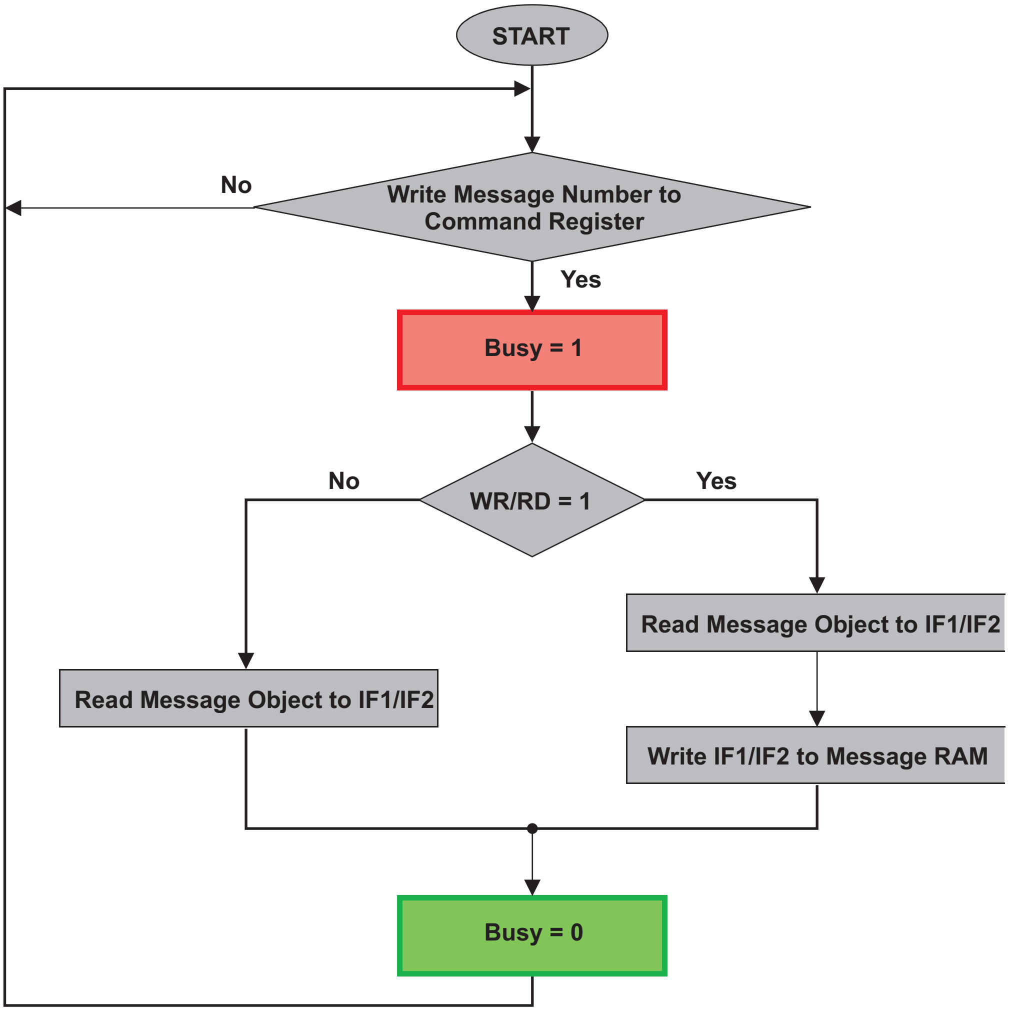
By buffering the data to be transferred, the interface register sets avoid conflicts between concurrent CPU accesses to the message RAM and CAN message reception and transmission. A complete message object (see Section 25.3.18.1) or parts of the message object may be transferred between the message RAM and the IF1/IF2 register set (see) in one single transfer. This transfer, performed in parallel on all selected parts of the message object, guarantees the data consistency of the CAN message.

25.3.17.1 Message Interface Register Sets 1 and 2

The IF1 and IF2 register sets control the data transfer to and from the message object. The IF1CMD and IF2CMD registers address the desired message object in the message RAM and specify whether a complete message object or only parts should be transferred. The data transfer is initiated by writing the message number to the bits [7:0] of the IF1CMD and IF2CMD register.

When the CPU initiates a data transfer between the IF1/IF2 registers and message RAM, the message handler sets the busy bit in the respective command register to ‘1’. After the transfer has completed, the busy bit is set back to ‘0’ (see Figure 25-18).

Figure 25-18. Data Transfer Between IF1/IF2 Registers and Message RAM



25.3.17.2 IF3 Register Set

The IF3 register set can automatically be updated with received message objects without the need to initiate the transfer from message RAM by CPU. The intention of this feature of IF3 is to provide an interface for the DMA to read packets efficiently. The automatic update functionality can be programmed for each message object using the update enable registers IF3UPD12 to IF3UPD78.

All valid message objects in message RAM which are configured for automatic update, will be checked for active NewDat flags. If such a message object is found, it will be transferred to the IF3 register (if no previous DMA transfers are ongoing), controlled by IF3 Observation register (IF3OBS). If more than one NewDat flag is active, the message object with the lowest number has the highest priority for automatic IF3 update.

The NewDat bit in the message object will be reset by a transfer to IF3.

If DCAN internal IF3 update is complete, a DMA request is generated. The DMA request stays active until first read access to one of the IF3 registers. The DMA functionality has to be enabled by setting bit DE3 in the CTL register. Please refer to the device datasheet to find out if this DMA source is available.

NOTE: The IF3 register set can not be used for transferring data into message objects.

25.3.18 Message RAM

The DCAN message RAM contains message objects and parity bits for the message objects. There are up to 64 message objects in the message RAM.

During normal operation, accesses to the message RAM are performed via the interface register sets, and the CPU cannot directly access the message RAM.

The interface register sets IF1 and IF2 provide indirect read/write access from the CPU to the message RAM. The IF1 and IF2 register sets can buffer control and user data to be transferred to and from the message objects.

The third interface register set IF3 can be configured to automatically receive control and user data from the message RAM when a message object has been updated after reception of a CAN message. The CPU does not need to initiate the transfer from message RAM to IF3 register set.

The message handler avoids potential conflicts between concurrent accesses to message RAM and CAN frame reception/transmission.

There are two modes where the message RAM can be directly accessed by the CPU:

- Debug/Suspend mode (see Section 25.3.18.3)
- RAM Direct Access (RDA) mode (see Section 25.3.18.4)

25.3.18.1 Structure of Message Objects

Table 25-8 shows the structure of a message object.

The grayed fields are those parts of the message object which are represented in dedicated registers. For example, the transmit request flags of all message objects are represented in centralized transmit request registers.

Table 25-8. Structure of a Message Object

Message Object												
UMask	**Msk[28:0]**	**MXtd**	**MDir**	**EoB**	**unused**	**NewDat**	**MsgLst**	**RxIE**	**TxIE**	**IntPnd**	**RmtEn**	**TxRqst**
MsgVal	ID[28:0]	Xtd	Dir	DLC[3:0]	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7

Table 25-9. Field Descriptions

Name	Value	Description
MsgVal		Message valid
	0	The message object is ignored by the message handler.
	1	The message object is to be used by the message handler.
		Note: This bit may be kept at level ‘1’ even when the identifier bits ID[28:0], the control bits Xtd, Dir, or the data length code are changed. It should be reset if the Messages Object is no longer required.
UMask		Use acceptance mask
	0	Mask bits (Msk[28:0], MXtd and MDir) are ignored and not used for acceptance filtering.
	1	Mask bits are used for acceptance filtering.
		Note: If the UMask bit is set to one, the message object’s mask bits have to be programmed during initialization of the message object before MsgVal is set to one.
ID[28:0]		Message identifier
	ID[28:0]	29-bit (“extended”) identifier bits
	ID[28:18]	11-bit (“standard”) identifier bits
Msk[28:0]		Identifier mask
	0	The corresponding bit in the message identifier is not used for acceptance filtering (don’t care).
	1	The corresponding bit in the message identifier is used for acceptance filtering.
Xtd		Mask extended identifier
	0	The extended identifier bit (IDE) has no effect on the acceptance filtering.
	1	The extended identifier bit (IDE) is used for acceptance filtering.
		Note: When 11-bit (“standard”) Identifiers are used for a message object, the identifiers of received data frames are written into bits ID[28:18]. For acceptance filtering, only these bits together with mask bits Msk[28:18] are considered.
Dir		Message direction
	0	Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, the message is stored in this message object.
	1	Direction = transmit: On TxRqst, a data frame is transmitted. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = one).
MDir		Mask message direction
	0	The message direction bit (Dir) has no effect on the acceptance filtering.
	1	The message direction bit (Dir) is used for acceptance filtering.
EOB		End of block
	0	The message object is part of a FIFO Buffer block and is not the last message object of this FIFO Buffer block.
	1	The message object is a single message object or the last message object in a FIFO Buffer Block.
		Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to one.

Table 25-9. Field Descriptions (continued)

Name	Value	Description
NewDat		New data
	0	No new data has been written into the data bytes of this message object by the message handler since the last time when this flag was cleared by the CPU.
	1	The message handler or the CPU has written new data into the data bytes of this message object.
MsgLst		Message lost (only valid for message objects with direction = receive)
	0	No message was lost since the last time when this bit was reset by the CPU.
	1	The message handler stored a new message into this message object when NewDat was still set, so the previous message has been overwritten.
RxIE		Receive interrupt enable
	0	IntPnd will not be triggered after the successful reception of a frame.
	1	IntPnd will be triggered after the successful reception of a frame.
TxIE		Transmit interrupt enable
	0	IntPnd will not be triggered after the successful transmission of a frame.
	1	IntPnd will be triggered after the successful transmission of a frame.
IntPnd		Interrupt pending
	0	This message object is not the source of an interrupt.
	1	This message object is the source of an interrupt. The interrupt Identifier in the interrupt register will point to this message object if there is no other interrupt source with higher priority.
RmtEn		Remote enable
	0	At the reception of a remote frame, TxRqst is not changed.
	1	At the reception of a remote frame, TxRqst is set.
TxRqst		Transmit request
	0	This message object is not waiting for a transmission.
	1	The transmission of this message object is requested and is not yet done.
DLC[3:0]		Data length code
	0	Data frame has 0 to 8 data bytes.
	1	Data frame has 8 data bytes.
		Note: The data length code of a message object must be defined to the same value as in the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.
Data 0		1st data byte of a CAN data frame
Data 1		2nd data byte of a CAN data frame
Data 2		3rd data byte of a CAN data frame
Data 3		4th data byte of a CAN data frame
Data 4		5th data byte of a CAN data frame
Data 5		6th data byte of a CAN data frame
Data 6		7th data byte of a CAN data frame
Data 7		8th data byte of a CAN data frame
		Note: Byte Data 0 is the first data byte shifted into the shift register of the CAN core during a reception, byte Data 7 is the last. When the message handler stores a data frame, it will write all the eight data bytes into a message object. If the data length code is less than 8, the remaining bytes of the message object may be overwritten by undefined values.

25.3.18.2 Addressing Message Objects in RAM

The starting location of a particular message object in RAM is:
Message RAM base address + (message object number) * 0x20

This means that message object 1 starts at offset 0x0020; message object 2 starts at offset 0x0040, etc.

‘0’ is not a valid message object number.

The base address for DCAN0 RAM is 0x481C_D000 and DCAN1 RAM is 0x481D_1000.

Message object number 1 has the highest priority.

Table 25-10. Message RAM addressing in Debug/Suspend and RDA Mode

Message Object Number	Offset From Base Address	Word Number	Debug/Suspend Mode, see Section 25.3.18.3	RDA mode, see Section 25.3.18.4
1	0x0020	1	Parity	Data Bytes 4-7
	0x0024	2	MXtd,MDir,Mask	Data Bytes 0-3
	0x0028	3	Xtd,Dir,ID	ID[27:0],DLC
	0x002C	4	Ctrl	Mask,Xtd,Dir,ID[28]
	0x0030	5	Data Bytes 3-0	Parity,Ctrl,MXtd,MDir
	0x0034	6	Data Bytes 7-4	…
…	…	…	…	…
31	0x03E0	1	Parity	Data Bytes 4-7
	0x03E4	2	MXtd,MDir,Mask	Data Bytes 0-3
	0x03E8	3	Xtd,Dir,ID	ID[27:0],DLC
	0x03EC	4	Ctrl	Mask,Xtd,Dir,ID[28]
	0x03F0	5	Data Bytes 3-0	Parity,Ctrl,MXtd,MDir
	0x03F4	6	Data Bytes 7-4	…
…	…	…	…	…
63	0x07E0	1	Parity	Data Bytes 4-7
	0x07E4	2	MXtd,MDir,Mask	Data Bytes 0-3
	0x07E8	3	Xtd,Dir,ID	ID[27:0],DLC
	0x07EC	4	Ctrl	Mask,Xtd,Dir,ID[28]
	0x07F0	5	Data Bytes 3-0	Parity,Ctrl,MXtd,MDir
	0x07F4	6	Data Bytes 7-4	…
last implemented (32(DCAN3) or 64)	0x0000	1	Parity	Data Bytes 4-7
	0x0004	2	MXtd,MDir,Mask	Data Bytes 0-3
	0x0008	3	Xtd,Dir,ID	ID[27:0],DLC
	0x000C	4	Ctrl	Mask,Xtd,Dir,ID[28]
	0x0010	5	Data Bytes 3-0	Parity,Ctrl,MXtd,MDir
	0x0014	6	Data Bytes 7-4	…

25.3.18.3 Message RAM Representation in Debug/Suspend Mode

In debug/suspend mode, the message RAM will be memory mapped. This allows the external debug unit to access the message RAM.

NOTE: During debug/suspend mode, the message RAM cannot be accessed via the IFx register sets.

Table 25-11. Message RAM Representation in Debug/Suspend Mode

Bit #	31/ 15	30/ 14	29/ 13	29/ 12	27/ 11	26/ 10	25/ 9	24/ 8	23/ 7	22/ 6	21/ 5	20/ 4	19/ 3	18/ 2	17/ 1	16/ 0
MsgAddr + 0x00	Reserved															
	Reserved											Parity[4:0]				
MsgAddr + 0x04	MXtd	MDir	Rsvd	Msk[28:16]												
	Msk[15:0]															
MsgAddr + 0x08	Rsvd	Xtd	Dir	ID[28:16]												
	ID[15:0]															
MsgAddr + 0x0C	Reserved															
	Rsvd	MsgLst	Rsvd	UMask	TxIE	RxTE	RmtEn	Rsvd	EOB	Reserved		DLC[3:0]				
MsgAddr + 0x10	Data 3								Data 2							
	Data 1								Data 0							
MsgAddr + 0x14	Data 7								Data 6							
	Data 5								Data 4							

25.3.18.4 Message RAM Representation in Direct Access Mode

When the RDA bit in the TEST register is set while the DCAN module is in test mode (test bit in the CTL register is set), the CPU has direct access to the message RAM. Due to the 32-bit bus structure, the RAM is split into word lines to support this feature. The CPU has access to one word line at a time only.

In RAM direct access mode, the RAM is represented by a continuous memory space within the address frame of the DCAN module, starting at the message RAM base address.

Note: During direct access mode, the message RAM cannot be accessed via the IFx register sets.

Any read or write to the RAM addresses for RamDirectAccess during normal operation mode (TestMode bit or RDA bit not set) will be ignored.

Table 25-12. Message RAM Representation in RAM Direct Access Mode

Bit #	31/ 15	30/ 14	29/ 13	29/ 12	27/ 11	26/ 10	25/ 9	24/ 8	23/ 7	22/ 6	21/ 5	20/ 4	19/ 3	18/ 2	17/ 1	16/ 0
MsgAddr + 0x00	Data 4								Data 5							
	Data 6								Data 7							
MsgAddr + 0x04	Data 0								Data 1							
	Data 2								Data 3							
MsgAddr + 0x08	ID[27:12]															
	ID[11:0]											DLC[3:0]				
	Msk[28:13]															

Table 25-12. Message RAM Representation in RAM Direct Access Mode (continued)

Bit #	31/15	30/14	29/13	29/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
MsgAddr + 0x0C	Msk[12:0]													Xtd	Dir	ID[28]
MsgAddr + 0x10	Reserved															Parity [4]
	MsgLst				Unused				Msg Lst	UMa sk	TxIE	RxT E	Rmt En	EOB	MX td	MDir

NOTE: Writes to unused bits have no effect.

25.3.19 GIO Support

The CAN_RX and CAN_TX pins of the DCAN module can be used as general purpose IO pins, if CAN functionality is not needed. This function is controlled by the CAN TX IO control register (TIOC) and the CAN RX IO control register (RIOC).

25.4 DCAN Registers

25.4.1 DCAN Registers

Table 25-13 lists the memory-mapped registers for the DCAN. All register offset addresses not listed in Table 25-13 should be considered as reserved locations and the register contents should not be modified.

Table 25-13. DCAN Registers

Offset	Acronym	Register Name	Section
0h	DCAN_CTL	CAN Control Register	Section 25.4.1.1
4h	DCAN_ES	Error and Status Register	Section 25.4.1.2
8h	DCAN_ERRC	Error Counter Register	Section 25.4.1.3
Ch	DCAN_BTR	Bit Timing Register	Section 25.4.1.4
10h	DCAN_INT	Interrupt Register	Section 25.4.1.5
14h	DCAN_TEST	Test Register	Section 25.4.1.6
1Ch	DCAN_PERR	Parity Error Code Register	Section 25.4.1.7
80h	DCAN_ABOTR	Auto-Bus-On Time Register	Section 25.4.1.8
84h	DCAN_TXRQ_X	Transmission Request X Register	Section 25.4.1.9
88h	DCAN_TXRQ12	Transmission Request Register 12	Section 25.4.1.10
8Ch	DCAN_TXRQ34	Transmission Request Register 34	Section 25.4.1.11
90h	DCAN_TXRQ56	Transmission Request Register 56	Section 25.4.1.12
94h	DCAN_TXRQ78	Transmission Request Register 78	Section 25.4.1.13
98h	DCAN_NWDAT_X	New Data X Register	Section 25.4.1.14
9Ch	DCAN_NWDAT12	New Data Register 12	Section 25.4.1.15
A0h	DCAN_NWDAT34	New Data Register 34	Section 25.4.1.16
A4h	DCAN_NWDAT56	New Data Register 56	Section 25.4.1.17
A8h	DCAN_NWDAT78	New Data Register 78	Section 25.4.1.18
ACh	DCAN_INTPND_X	Interrupt Pending X Register	Section 25.4.1.19
B0h	DCAN_INTPND12	Interrupt Pending Register 12	Section 25.4.1.20
B4h	DCAN_INTPND34	Interrupt Pending Register 34	Section 25.4.1.21
B8h	DCAN_INTPND56	Interrupt Pending Register 56	Section 25.4.1.22
BCh	DCAN_INTPND78	Interrupt Pending Register 78	Section 25.4.1.23
C0h	DCAN_MSGVAL_X	Message Valid X Register	Section 25.4.1.24

Table 25-13. DCAN Registers (continued)

Offset	Acronym	Register Name	Section
C4h	DCAN_MSGVAL12	Message Valid Register 12	Section 25.4.1.25
C8h	DCAN_MSGVAL34	Message Valid Register 34	Section 25.4.1.26
CCh	DCAN_MSGVAL56	Message Valid Register 56	Section 25.4.1.27
D0h	DCAN_MSGVAL78	Message Valid Register 78	Section 25.4.1.28
D8h	DCAN_INTMUX12	Interrupt Multiplexer Register 12	Section 25.4.1.29
DCh	DCAN_INTMUX34	Interrupt Multiplexer Register 34	Section 25.4.1.30
E0h	DCAN_INTMUX56	Interrupt Multiplexer Register 56	Section 25.4.1.31
E4h	DCAN_INTMUX78	Interrupt Multiplexer Register 78	Section 25.4.1.32
100h	DCAN_IF1CMD	IF1 Command Registers	Section 25.4.1.33
104h	DCAN_IF1MSK	IF1 Mask Register	Section 25.4.1.34
108h	DCAN_IF1ARB	IF1 Arbitration Register	Section 25.4.1.35
10Ch	DCAN_IF1MCTL	IF1 Message Control Register	Section 25.4.1.36
110h	DCAN_IF1DATA	IF1 Data A Register	Section 25.4.1.37
114h	DCAN_IF1DATB	IF1 Data B Register	Section 25.4.1.38
120h	DCAN_IF2CMD	IF2 Command Registers	Section 25.4.1.39
124h	DCAN_IF2MSK	IF2 Mask Register	Section 25.4.1.40
128h	DCAN_IF2ARB	IF2 Arbitration Register	Section 25.4.1.41
12Ch	DCAN_IF2MCTL	IF2 Message Control Register	Section 25.4.1.42
130h	DCAN_IF2DATA	IF2 Data A Register	Section 25.4.1.43
134h	DCAN_IF2DATB	IF2 Data B Register	Section 25.4.1.44
140h	DCAN_IF3OBS	IF3 Observation Register	Section 25.4.1.45
144h	DCAN_IF3MSK	IF3 Mask Register	Section 25.4.1.46
148h	DCAN_IF3ARB	IF3 Arbitration Register	Section 25.4.1.47
14Ch	DCAN_IF3MCTL	IF3 Message Control Register	Section 25.4.1.48
150h	DCAN_IF3DATA	IF3 Data A Register	Section 25.4.1.49
154h	DCAN_IF3DATB	IF3 Data B Register	Section 25.4.1.50
160h	DCAN_IF3UPD12	IF3 Update Enable Register 12	Section 25.4.1.51
164h	DCAN_IF3UPD34	IF3 Update Enable Register 34	Section 25.4.1.52
168h	DCAN_IF3UPD56	IF3 Update Enable Register 56	Section 25.4.1.53
16Ch	DCAN_IF3UPD78	IF3 Update Enable Register 78	Section 25.4.1.54
1E0h	DCAN_TIOC	CAN TX IO Control Register	Section 25.4.1.55
1E4h	DCAN_RIOC	CAN RX IO Control Register	Section 25.4.1.56

25.4.1.1 DCAN_CTL Register (offset = 0h) [reset = 1401h]

DCAN_CTL is shown in Figure 25-19 and described in Table 25-14.

The Bus-Off recovery sequence (refer to CAN specification) cannot be shortened by setting or resetting Init bit. If the module goes Bus-Off, it will automatically set the Init bit and stop all bus activities. When the Init bit is cleared by the application again, the module will then wait for 129 occurrences of Bus Idle (129 * 11 consecutive recessive bits) before resuming normal operation. At the end of the bus-off recovery sequence, the error counters will be reset. After the Init bit is reset, each time when a sequence of 11 recessive bits is monitored, a Bit0 error code is written to the error and status register, enabling the CPU to check whether the CAN bus is stuck at dominant or continuously disturbed, and to monitor the proceeding of the bus-off recovery sequence.

Figure 25-19. DCAN_CTL Register

31	30	29	28	27	26	25	24
RESERVED						WUBA	PDR
R-0h						R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
RESERVED			DE3	DE2	DE1	IE1	INITDBG
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

15	14	13	12	11	10	9	8
SWR	RESERVED	PMD				ABO	IDS
R/WP-0h	R-0h	R/W-5h				R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
TEST	CCE	DAR	RESERVED	EIE	SIE	IE0	INIT
R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h

Table 25-14. DCAN_CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25	WUBA	R/W	0h	Automatic wake up on bus activity when in local power-down mode. Note: The CAN message, which initiates the bus activity, cannot be received. This means that the first message received in power down and automatic wake-up mode, will be lost. 0h (R/W) = No detection of a dominant CAN bus level while in local power-down mode. 1h (R/W) = Detection of a dominant CAN bus level while in local power-down mode is enabled. On occurrence of a dominant CAN bus level, the wake up sequence is started.
24	PDR	R/W	0h	Request for local low power-down mode 0h (R/W) = No application request for local low power-down mode. If the application has cleared this bit while DCAN in local power-down mode, also the Init bit has to be cleared. 1h (R/W) = Local power-down mode has been requested by application. The DCAN will acknowledge the local power-down mode by setting bit PDA in the error and status register. The local clocks will be turned off by DCAN internal logic.
23-21	RESERVED	R	0h	
20	DE3	R/W	0h	Enable DMA request line for IF3. Note: A pending DMA request for IF3 remains active until first access to one of the IF3 registers. 0h (R/W) = Disabled 1h (R/W) = Enabled
19	DE2	R/W	0h	Enable DMA request line for IF2. Note: A pending DMA request for IF2 remains active until first access to one of the IF2 registers. 0h (R/W) = Disabled 1h (R/W) = Enabled

Table 25-14. DCAN_CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	DE1	R/W	0h	Enable DMA request line for IF1. Note: A pending DMA request for IF1 remains active until first access to one of the IF1 registers. 0h (R/W) = Disabled 1h (R/W) = Enabled
17	IE1	R/W	0h	Interrupt line 1 enable 0h (R/W) = Disabled - Module interrupt DCAN1INT is always low. 1h (R/W) = Enabled - interrupts will assert line DCAN1INT to one; line remains active until pending interrupts are processed.
16	INITDBG	R	0h	Internal init state while debug access 0h (R/W) = Not in debug mode, or debug mode requested but not entered. 1h (R/W) = Debug mode requested and internally entered; the DCAN is ready for debug accesses.
15	SWR	R/WP	0h	SW reset enable. Note: To execute software reset, the following procedure is necessary: (a) Set Init bit to shut down CAN communication and (b) Set SWR bit additionally to Init bit. 0h (R/W) = Normal Operation 1h (R/W) = Module is forced to reset state. This bit will automatically get cleared after execution of SW reset after one OCP clock cycle.
14	RESERVED	R	0h	
13-10	PMD	R/W	5h	Parity on/off. 5 = Parity function disabled. Others = Parity function enabled.
9	ABO	R/W	0h	Auto-Bus-On enable 0h (R/W) = The Auto-Bus-On feature is disabled 1h (R/W) = The Auto-Bus-On feature is enabled
8	IDS	R/W	0h	Interruption debug support enable 0h (R/W) = When Debug/Suspend mode is requested, DCAN will wait for a started transmission or reception to be completed before entering Debug/Suspend mode 1h (R/W) = When Debug/Suspend mode is requested, DCAN will interrupt any transmission or reception, and enter Debug/Suspend mode immediately.
7	TEST	R/W	0h	Test mode enable 0h (R/W) = Normal Operation 1h (R/W) = Test Mode
6	CCE	R/W	0h	Configuration change enable 0h (R/W) = The CPU has no write access to the configuration registers. 1h (R/W) = The CPU has write access to the configuration registers (when Init bit is set).
5	DAR	R/W	0h	Disable automatic retransmission 0h (R/W) = Automatic retransmission of not successful messages enabled. 1h (R/W) = Automatic retransmission disabled.
4	RESERVED	R	0h	
3	EIE	R/W	0h	Error interrupt enable 0h (R/W) = Disabled - PER, BOff and EWarn bits can not generate an interrupt. 1h (R/W) = Enabled - PER, BOff and EWarn bits can generate an interrupt at DCAN0INT line and affect the interrupt register.

Table 25-14. DCAN_CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	SIE	R/W	0h	Status change interrupt enable 0h (R/W) = Disabled - WakeUpPnd, RxOk, TxOk and LEC bits can not generate an interrupt. 1h (R/W) = Enabled - WakeUpPnd, RxOk, TxOk and LEC can generate an interrupt at DCAN0INT line and affect the interrupt register.
1	IE0	R/W	0h	Interrupt line 0 enable 0h (R/W) = Disabled - Module interrupt DCAN0INT is always low. 1h (R/W) = Enabled - interrupts will assert line DCAN0INT to one; line remains active until pending interrupts are processed.
0	INIT	R/W	1h	Initialization 0h (R/W) = Normal operation 1h (R/W) = Initialization mode is entered

25.4.1.2 DCAN_ES Register (offset = 4h) [reset = 6Fh]

DCAN_ES is shown in Figure 25-20 and described in Table 25-15.

Interrupts are generated by bits PER, BOff and EWarn (if EIE bit in CAN control register is set) and by bits WakeUpPnd, RxOk, TxOk, and LEC (if SIE bit in CAN control register is set). A change of bit EPass will not generate an interrupt. Reading the error and status register clears the WakeUpPnd, PER, RxOk and TxOk bits and set the LEC to value '7.' Additionally, the status interrupt value (0x8000) in the interrupt register will be replaced by the next lower priority interrupt value. The EOI for all other interrupts (DCANINT0 and DCANINT1) are automatically handled by hardware. For debug support, the auto clear functionality of error and status register (clear of status flags by read) is disabled when in debug/suspend mode.

Figure 25-20. DCAN_ES Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED					PDA	WAKEUP_PND	PER
R-0h					R-0h	0h	0h
7	6	5	4	3	2	1	0
BOFF	EWARN	EPASS	RXOK	TXOK	LEC		
R-0h	R-1h	R-1h	0h	1h	7h		

Table 25-15. DCAN_ES Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10	PDA	R	0h	Local power-down mode acknowledge 0h (R/W) = DCAN is not in local power-down mode. 1h (R/W) = Application request for setting DCAN to local power-down mode was successful. DCAN is in local power-down mode.
9	WAKEUP_PND		0h	Wake up pending. This bit can be used by the CPU to identify the DCAN as the source to wake up the system. This bit will be reset if error and status register is read. 0h (R/W) = No Wake Up is requested by DCAN. 1h (R/W) = DCAN has initiated a wake up of the system due to dominant CAN bus while module power down.
8	PER		0h	Parity error detected. This bit will be reset if error and status register is read. 0h (R/W) = No parity error has been detected since last read access. 1h (R/W) = The parity check mechanism has detected a parity error in the Message RAM.
7	BOFF	R	0h	Bus-Off state 0h (R/W) = The CAN module is not bus-off state. 1h (R/W) = The CAN module is in bus-off state.
6	EWARN	R	1h	Warning state 0h (R/W) = Both error counters are below the error warning limit of 96. 1h (R/W) = At least one of the error counters has reached the error warning limit of 96.

Table 25-15. DCAN_ES Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	EPASS	R	1h	Error passive state 0h (R/W) = On CAN Bus error, the DCAN could send active error frames. 1h (R/W) = The CAN core is in the error passive state as defined in the CAN Specification.
4	RXOK		0h	Received a message successfully. This bit will be reset if error and status register is read. 0h (R/W) = No message has been successfully received since the last time when this bit was read by the CPU. This bit is never reset by DCAN internal events. 1h (R/W) = A message has been successfully received since the last time when this bit was reset by a read access of the CPU (independent of the result of acceptance filtering).
3	TXOK		1h	Transmitted a message successfully. This bit will be reset if error and status register is read. 0h (R/W) = No message has been successfully transmitted since the last time when this bit was read by the CPU. This bit is never reset by DCAN internal events. 1h (R/W) = A message has been successfully transmitted (error free and acknowledged by at least one other node) since the last time when this bit was reset by a read access of the CPU.
2-0	LEC		7h	Last error code. The LEC field indicates the type of the last error on the CAN bus. This field will be cleared to '0' when a message has been transferred (reception or transmission) without error. 0h (R/W) = No error 1h (R/W) = Stuff error. More than five equal bits in a row have been detected in a part of a received message where this is not allowed. 2h (R/W) = Form error. A fixed format part of a received frame has the wrong format. 3h (R/W) = Ack error. The message this CAN core transmitted was not acknowledged by another node. 4h (R/W) = Bit1 error. During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value '1'), but the monitored bus value was dominant. 5h (R/W) = Bit0 error. During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (logical value '0'), but the monitored bus level was recessive. During Bus-Off recovery, this status is set each time a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceeding of the Bus-Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed). 6h (R/W) = CRC error. In a received message, the CRC check sum was incorrect. (CRC received for an incoming message does not match the calculated CRC for the received data). 7h (R/W) = No CAN bus event was detected since the last time the CPU read the error and status register. Any read access to the error and status register re-initializes the LEC to value '7.'

25.4.1.3 DCAN_ERRC Register (offset = 8h) [reset = 0h]

DCAN_ERRC is shown in Figure 25-21 and described in Table 25-16.

Figure 25-21. DCAN_ERRC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RP	REC							TEC							
R-0h	R-0h							R-0h							

Table 25-16. DCAN_ERRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	RP	R	0h	Receive error passive 0h (R/W) = The receive error counter is below the error passive level. 1h (R/W) = The receive error counter has reached the error passive level as defined in the CAN specification.
14-8	REC	R	0h	Receive error counter. Actual state of the receive error counter (values from 0 to 255).
7-0	TEC	R	0h	Transmit error counter. Actual state of the transmit error counter (values from 0 to 255).

25.4.1.4 DCAN_BTR Register (offset = Ch) [reset = 2301h]

DCAN_BTR is shown in Figure 25-22 and described in Table 25-17.

This register is only writable if CCE and Init bits in the CAN control register are set. The CAN bit time may be programmed in the range of 8 to 25 time quanta. The CAN time quantum may be programmed in the range of 1 to1024 CAN_CLK periods. With a CAN_CLK of 8 MHz and BRPE = 0x00, the reset value of 0x00002301 configures the DCAN for a bit rate of 500kBit/s.

Figure 25-22. DCAN_BTR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				BRPE			
R-0h				R-0h			
15	14	13	12	11	10	9	8
RESERVED	TSEG2			TSEG1			
R-0h	2h			3h			
7	6	5	4	3	2	1	0
SJW		BRP					
0h		1h					

Table 25-17. DCAN_BTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-16	BRPE	R	0h	Baud rate prescaler extension. Valid programmed values are 0 to 15. By programming BRPE the baud rate prescaler can be extended to values up to 1024.
15	RESERVED	R	0h	
14-12	TSEG2		2h	Time segment after the sample point. Valid programmed values are 0 to 7. The actual TSeg2 value which is interpreted for the bit timing will be the programmed TSeg2 value + 1.
11-8	TSEG1		3h	Time segment before the sample point. Valid programmed values are 1 to 15. The actual TSeg1 value interpreted for the bit timing will be the programmed TSeg1 value + 1.
7-6	SJW		0h	Synchronization Jump Width. Valid programmed values are 0 to 3. The actual SJW value interpreted for the synchronization will be the programmed SJW value + 1.
5-0	BRP		1h	Baud rate prescaler. Value by which the CAN_CLK frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid programmed values are 0 to 63. The actual BRP value interpreted for the bit timing will be the programmed BRP value + 1.

25.4.1.5 DCAN_INT Register (offset = 10h) [reset = 0h]

DCAN_INT is shown in Figure 25-23 and described in Table 25-18.

Figure 25-23. DCAN_INT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								INT1ID_23_16							
R-0h								R-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT0ID_15_0															
R-0h															

Table 25-18. DCAN_INT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-16	INT1ID_23_16	R	0h	Interrupt 1 Identifier (indicates the message object with the highest pending interrupt). If several interrupts are pending, the CAN interrupt register will point to the pending interrupt with the highest priority. The DCAN1INT interrupt line remains active until Int1ID reaches value 0 (the cause of the interrupt is reset) or until IE1 is cleared. A message interrupt is cleared by clearing the message object's IntPnd bit. Among the message interrupts, the message object's interrupt priority decreases with increasing message number. 0x 00: No interrupt is pending. 0x 01-0x 80: Number of message object which caused the interrupt. 0xFF: Unused.
15-0	INT0ID_15_0	R	0h	Interrupt Identifier (the number here indicates the source of the interrupt). If several interrupts are pending, the CAN interrupt register will point to the pending interrupt with the highest priority. The DCAN0INT interrupt line remains active until Int0ID reaches value 0 (the cause of the interrupt is reset) or until IE0 is cleared. The Status interrupt has the highest priority. Among the message interrupts, the message object's interrupt priority decreases with increasing message number. 0x 0000: No interrupt is pending. 0x 0001-0x 0080: Number of message object which caused the interrupt. 0x 0081-0x7FFF: Unused (values 0081 to 7FFF). 0x 8000: Error and status register value is not 0x07. 0xFFFF: Unused.

25.4.1.6 DCAN_TEST Register (offset = 14h) [reset = 0h]

DCAN_TEST is shown in Figure 25-24 and described in Table 25-19.

For all test modes, the test bit in CAN control register needs to be set to one. If test bit is set, the RDA, EXL, Tx1, Tx0, LBack and Silent bits are writable. Bit Rx monitors the state of pin CAN_RX and therefore is only readable. All test register functions are disabled when test bit is cleared. The test register is only writable if test bit in CAN control register is set. Setting Tx[1:0] other than '00' will disturb message transfer. When the internal loop-back mode is active (bit LBack is set), bit EXL will be ignored.

Figure 25-24. DCAN_TEST Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED						RDA	EXL
R-0h						0h	0h
7	6	5	4	3	2	1	0
RX	TX		LBACK	SILENT	RESERVED		
R-0h	0h		0h	0h	R-0h		

Table 25-19. DCAN_TEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9	RDA		0h	RAM direct access enable 0h (R/W) = Normal operation 1h (R/W) = Direct access to the RAM is enabled while in test mode
8	EXL		0h	External loopback mode 0h (R/W) = Disabled 1h (R/W) = Enabled
7	RX	R	0h	Receive pin. Monitors the actual value of the CAN_RX pin 0h (R/W) = The CAN bus is dominant 1h (R/W) = The CAN bus is recessive
6-5	TX		0h	Control of CAN_TX pin. 0h (R/W) = Normal operation, CAN_TX is controlled by the CAN core. 1h (R/W) = Sample point can be monitored at CAN_TX pin. 10h (R/W) = CAN_TX pin drives a dominant value. 11h (R/W) = CAN_TX pin drives a recessive value.
4	LBACK		0h	Loopback mode 0h (R/W) = Disabled 1h (R/W) = Enabled
3	SILENT		0h	Silent mode 0h (R/W) = Disabled 1h (R/W) = Enabled
2-0	RESERVED	R	0h	

25.4.1.7 DCAN_PERR Register (offset = 1Ch) [reset = 0h]

DCAN_PERR is shown in Figure 25-25 and described in Table 25-20.

If a parity error is detected, the PER flag will be set in the error and status register. This bit is not reset by the parity check mechanism; it must be reset by reading the error and status register. In addition to the PER flag, the parity error code register will indicate the memory area where the parity error has been detected (message number and word number). If more than one word with a parity error was detected, the highest word number with a parity error will be displayed. After a parity error has been detected, the register will hold the last error code until power is removed.

Figure 25-25. DCAN_PERR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED					WORD_NUMBER		
R-0h					R-0h		
7	6	5	4	3	2	1	0
MESSAGE_NUMBER							
R-0h							

Table 25-20. DCAN_PERR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10-8	WORD_NUMBER	R	0h	Word number where parity error has been detected. 0x 01-0x 05: RDA word number (1 to 5) of the message object (according to the message RAM representation in RDA mode).
7-0	MESSAGE_NUMBER	R	0h	Message number. 0x 01-0x 80: Message object number where parity error has been detected.

25.4.1.8 DCAN_ABOTR Register (offset = 80h) [reset = 0h]

DCAN_ABOTR is shown in Figure 25-26 and described in Table 25-21.

On write access to the CAN control register while Auto-Bus-On timer is running, the Auto-Bus-On procedure will be aborted. During Debug/Suspend mode, running Auto-Bus-On timer will be paused.

Figure 25-26. DCAN_ABOTR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABO_TIME																															
R/W-0h																															

Table 25-21. DCAN_ABOTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ABO_TIME	R/W	0h	Number of OCP clock cycles before a Bus-Off recovery sequence is started by clearing the Init bit. This function has to be enabled by setting bit ABO in CAN control register. The Auto-Bus-On timer is realized by a 32 bit counter that starts to count down to zero when the module goes Bus-Off. The counter will be reloaded with the preload value of the ABO time register after this phase.

25.4.1.9 DCAN_TXRQ_X Register (offset = 84h) [reset = 0h]

DCAN_TXRQ_X is shown in Figure 25-27 and described in Table 25-22.

Example 1. Bit 0 of the transmission request X register represents byte 0 of the transmission request 1 register. If one or more bits in this byte are set, bit 0 of the transmission request X register will be set.

Figure 25-27. DCAN_TXRQ_X Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TXRQSTREG8		TXRQSTREG7		TXRQSTREG6		TXRQSTREG5	
R-0h		R-0h		R-0h		R-0h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXRQSTREG4		TXRQSTREG3		TXRQSTREG2		TXRQSTREG1	
R-0h		R-0h		R-0h		R-0h	

Table 25-22. DCAN_TXRQ_X Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-14	TXRQSTREG8	R	0h	TxRqstReg8
13-12	TXRQSTREG7	R	0h	TxRqstReg7
11-10	TXRQSTREG6	R	0h	TxRqstReg6
9-8	TXRQSTREG5	R	0h	TxRqstReg5
7-6	TXRQSTREG4	R	0h	TxRqstReg4
5-4	TXRQSTREG3	R	0h	TxRqstReg3
3-2	TXRQSTREG2	R	0h	TxRqstReg2
1-0	TXRQSTREG1	R	0h	TxRqstReg1

25.4.1.10 DCAN_TXRQ12 Register (offset = 88h) [reset = 0h]

DCAN_TXRQ12 is shown in Figure 25-28 and described in Table 25-23.

The TXRQ12 to TXRQ78 registers hold the TxRqst bits of the implemented message objects. By reading out these bits, the CPU can check for pending transmission requests. The TxRqst bit in a specific message object can be set/reset by the CPU via the IF1/IF2 message interface registers, or by the message handler after reception of a remote frame or after a successful transmission.

Figure 25-28. DCAN_TXRQ12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXRQS																															
R-0h																															

Table 25-23. DCAN_TXRQ12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TXRQS	R	0h	Transmission request bits [0:31] (for all message objects) 0h (R/W) = No transmission has been requested for this message object. 1h (R/W) = The transmission of this message object is requested and is not yet done.

25.4.1.11 DCAN_TXRQ34 Register (offset = 8Ch) [reset = 0h]

DCAN_TXRQ34 is shown in Figure 25-29 and described in Table 25-24.

The TXRQ12 to TXRQ78 registers hold the TxRqst bits of the implemented message objects. By reading out these bits, the CPU can check for pending transmission requests. The TxRqst bit in a specific message object can be set/reset by the CPU via the IF1/IF2 message interface registers, or by the message handler after reception of a remote frame or after a successful transmission.

Figure 25-29. DCAN_TXRQ34 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXRQS																															
R-0h																															

Table 25-24. DCAN_TXRQ34 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TXRQS	R	0h	Transmission request bits [32:63] (for all message objects) 0h (R/W) = No transmission has been requested for this message object. 1h (R/W) = The transmission of this message object is requested and is not yet done.

25.4.1.12 DCAN_TXRQ56 Register (offset = 90h) [reset = 0h]

DCAN_TXRQ56 is shown in Figure 25-30 and described in Table 25-25.

The TXRQ12 to TXRQ78 registers hold the TxRqst bits of the implemented message objects. By reading out these bits, the CPU can check for pending transmission requests. The TxRqst bit in a specific message object can be set/reset by the CPU via the IF1/IF2 message interface registers, or by the message handler after reception of a remote frame or after a successful transmission.

Figure 25-30. DCAN_TXRQ56 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXRQS																															
R-0h																															

Table 25-25. DCAN_TXRQ56 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TXRQS	R	0h	Transmission request bits [64:95] (for all message objects) 0h (R/W) = No transmission has been requested for this message object. 1h (R/W) = The transmission of this message object is requested and is not yet done.

25.4.1.13 DCAN_TXRQ78 Register (offset = 94h) [reset = 0h]

DCAN_TXRQ78 is shown in Figure 25-31 and described in Table 25-26.

The TXRQ12 to TXRQ78 registers hold the TxRqst bits of the implemented message objects. By reading out these bits, the CPU can check for pending transmission requests. The TxRqst bit in a specific message object can be set/reset by the CPU via the IF1/IF2 message interface registers, or by the message handler after reception of a remote frame or after a successful transmission.

Figure 25-31. DCAN_TXRQ78 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXRQS																															
R-0h																															

Table 25-26. DCAN_TXRQ78 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TXRQS	R	0h	Transmission request bits [96:127] (for all message objects) 0h (R/W) = No transmission has been requested for this message object. 1h (R/W) = The transmission of this message object is requested and is not yet done.

25.4.1.14 DCAN_NWDAT_X Register (offset = 98h) [reset = 0h]

DCAN_NWDAT_X is shown in Figure 25-32 and described in Table 25-27.

With the new data X register, the CPU can detect if one or more bits in the different new data registers are set. Each register bit represents a group of eight message objects. If at least on of the NewDat bits of these message objects are set, the corresponding bit in the new data X register will be set. Example 1. Bit 0 of the new data X register represents byte 0 of the new data 1 register. If one or more bits in this byte are set, bit 0 of the new data X register will be set.

Figure 25-32. DCAN_NWDAT_X Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
NEWDATREG8		NEWDATREG7		NEWDATREG6		NEWDATREG5	
R-0h		R-0h		R-0h		R-0h	
7	6	5	4	3	2	1	0
NEWDATREG4		NEWDATREG3		NEWDATREG2		NEWDATREG1	
R-0h		R-0h		R-0h		R-0h	

Table 25-27. DCAN_NWDAT_X Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-14	NEWDATREG8	R	0h	NewDatReg8
13-12	NEWDATREG7	R	0h	NewDatReg7
11-10	NEWDATREG6	R	0h	NewDatReg6
9-8	NEWDATREG5	R	0h	NewDatReg5
7-6	NEWDATREG4	R	0h	NewDatReg4
5-4	NEWDATREG3	R	0h	NewDatReg3
3-2	NEWDATREG2	R	0h	NewDatReg2
1-0	NEWDATREG1	R	0h	NewDatReg1

25.4.1.15 DCAN_NWDAT12 Register (offset = 9Ch) [reset = 0h]

DCAN_NWDAT12 is shown in Figure 25-33 and described in Table 25-28.

These registers hold the NewDat bits of the implemented message objects. By reading out these bits, the CPU can check for new data in the message objects. The NewDat bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after reception of a data frame or after a successful transmission.

Figure 25-33. DCAN_NWDAT12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEWDAT																															
R-0h																															

Table 25-28. DCAN_NWDAT12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NEWDAT	R	0h	New Data Bits [0:31] (for all message objects) 0h (R/W) = No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. 1h (R/W) = The message handler or the CPU has written new data into the data portion of this message object.

25.4.1.16 DCAN_NWDAT34 Register (offset = A0h) [reset = 0h]

DCAN_NWDAT34 is shown in Figure 25-34 and described in Table 25-29.

These registers hold the NewDat bits of the implemented message objects. By reading out these bits, the CPU can check for new data in the message objects. The NewDat bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after reception of a data frame or after a successful transmission.

Figure 25-34. DCAN_NWDAT34 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEWDAT																															
R-0h																															

Table 25-29. DCAN_NWDAT34 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NEWDAT	R	0h	New Data Bits [32:63] (for all message objects) 0h (R/W) = No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. 1h (R/W) = The message handler or the CPU has written new data into the data portion of this message object.

25.4.1.17 DCAN_NWDAT56 Register (offset = A4h) [reset = 0h]

DCAN_NWDAT56 is shown in Figure 25-35 and described in Table 25-30.

These registers hold the NewDat bits of the implemented message objects. By reading out these bits, the CPU can check for new data in the message objects. The NewDat bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after reception of a data frame or after a successful transmission.

Figure 25-35. DCAN_NWDAT56 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEWDAT																															
R-0h																															

Table 25-30. DCAN_NWDAT56 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NEWDAT	R	0h	New Data Bits [64:95] (for all message objects) 0h (R/W) = No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. 1h (R/W) = The message handler or the CPU has written new data into the data portion of this message object.

25.4.1.18 DCAN_NWDAT78 Register (offset = A8h) [reset = 0h]

DCAN_NWDAT78 is shown in Figure 25-36 and described in Table 25-31.

These registers hold the NewDat bits of the implemented message objects. By reading out these bits, the CPU can check for new data in the message objects. The NewDat bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after reception of a data frame or after a successful transmission.

Figure 25-36. DCAN_NWDAT78 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">NEWDAT</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 25-31. DCAN_NWDAT78 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NEWDAT	R	0h	New Data Bits [96:127] (for all message objects) 0h (R/W) = No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. 1h (R/W) = The message handler or the CPU has written new data into the data portion of this message object.

25.4.1.19 DCAN_INTPND_X Register (offset = ACh) [reset = 0h]

DCAN_INTPND_X is shown in Figure 25-37 and described in Table 25-32.

With the interrupt pending X register, the CPU can detect if one or more bits in the different interrupt pending registers are set. Each bit of this register represents a group of eight message objects. If at least one of the IntPnd bits of these message objects are set, the corresponding bit in the interrupt pending X register will be set. Example 2. Bit 0 of the interrupt pending X register represents byte 0 of the interrupt pending 1 register. If one or more bits in this byte are set, bit 0 of the interrupt pending X register will be set.

Figure 25-37. DCAN_INTPND_X Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
INTPNDREG8		INTPNDREG7		INTPNDREG6		INTPNDREG5	
R-0h		R-0h		R-0h		R-0h	
7	6	5	4	3	2	1	0
INTPNDREG4		INTPNDREG3		INTPNDREG2		INTPNDREG1	
R-0h		R-0h		R-0h		R-0h	

Table 25-32. DCAN_INTPND_X Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-14	INTPNDREG8	R	0h	IntPndReg8
13-12	INTPNDREG7	R	0h	IntPndReg7
11-10	INTPNDREG6	R	0h	IntPndReg6
9-8	INTPNDREG5	R	0h	IntPndReg5
7-6	INTPNDREG4	R	0h	IntPndReg4
5-4	INTPNDREG3	R	0h	IntPndReg3
3-2	INTPNDREG2	R	0h	IntPndReg2
1-0	INTPNDREG1	R	0h	IntPndReg1

25.4.1.20 DCAN_INTPND12 Register (offset = B0h) [reset = 0h]

DCAN_INTPND12 is shown in Figure 25-38 and described in Table 25-33.

These registers hold the IntPnd bits of the implemented message objects. By reading out these bits, the CPU can check for pending interrupts in the message objects. The IntPnd bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

Figure 25-38. DCAN_INTPND12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTPND																															
R-0h																															

Table 25-33. DCAN_INTPND12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTPND	R	0h	Interrupt Pending Bits [0:31] (for all message objects) 0h (R/W) = This message object is not the source of an interrupt. 1h (R/W) = This message object is the source of an interrupt.

25.4.1.21 DCAN_INTPND34 Register (offset = B4h) [reset = 0h]

DCAN_INTPND34 is shown in Figure 25-39 and described in Table 25-34.

These registers hold the IntPnd bits of the implemented message objects. By reading out these bits, the CPU can check for pending interrupts in the message objects. The IntPnd bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

Figure 25-39. DCAN_INTPND34 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">INTPND</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 25-34. DCAN_INTPND34 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTPND	R	0h	Interrupt Pending Bits [32:63] (for all message objects) 0h (R/W) = This message object is not the source of an interrupt. 1h (R/W) = This message object is the source of an interrupt.

25.4.1.22 DCAN_INTPND56 Register (offset = B8h) [reset = 0h]

DCAN_INTPND56 is shown in Figure 25-40 and described in Table 25-35.

These registers hold the IntPnd bits of the implemented message objects. By reading out these bits, the CPU can check for pending interrupts in the message objects. The IntPnd bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

Figure 25-40. DCAN_INTPND56 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTPND																															
R-0h																															

Table 25-35. DCAN_INTPND56 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTPND	R	0h	Interrupt Pending Bits [64:95] (for all message objects) 0h (R/W) = This message object is not the source of an interrupt. 1h (R/W) = This message object is the source of an interrupt.

25.4.1.23 DCAN_INTPND78 Register (offset = BCh) [reset = 0h]

DCAN_INTPND78 is shown in Figure 25-41 and described in Table 25-36.

These registers hold the IntPnd bits of the implemented message objects. By reading out these bits, the CPU can check for pending interrupts in the message objects. The IntPnd bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

Figure 25-41. DCAN_INTPND78 Register

31-0
INTPND
R-0h

Table 25-36. DCAN_INTPND78 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTPND	R	0h	Interrupt Pending Bits [96:127] (for all message objects) 0h (R/W) = This message object is not the source of an interrupt. 1h (R/W) = This message object is the source of an interrupt.

25.4.1.24 DCAN_MSGVAL_X Register (offset = C0h) [reset = 0h]

DCAN_MSGVAL_X is shown in Figure 25-42 and described in Table 25-37.

With the message valid X register, the CPU can detect if one or more bits in the different message valid registers are set. Each bit of this register represents a group of eight message objects. If at least one of the MsgVal bits of these message objects are set, the corresponding bit in the message valid X register will be set. Example 3. Bit 0 of the message valid X register represents byte 0 of the message valid 1 register. If one or more bits in this byte are set, bit 0 of the message valid X register will be set.

Figure 25-42. DCAN_MSGVAL_X Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
MSGVALREG8		MSGVALREG7		MSGVALREG6		MSGVALREG5	
R-0h		R-0h		R-0h		R-0h	
7	6	5	4	3	2	1	0
MSGVALREG4		MSGVALREG3		MSGVALREG2		MSGVALREG1	
R-0h		R-0h		R-0h		R-0h	

Table 25-37. DCAN_MSGVAL_X Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-14	MSGVALREG8	R	0h	MsgValReg8
13-12	MSGVALREG7	R	0h	MsgValReg7
11-10	MSGVALREG6	R	0h	MsgValReg6
9-8	MSGVALREG5	R	0h	MsgValReg5
7-6	MSGVALREG4	R	0h	MsgValReg4
5-4	MSGVALREG3	R	0h	MsgValReg3
3-2	MSGVALREG2	R	0h	MsgValReg2
1-0	MSGVALREG1	R	0h	MsgValReg1

25.4.1.25 DCAN_MSGVAL12 Register (offset = C4h) [reset = 0h]

DCAN_MSGVAL12 is shown in Figure 25-43 and described in Table 25-38.

These registers hold the MsgVal bits of the implemented message objects. By reading out these bits, the CPU can check which message objects are valid. The MsgVal bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

Figure 25-43. DCAN_MSGVAL12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSGVAL																															
R-0h																															

Table 25-38. DCAN_MSGVAL12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MSGVAL	R	0h	Message valid bits [0:31] (for all message objects) 0h (R/W) = This message object is ignored by the message handler. 1h (R/W) = This message object is configured and will be considered by the message handler.

25.4.1.26 DCAN_MSGVAL34 Register (offset = C8h) [reset = 0h]

DCAN_MSGVAL34 is shown in Figure 25-44 and described in Table 25-39.

These registers hold the MsgVal bits of the implemented message objects. By reading out these bits, the CPU can check which message objects are valid. The MsgVal bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

Figure 25-44. DCAN_MSGVAL34 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSGVAL																															
R-0h																															

Table 25-39. DCAN_MSGVAL34 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MSGVAL	R	0h	Message valid bits [32:63] (for all message objects) 0h (R/W) = This message object is ignored by the message handler. 1h (R/W) = This message object is configured and will be considered by the message handler.

25.4.1.27 DCAN_MSGVAL56 Register (offset = CCh) [reset = 0h]

DCAN_MSGVAL56 is shown in Figure 25-45 and described in Table 25-40.

These registers hold the MsgVal bits of the implemented message objects. By reading out these bits, the CPU can check which message objects are valid. The MsgVal bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

Figure 25-45. DCAN_MSGVAL56 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">MSGVAL</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 25-40. DCAN_MSGVAL56 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MSGVAL	R	0h	Message valid bits [64:95] (for all message objects) 0h (R/W) = This message object is ignored by the message handler. 1h (R/W) = This message object is configured and will be considered by the message handler.

25.4.1.28 DCAN_MSGVAL78 Register (offset = D0h) [reset = 0h]

DCAN_MSGVAL78 is shown in Figure 25-46 and described in Table 25-41.

These registers hold the MsgVal bits of the implemented message objects. By reading out these bits, the CPU can check which message objects are valid. The MsgVal bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

Figure 25-46. DCAN_MSGVAL78 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSGVAL																															
R-0h																															

Table 25-41. DCAN_MSGVAL78 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MSGVAL	R	0h	Message valid bits [96:127] (for all message objects) 0h (R/W) = This message object is ignored by the message handler. 1h (R/W) = This message object is configured and will be considered by the message handler.

25.4.1.29 DCAN_INTMUX12 Register (offset = D8h) [reset = 0h]

DCAN_INTMUX12 is shown in Figure 25-47 and described in Table 25-42.

The IntMux flag determine for each message object, which of the two interrupt lines (DCAN0INT or DCAN1INT) will be asserted when the IntPnd of this message object is set. Both interrupt lines can be globally enabled or disabled by setting or clearing IE0 and IE1 bits in CAN control register. The IntPnd bit of a specific message object can be set or reset by the CPU via the IF1/IF2 interface register sets, or by message handler after reception or successful transmission of a frame. This will also affect the Int0ID resp Int1ID flags in the interrupt register.

Figure 25-47. DCAN_INTMUX12 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">INTMUX</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 25-42. DCAN_INTMUX12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTMUX	R/W	0h	IntMux[31:0] multiplexes IntPnd value to either DCAN0INT or DCAN1INT interrupt lines. The mapping from the bits to the message objects is as follows. Bit 0 -> last implemented message object. Bit 1 -> message object number 1. Bit 2 -> message object number 2. ... Bit 31 -> message object number 31. 0h (R/W) = DCAN0INT line is active if corresponding IntPnd flag is one. 1h (R/W) = DCAN1INT line is active if corresponding IntPnd flag is one.

25.4.1.30 DCAN_INTMUX34 Register (offset = DCh) [reset = 0h]

DCAN_INTMUX34 is shown in Figure 25-48 and described in Table 25-43.

The IntMux flag determine for each message object, which of the two interrupt lines (DCAN0INT or DCAN1INT) will be asserted when the IntPnd of this message object is set. Both interrupt lines can be globally enabled or disabled by setting or clearing IE0 and IE1 bits in CAN control register. The IntPnd bit of a specific message object can be set or reset by the CPU via the IF1/IF2 interface register sets, or by message handler after reception or successful transmission of a frame. This will also affect the Int0ID resp Int1ID flags in the interrupt register.

Figure 25-48. DCAN_INTMUX34 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTMUX																															
R-0h																															

Table 25-43. DCAN_INTMUX34 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTMUX	R/W	0h	IntMux[63:32] multiplexes IntPnd value to either DCAN0INT or DCAN1INT interrupt lines. The mapping from the bits to the message objects is as follows. Bit 0 -> last implemented message object. Bit 1 -> message object number 1. Bit 2 -> message object number 2. ... Bit 31 -> message object number 31. 0x 0h (R/W) = DCAN0INT line is active if corresponding IntPnd flag is one. 1h (R/W) = DCAN1INT line is active if corresponding IntPnd flag is one.

25.4.1.31 DCAN_INTMUX56 Register (offset = E0h) [reset = 0h]

DCAN_INTMUX56 is shown in Figure 25-49 and described in Table 25-44.

The IntMux flag determine for each message object, which of the two interrupt lines (DCAN0INT or DCAN1INT) will be asserted when the IntPnd of this message object is set. Both interrupt lines can be globally enabled or disabled by setting or clearing IE0 and IE1 bits in CAN control register. The IntPnd bit of a specific message object can be set or reset by the CPU via the IF1/IF2 interface register sets, or by message handler after reception or successful transmission of a frame. This will also affect the Int0ID resp Int1ID flags in the interrupt register.

Figure 25-49. DCAN_INTMUX56 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTMUX																															
R-0h																															

Table 25-44. DCAN_INTMUX56 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTMUX	R/W	0h	IntMux[95:64] multiplexes IntPnd value to either DCAN0INT or DCAN1INT interrupt lines. The mapping from the bits to the message objects is as follows. Bit 0 -> last implemented message object. Bit 1 -> message object number 1. Bit 2 -> message object number 2. ... Bit 31 -> message object number 31. 0h (R/W) = DCAN0INT line is active if corresponding IntPnd flag is one. 1h (R/W) = DCAN1INT line is active if corresponding IntPnd flag is one.

25.4.1.32 DCAN_INTMUX78 Register (offset = E4h) [reset = 0h]

DCAN_INTMUX78 is shown in Figure 25-50 and described in Table 25-45.

The IntMux flag determine for each message object, which of the two interrupt lines (DCAN0INT or DCAN1INT) will be asserted when the IntPnd of this message object is set. Both interrupt lines can be globally enabled or disabled by setting or clearing IE0 and IE1 bits in CAN control register. The IntPnd bit of a specific message object can be set or reset by the CPU via the IF1/IF2 interface register sets, or by message handler after reception or successful transmission of a frame. This will also affect the Int0ID resp Int1ID flags in the interrupt register.

Figure 25-50. DCAN_INTMUX78 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTMUX																															
R-0h																															

Table 25-45. DCAN_INTMUX78 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTMUX	R/W	0h	IntMux[127:96] multiplexes IntPnd value to either DCAN0INT or DCAN1INT interrupt lines. The mapping from the bits to the message objects is as follows. Bit 0 -> last implemented message object. Bit 1 -> message object number 1. Bit 2 -> message object number 2. ... Bit 31 -> message object number 31. 0h (R/W) = DCAN0INT line is active if corresponding IntPnd flag is one. 1h (R/W) = DCAN1INT line is active if corresponding IntPnd flag is one.

25.4.1.33 DCAN_IF1CMD Register (offset = 100h) [reset = 0h]

DCAN_IF1CMD is shown in Figure 25-51 and described in Table 25-46.

The IF1 Command Register (IF1CMD) configures and initiates the transfer between the IF1 register sets and the message RAM. It is configurable which portions of the message object should be transferred. A transfer is started when the CPU writes the message number to bits [7:0] of the IF1 command register. With this write operation, the Busy bit is automatically set to '1' to indicate that a transfer is in progress. After 4 to 14 OCP clock cycles, the transfer between the interface register and the message RAM will be completed and the Busy bit is cleared. The maximum number of cycles is needed when the message transfer concurs with a CAN message transmission, acceptance filtering, or message storage. If the CPU writes to both IF1 command registers consecutively (request of a second transfer while first transfer is still in progress), the second transfer will start after the first one has been completed. While Busy bit is one, IF1 register sets are write protected. For debug support, the auto clear functionality of the IF1 command registers (clear of DMAactive flag by r/w) is disabled during Debug/Suspend mode. If an invalid Message Number is written to bits [7:0] of the IF1 command register, the message handler may access an implemented (valid) message object instead.

Figure 25-51. DCAN_IF1CMD Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WR_RD	MASK	ARB	CTRL	CLRINTPND	TXRQST_NEWDAT	DATA_A	DATA_B
0h	0h	0h	0h	0h	0h	0h	0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BUSY	DMAACTIVE	RESERVED					
0h	0h	R-0h					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MESSAGE_NUMBER							
0h							

Table 25-46. DCAN_IF1CMD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23	WR_RD		0h	Write/Read 0h (R/W) = Direction = Read: Transfer direction is from the message object addressed by Message Number (Bits [7:0]) to the IF1 register set. 1h (R/W) = Direction = Write: Transfer direction is from the IF1 register set to the message object addressed by Message Number (Bits [7:0]).
22	MASK		0h	Access mask bits 0h (R/W) = Mask bits will not be changed 1h (R/W) = Direction = Read: The mask bits (identifier mask + MDir + MXtd) will be transferred from the message object addressed by Message Number (Bits [7:0]) to the IF1 register set. Direction = Write: The mask bits (identifier mask + MDir + MXtd) will be transferred from the IF1 register set to the message object addressed by Message Number (Bits [7:0]).
21	ARB		0h	Access arbitration bits 0h (R/W) = Arbitration bits will not be changed 1h (R/W) = Direction = Read: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the message object addressed by Message Number (Bits [7:0]) to the corresponding IF1 register set. Direction = Write: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the IF1 register set to the message object addressed by Message Number (Bits [7:0]).

Table 25-46. DCAN_IF1CMD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	CTRL		0h	Access control bits. If the TxRqst/NewDat bit in this register(Bit [18]) is set, the TxRqst/ NewDat bit in the IF1 message control register will be ignored. 0h (R/W) = Control bits will not be changed 1h (R/W) = Direction = Read: The message control bits will be transferred from the message object addressed by message number (Bits [7:0]) to the IF1 register set. Direction = Write: The message control bits will be transferred from the IF1 register set to the message object addressed by message number (Bits [7:0]).
19	CLRINTPND		0h	Clear interrupt pending bit 0h (R/W) = IntPnd bit will not be changed 1h (R/W) = Direction = Read: Clears IntPnd bit in the message object. Direction = Write: This bit is ignored. Copying of IntPnd flag from IF1 Registers to message RAM can only be controlled by the control flag (Bit [20]).
18	TXRQST_NEWDAT		0h	Access transmission request bit. Note: If a CAN transmission is requested by setting TxRqst/NewDat in this register, the TxRqst/NewDat bits in the message object will be set to one independent of the values in IF1 message control Register. Note: A read access to a message object can be combined with the reset of the control bits IntPnd and NewDat. The values of these bits transferred to the IF1 message control register always reflect the status before resetting them. 0h (R/W) = Direction = Read: NewDat bit will not be changed. Direction = Write: TxRqst/NewDat bit will be handled according to the control bit. 1h (R/W) = Direction = Read: Clears NewDat bit in the message object. Direction = Write: Sets TxRqst/NewDat in message object.
17	DATA_A		0h	Access Data Bytes 0 to 3. 0h (R/W) = Data Bytes 0-3 will not be changed. 1h (R/W) = Direction = Read: The data bytes 0-3 will be transferred from the message object addressed by the Message Number (Bits [7:0]) to the corresponding IF1 register set. Direction = Write: The data bytes 0-3 will be transferred from the IF1 register set to the message object addressed by the Message Number (Bits [7:0]). Note: The duration of the message transfer is independent of the number of bytes to be transferred.
16	DATA_B		0h	Access Data Bytes 4 to 7. 0h (R/W) = Data Bytes 4-7 will not be changed. 1h (R/W) = Direction = Read: The data bytes 4-7 will be transferred from the message object addressed by Message Number (Bits [7:0]) to the corresponding IF1 register set. Direction = Write: The data bytes 4-7 will be transferred from the IF1 register set to the message object addressed by message number (Bits [7:0]). Note: The duration of the message transfer is independent of the number of bytes to be transferred.
15	BUSY		0h	Busy flag. This bit is set to one after the message number has been written to bits 7 to 0. IF1 register set will be write protected. The bit is cleared after read/write action has been finished. 0h (R/W) = No transfer between IF1 register set and message RAM is in progress. 1h (R/W) = Transfer between IF1 register set and message RAM is in progress.

Table 25-46. DCAN_IF1CMD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	DMAACTIVE		0h	Activation of DMA feature for subsequent internal IF1 update. Note: Due to the auto reset feature of the DMAactive bit, this bit has to be set for each subsequent DMA cycle separately. 0h (R/W) = DMA request line is independent of IF1 activities. 1h (R/W) = DMA is requested after completed transfer between IF1 register set and message RAM. The DMA request remains active until the first read or write to one of the IF1 registers; an exception is a write to Message Number (Bits [7:0]) when DMAactive is one.
13-8	RESERVED	R	0h	
7-0	MESSAGE_NUMBER		0h	Number of message object in message RAM which is used for data transfer. 0h (R/W) = Invalid message number. 1h (R/W) = Valid message numbers (value 01 to 80). 80h (R/W) = Valid message number. 81h (R/W) = Invalid message numbers (value 81 to FF). FFh (R/W) = Invalid message numbers.

25.4.1.34 DCAN_IF1MSK Register (offset = 104h) [reset = E0000000h]

DCAN_IF1MSK is shown in Figure 25-52 and described in Table 25-47.

The bits of the IF1 mask registers mirror the mask bits of a message object. While Busy bit of IF1 command register is one, IF1 register set is write protected.

Figure 25-52. DCAN_IF1MSK Register

31	30	29	28	27	26	25	24
MXTD	MDIR	RESERVED	MSK_28:0				
1h	1h	R-1h	0h				
23	22	21	20	19	18	17	16
MSK_28:0							
0h							
15	14	13	12	11	10	9	8
MSK_28:0							
0h							
7	6	5	4	3	2	1	0
MSK_28:0							
0h							

Table 25-47. DCAN_IF1MSK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MXTD		1h	Mask Extended Identifier. When 11 bit (standard) identifiers are used for a message object, the identifiers of received data frames are written into bits ID28 to ID18. For acceptance filtering, only these bits together with mask bits Msk28 to Msk18 are considered. 0h (R/W) = The extended identifier bit (IDE) has no effect on the acceptance filtering. 1h (R/W) = The extended identifier bit (IDE) is used for acceptance filtering.
30	MDIR		1h	Mask Message Direction 0h (R/W) = The message direction bit (Dir) has no effect on the acceptance filtering. 1h (R/W) = The message direction bit (Dir) is used for acceptance filtering.
29	RESERVED	R	1h	
28-0	MSK_28:0		0h	Identifier Mask 0h (R/W) = The corresponding bit in the identifier of the message object is not used for acceptance filtering (don't care). 1h (R/W) = The corresponding bit in the identifier of the message object is used for acceptance filtering.

25.4.1.35 DCAN_IF1ARB Register (offset = 108h) [reset = 0h]

DCAN_IF1ARB is shown in Figure 25-53 and described in Table 25-48.

The bits of the IF1 arbitration registers mirror the arbitration bits of a message object. While Busy bit of IF1 command register is one, IF1 register set is write protected.

Figure 25-53. DCAN_IF1ARB Register

31	30	29	28	27	26	25	24
MSGVAL	XTD	DIR	ID28_TO_ID0				
0h	0h	0h	0h				
23	22	21	20	19	18	17	16
ID28_TO_ID0							
0h							
15	14	13	12	11	10	9	8
ID28_TO_ID0							
0h							
7	6	5	4	3	2	1	0
ID28_TO_ID0							
0h							

Table 25-48. DCAN_IF1ARB Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MSGVAL		0h	Message valid. The CPU should reset the MsgVal bit of all unused Messages Objects during the initialization before it resets bit Init in the CAN control register. This bit must also be reset before the identifier ID28 to ID0, the control bits Xtd, Dir or DLC3 to DLC0 are modified, or if the messages object is no longer required. 0h (R/W) = The message object is ignored by the message handler. 1h (R/W) = The message object is to be used by the message handler.
30	XTD		0h	Extended identifier 0h (R/W) = The 11-bit (standard) Identifier is used for this message object. 1h (R/W) = The 29-bit (extended) Identifier is used for this message object.
29	DIR		0h	Message direction 0h (R/W) = Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, this message is stored in this message object. 1h (R/W) = Direction = transmit: On TxRqst, the respective message object is transmitted as a data frame. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = 1).
28-0	ID28_TO_ID0		0h	Message identifier. ID28 to ID0 is equal to 29 bit identifier (extended frame) ID28 to ID18 is equal to 11 bit identifier (standard frame)

25.4.1.36 DCAN_IF1MCTL Register (offset = 10Ch) [reset = 0h]

DCAN_IF1MCTL is shown in Figure 25-54 and described in Table 25-49.

The bits of the IF1 message control registers mirror the message control bits of a message object. While Busy bit of IF1 command register is one, IF1 register set is write protected.

Figure 25-54. DCAN_IF1MCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
NEWDAT	MSGLST	INTPND	UMASK	TXIE	RXIE	RMTEN	TXRQST
0h	0h	0h	0h	0h	0h	0h	0h
7	6	5	4	3	2	1	0
EOB	RESERVED			DLC			
0h	R-0h			0h			

Table 25-49. DCAN_IF1MCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NEWDAT		0h	New data 0h (R/W) = No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. 1h (R/W) = The message handler or the CPU has written new data into the data portion of this message object.
14	MSGLST		0h	Message lost (only valid for message objects with direction = receive) 0h (R/W) = No message lost since the last time when this bit was reset by the CPU. 1h (R/W) = The message handler stored a new message into this object when NewDat was still set, so the previous message has been overwritten.
13	INTPND		0h	Interrupt pending 0h (R/W) = This message object is not the source of an interrupt. 1h (R/W) = This message object is the source of an interrupt. The Interrupt Identifier in the interrupt register will point to this message object if there is no other interrupt source with higher priority.
12	UMASK		0h	Use acceptance mask. If the UMask bit is set to one, the message object's mask bits have to be programmed during initialization of the message object before MsgVal is set to one. 0h (R/W) = Mask ignored 1h (R/W) = Use mask (Msk[28:0], MXtd, and MDir) for acceptance filtering
11	TXIE		0h	Transmit interrupt enable 0h (R/W) = IntPnd will not be triggered after the successful transmission of a frame. 1h (R/W) = IntPnd will be triggered after the successful transmission of a frame.

Table 25-49. DCAN_IF1MCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	RXIE		0h	Receive interrupt enable 0h (R/W) = IntPnd will not be triggered after the successful reception of a frame. 1h (R/W) = IntPnd will be triggered after the successful reception of a frame.
9	RMTEN		0h	Remote enable 0h (R/W) = At the reception of a remote frame, TxRqst is not changed. 1h (R/W) = At the reception of a remote frame, TxRqst is set.
8	TXRQST		0h	Transmit request 0h (R/W) = This message object is not waiting for a transmission. 1h (R/W) = The transmission of this message object is requested and is not yet done.
7	EOB		0h	Data frame has 0 to 8 data bits. Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to one. 0h (R/W) = Data frame has 8 data bytes. 1h (R/W) = Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.
6-4	RESERVED	R	0h	
3-0	DLC		0h	Data length code. Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message. 0x 0-0x 8: Data frame has 0-8 data bytes. 0x 9-0x 15: Data frame has 8 data bytes.

25.4.1.37 DCAN_IF1DATA Register (offset = 110h) [reset = 0h]

DCAN_IF1DATA is shown in Figure 25-55 and described in Table 25-50.

The data bytes of CAN messages are stored in the IF1 registers in the following order: (1) In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. (2) In CAN's serial bit stream, the MSB of each byte will be transmitted first.

Figure 25-55. DCAN_IF1DATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_3								DATA_2								DATA_1								DATA_0							
0h								0h								0h								0h							

Table 25-50. DCAN_IF1DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	DATA_3		0h	Data 3.
23-16	DATA_2		0h	Data 2.
15-8	DATA_1		0h	Data 1.
7-0	DATA_0		0h	Data 0.

25.4.1.38 DCAN_IF1DATB Register (offset = 114h) [reset = 0h]

DCAN_IF1DATB is shown in Figure 25-56 and described in Table 25-51.

The data bytes of CAN messages are stored in the IF1 registers in the following order: (1) In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. (2) In CAN's serial bit stream, the MSB of each byte will be transmitted first.

Figure 25-56. DCAN_IF1DATB Register

31-24	23-16	15-8	7-0
DATA_7	DATA_6	DATA_5	DATA_4
0h	0h	0h	0h

Table 25-51. DCAN_IF1DATB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	DATA_7		0h	Data 7.
23-16	DATA_6		0h	Data 6.
15-8	DATA_5		0h	Data 5.
7-0	DATA_4		0h	Data 4.

25.4.1.39 DCAN_IF2CMD Register (offset = 120h) [reset = 0h]

DCAN_IF2CMD is shown in Figure 25-57 and described in Table 25-52.

The IF2 Command Register (IF1CMD) configures and initiates the transfer between the IF2 register sets and the message RAM. It is configurable which portions of the message object should be transferred. A transfer is started when the CPU writes the message number to bits [7:0] of the IF2 command register. With this write operation, the Busy bit is automatically set to '1' to indicate that a transfer is in progress. After 4 to 14 OCP clock cycles, the transfer between the interface register and the message RAM will be completed and the Busy bit is cleared. The maximum number of cycles is needed when the message transfer concurs with a CAN message transmission, acceptance filtering, or message storage. If the CPU writes to both IF2 command registers consecutively (request of a second transfer while first transfer is still in progress), the second transfer will start after the first one has been completed. While Busy bit is one, IF2 register sets are write protected. For debug support, the auto clear functionality of the IF2 command registers (clear of DMAactive flag by r/w) is disabled during Debug/Suspend mode. If an invalid Message Number is written to bits [7:0] of the IF2 command register, the message handler may access an implemented (valid) message object instead.

Figure 25-57. DCAN_IF2CMD Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>WR_RD</td><td>MASK</td><td>ARB</td><td>CTRL</td><td>CLRINTPND</td><td>TXRQST_NEWDAT</td><td>DATA_A</td><td>DATA_B</td></tr>
<tr><td>0h</td><td>0h</td><td>0h</td><td>0h</td><td>0h</td><td>0h</td><td>0h</td><td>0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>BUSY</td><td>DMAACTIVE</td><td colspan="6">RESERVED</td></tr>
<tr><td>0h</td><td>0h</td><td colspan="6">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">MESSAGE_NUMBER</td></tr>
<tr><td colspan="8">0h</td></tr>
</table>

Table 25-52. DCAN_IF2CMD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23	WR_RD		0h	Write/Read 0h (R/W) = Direction = Read: Transfer direction is from the message object addressed by Message Number (Bits [7:0]) to the IF2 register set. 1h (R/W) = Direction = Write: Transfer direction is from the IF2 register set to the message object addressed by Message Number (Bits [7:0]).
22	MASK		0h	Access mask bits 0h (R/W) = Mask bits will not be changed 1h (R/W) = Direction = Read: The mask bits (identifier mask + MDir + MXtd) will be transferred from the message object addressed by Message Number (Bits [7:0]) to the IF2 register set. Direction = Write: The mask bits (identifier mask + MDir + MXtd) will be transferred from the IF2 register set to the message object addressed by Message Number (Bits [7:0]).
21	ARB		0h	Access arbitration bits 0h (R/W) = Arbitration bits will not be changed 1h (R/W) = Direction = Read: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the message object addressed by Message Number (Bits [7:0]) to the corresponding IF2 register set. Direction = Write: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the IF2 register set to the message object addressed by Message Number (Bits [7:0]).

Table 25-52. DCAN_IF2CMD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	CTRL		0h	Access control bits. If the TxRqst/NewDat bit in this register(Bit [18]) is set, the TxRqst/ NewDat bit in the IF2 message control register will be ignored. 0h (R/W) = Control bits will not be changed 1h (R/W) = Direction = Read: The message control bits will be transferred from the message object addressed by message number (Bits [7:0]) to the IF2 register set. Direction = Write: The message control bits will be transferred from the IF2 register set to the message object addressed by message number (Bits [7:0]).
19	CLRINTPND		0h	Clear interrupt pending bit 0h (R/W) = IntPnd bit will not be changed 1h (R/W) = Direction = Read: Clears IntPnd bit in the message object. Direction = Write: This bit is ignored. Copying of IntPnd flag from IF2 Registers to message RAM can only be controlled by the control flag (Bit [20]).
18	TXRQST_NEWDAT		0h	Access transmission request bit. Note: If a CAN transmission is requested by setting TxRqst/NewDat in this register, the TxRqst/NewDat bits in the message object will be set to one independent of the values in IF2 message control Register. Note: A read access to a message object can be combined with the reset of the control bits IntPnd and NewDat. The values of these bits transferred to the IF2 message control register always reflect the status before resetting them. 0h (R/W) = Direction = Read: NewDat bit will not be changed. Direction = Write: TxRqst/NewDat bit will be handled according to the control bit. 1h (R/W) = Direction = Read: Clears NewDat bit in the message object. Direction = Write: Sets TxRqst/NewDat in message object.
17	DATA_A		0h	Access Data Bytes 0 to 3. 0h (R/W) = Data Bytes 0-3 will not be changed. 1h (R/W) = Direction = Read: The data bytes 0-3 will be transferred from the message object addressed by the Message Number (Bits [7:0]) to the corresponding IF2 register set. Direction = Write: The data bytes 0-3 will be transferred from the IF2 register set to the message object addressed by the Message Number (Bits [7:0]). Note: The duration of the message transfer is independent of the number of bytes to be transferred.
16	DATA_B		0h	Access Data Bytes 4 to 7. 0h (R/W) = Data Bytes 4-7 will not be changed. 1h (R/W) = Direction = Read: The data bytes 4-7 will be transferred from the message object addressed by Message Number (Bits [7:0]) to the corresponding IF2 register set. Direction = Write: The data bytes 4-7 will be transferred from the IF2 register set to the message object addressed by message number (Bits [7:0]). Note: The duration of the message transfer is independent of the number of bytes to be transferred.
15	BUSY		0h	Busy flag. This bit is set to one after the message number has been written to bits 7 to 0. IF2 register set will be write protected. The bit is cleared after read/write action has been finished. 0h (R/W) = No transfer between IF2 register set and message RAM is in progress. 1h (R/W) = Transfer between IF2 register set and message RAM is in progress.

Table 25-52. DCAN_IF2CMD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	DMAACTIVE		0h	Activation of DMA feature for subsequent internal IF2 update. Note: Due to the auto reset feature of the DMAactive bit, this bit has to be set for each subsequent DMA cycle separately. 0h (R/W) = DMA request line is independent of IF2 activities. 1h (R/W) = DMA is requested after completed transfer between IF2 register set and message RAM. The DMA request remains active until the first read or write to one of the IF2 registers; an exception is a write to Message Number (Bits [7:0]) when DMAactive is one.
13-8	RESERVED	R	0h	
7-0	MESSAGE_NUMBER		0h	Number of message object in message RAM which is used for data transfer. 0h (R/W) = Invalid message number. 1h (R/W) = Valid message numbers (values 01 to 80). 80h (R/W) = Valid message number. 81h (R/W) = Invalid message numbers (values 81 to FF). FFh (R/W) = Invalid message number.

25.4.1.40 DCAN_IF2MSK Register (offset = 124h) [reset = E0000000h]

DCAN_IF2MSK is shown in Figure 25-58 and described in Table 25-53.

The bits of the IF2 mask registers mirror the mask bits of a message object. While Busy bit of IF2 command register is one, IF2 register set is write protected.

Figure 25-58. DCAN_IF2MSK Register

31	30	29	28	27	26	25	24
MXTD	MDIR	RESERVED	MSK_28:0				
1h	1h	R-1h	0h				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
MSK_28:0							
0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MSK_28:0							
0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MSK_28:0							
0h							

Table 25-53. DCAN_IF2MSK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MXTD		1h	Mask Extended Identifier. When 11 bit (standard) identifiers are used for a message object, the identifiers of received data frames are written into bits ID28 to ID18. For acceptance filtering, only these bits together with mask bits Msk28 to Msk18 are considered. 0h (R/W) = The extended identifier bit (IDE) has no effect on the acceptance filtering. 1h (R/W) = The extended identifier bit (IDE) is used for acceptance filtering.
30	MDIR		1h	Mask Message Direction 0h (R/W) = The message direction bit (Dir) has no effect on the acceptance filtering. 1h (R/W) = The message direction bit (Dir) is used for acceptance filtering.
29	RESERVED	R	1h	
28-0	MSK_28:0		0h	Identifier Mask 0h (R/W) = The corresponding bit in the identifier of the message object is not used for acceptance filtering (don't care). 1h (R/W) = The corresponding bit in the identifier of the message object is used for acceptance filtering.

25.4.1.41 DCAN_IF2ARB Register (offset = 128h) [reset = 0h]

DCAN_IF2ARB is shown in Figure 25-59 and described in Table 25-54.

The bits of the IF2 arbitration registers mirror the arbitration bits of a message object. While Busy bit of IF2 command register is one, IF2 register set is write protected.

Figure 25-59. DCAN_IF2ARB Register

31	30	29	28	27	26	25	24
MSGVAL	XTD	DIR	ID28_TO_ID0				
0h	0h	0h	0h				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ID28_TO_ID0							
0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ID28_TO_ID0							
0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ID28_TO_ID0							
0h							

Table 25-54. DCAN_IF2ARB Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MSGVAL		0h	Message valid. The CPU should reset the MsgVal bit of all unused Messages Objects during the initialization before it resets bit Init in the CAN control register. This bit must also be reset before the identifier ID28 to ID0, the control bits Xtd, Dir or DLC3 to DLC0 are modified, or if the messages object is no longer required. 0h (R/W) = The message object is ignored by the message handler. 1h (R/W) = The message object is to be used by the message handler.
30	XTD		0h	Extended identifier 0h (R/W) = The 11-bit (standard) Identifier is used for this message object. 1h (R/W) = The 29-bit (extended) Identifier is used for this message object.
29	DIR		0h	Message direction 0h (R/W) = Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, this message is stored in this message object. 1h (R/W) = Direction = transmit: On TxRqst, the respective message object is transmitted as a data frame. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = 1).
28-0	ID28_TO_ID0		0h	Message identifier. ID28 to ID0 is equal to 29-bit identifier (extended frame) ID28 to ID18 is equal to 11-bit identifier (standard frame)

25.4.1.42 DCAN_IF2MCTL Register (offset = 12Ch) [reset = 0h]

DCAN_IF2MCTL is shown in Figure 25-60 and described in Table 25-55.

The bits of the IF2 message control registers mirror the message control bits of a message object. While Busy bit of IF2 command register is one, IF2 register set is write protected.

Figure 25-60. DCAN_IF2MCTL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>NEWDAT</td><td>MSGLST</td><td>INTPND</td><td>UMASK</td><td>TXIE</td><td>RXIE</td><td>RMTEN</td><td>TXRQST</td></tr>
<tr><td>0h</td><td>0h</td><td>0h</td><td>0h</td><td>0h</td><td>0h</td><td>0h</td><td>0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>EOB</td><td colspan="3">RESERVED</td><td colspan="4">DLC</td></tr>
<tr><td>0h</td><td colspan="3">R-0h</td><td colspan="4">0h</td></tr>
</table>

Table 25-55. DCAN_IF2MCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NEWDAT		0h	New data 0h (R/W) = No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. 1h (R/W) = The message handler or the CPU has written new data into the data portion of this message object.
14	MSGLST		0h	Message lost (only valid for message objects with direction = receive) 0h (R/W) = No message lost since the last time when this bit was reset by the CPU. 1h (R/W) = The message handler stored a new message into this object when NewDat was still set, so the previous message has been overwritten.
13	INTPND		0h	Interrupt pending 0h (R/W) = This message object is not the source of an interrupt. 1h (R/W) = This message object is the source of an interrupt. The Interrupt Identifier in the interrupt register will point to this message object if there is no other interrupt source with higher priority.
12	UMASK		0h	Use acceptance mask. If the UMask bit is set to one, the message object's mask bits have to be programmed during initialization of the message object before MsgVal is set to one. 0h (R/W) = Mask ignored 1h (R/W) = Use mask (Msk[28:0], MXtd, and MDir) for acceptance filtering
11	TXIE		0h	Transmit interrupt enable 0h (R/W) = IntPnd will not be triggered after the successful transmission of a frame. 1h (R/W) = IntPnd will be triggered after the successful transmission of a frame.

Table 25-55. DCAN_IF2MCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	RXIE		0h	Receive interrupt enable 0h (R/W) = IntPnd will not be triggered after the successful reception of a frame. 1h (R/W) = IntPnd will be triggered after the successful reception of a frame.
9	RMTEN		0h	Remote enable 0h (R/W) = At the reception of a remote frame, TxRqst is not changed. 1h (R/W) = At the reception of a remote frame, TxRqst is set.
8	TXRQST		0h	Transmit request 0h (R/W) = This message object is not waiting for a transmission. 1h (R/W) = The transmission of this message object is requested and is not yet done.
7	EOB		0h	Data frame has 0 to 8 data bits. Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to one. 0h (R/W) = Data frame has 8 data bytes. 1h (R/W) = Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.
6-4	RESERVED	R	0h	
3-0	DLC		0h	Data length code. Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message. 0x 0-0x 8: Data frame has 0-8 data bytes. 0x 9-0x 15: Data frame has 8 data bytes.

25.4.1.43 DCAN_IF2DATA Register (offset = 130h) [reset = 0h]

DCAN_IF2DATA is shown in Figure 25-61 and described in Table 25-56.

The data bytes of CAN messages are stored in the IF2 registers in the following order: (1) In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. (2) In CAN's serial bit stream, the MSB of each byte will be transmitted first.

Figure 25-61. DCAN_IF2DATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_3								DATA_2								DATA_1								DATA_0							
0h								0h								0h								0h							

Table 25-56. DCAN_IF2DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	DATA_3		0h	Data 3.
23-16	DATA_2		0h	Data 2.
15-8	DATA_1		0h	Data 1.
7-0	DATA_0		0h	Data 0.

25.4.1.44 DCAN_IF2DATB Register (offset = 134h) [reset = 0h]

DCAN_IF2DATB is shown in Figure 25-62 and described in Table 25-57.

The data bytes of CAN messages are stored in the IF2 registers in the following order: (1) In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. (2) In CAN's serial bit stream, the MSB of each byte will be transmitted first.

Figure 25-62. DCAN_IF2DATB Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">DATA_7</td><td colspan="8">DATA_6</td><td colspan="8">DATA_5</td><td colspan="8">DATA_4</td></tr>
<tr><td colspan="8">0h</td><td colspan="8">0h</td><td colspan="8">0h</td><td colspan="8">0h</td></tr>
</table>

Table 25-57. DCAN_IF2DATB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	DATA_7		0h	Data 7.
23-16	DATA_6		0h	Data 6.
15-8	DATA_5		0h	Data 5.
7-0	DATA_4		0h	Data 4.

25.4.1.45 DCAN_IF3OBS Register (offset = 140h) [reset = 0h]

DCAN_IF3OBS is shown in Figure 25-63 and described in Table 25-58.

The IF3 register set can automatically be updated with received message objects without the need to initiate the transfer from message RAM by CPU. The observation flags (Bits [4:0]) in the IF3 observation register are used to determine, which data sections of the IF3 interface register set have to be read in order to complete a DMA read cycle. After all marked data sections are read, the DCAN is enabled to update the IF3 interface register set with new data. Any access order of single bytes or half-words is supported. When using byte or half-word accesses, a data section is marked as completed, if all bytes are read. Note: If IF3 Update Enable is used and no Observation flag is set, the corresponding message objects will be copied to IF3 without activating the DMA request line and without waiting for DMA read accesses. A write access to this register aborts a pending DMA cycle by resetting the DMA line and enables updating of IF3 interface register set with new data. To avoid data inconsistency, the DMA controller should be disabled before reconfiguring IF3 observation register. The status of the current read-cycle can be observed via status flags (Bits [12:8]). If an interrupt line is available for IF3, an interrupt will be generated by IF3Upd flag. See the device-specific data sheet for the availability of this interrupt source. With this interrupt, the observation status bits and the IF3Upd bit could be used by the application to realize the notification about new IF3 content in polling or interrupt mode.

Figure 25-63. DCAN_IF3OBS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
IF3_UPD	RESERVED		IF3_SDB	IF3_SDA	IF3_SC	IF3_SA	IF3_SM
R-0h	R-0h		R-0h	R-0h	R-0h	R-0h	R-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED			DATAB	DATAA	CTRL	ARB	MASK
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 25-58. DCAN_IF3OBS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	IF3_UPD	R	0h	IF3 Update Data 0h (R/W) = No new data has been loaded since last IF3 read. 1h (R/W) = New data has been loaded since last IF3 read.
14-13	RESERVED	R	0h	
12	IF3_SDB	R	0h	IF3 Status of Data B read access 0h (R/W) = All Data B bytes are already read out, or are not marked to be read. 1h (R/W) = Data B section has still data to be read out.
11	IF3_SDA	R	0h	IF3 Status of Data A read access 0h (R/W) = All Data A bytes are already read out, or are not marked to be read. 1h (R/W) = Data A section has still data to be read out.
10	IF3_SC	R	0h	IF3 Status of control bits read access 0h (R/W) = All control section bytes are already read out, or are not marked to be read. 1h (R/W) = Control section has still data to be read out.

Table 25-58. DCAN_IF3OBS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	IF3_SA	R	0h	IF3 Status of Arbitration data read access 0h (R/W) = All Arbitration data bytes are already read out, or are not marked to be read. 1h (R/W) = Arbitration section has still data to be read out.
8	IF3_SM	R	0h	IF3 Status of Mask data read access 0h (R/W) = All mask data bytes are already read out, or are not marked to be read. 1h (R/W) = Mask section has still data to be read out.
7-5	RESERVED	R	0h	
4	DATAB	R/W	0h	Data B read observation 0h (R/W) = Data B section has not to be read. 1h (R/W) = Data B section has to be read to enable next IF3 update.
3	DATAA	R/W	0h	Data A read observation 0h (R/W) = Data A section has not to be read. 1h (R/W) = Data A section has to be read to enable next IF3 update.
2	CTRL	R/W	0h	Ctrl read observation 0h (R/W) = Ctrl section has not to be read. 1h (R/W) = Ctrl section has to be read to enable next IF3 update.
1	ARB	R/W	0h	Arbitration data read observation 0h (R/W) = Arbitration data has not to be read. 1h (R/W) = Arbitration data has to be read to enable next IF3 update.
0	MASK	R/W	0h	Mask data read observation 0h (R/W) = Mask data has not to be read. 1h (R/W) = Mask data has to be read to enable next IF3 update.

25.4.1.46 DCAN_IF3MSK Register (offset = 144h) [reset = E0000000h]

DCAN_IF3MSK is shown in Figure 25-64 and described in Table 25-59.

Figure 25-64. DCAN_IF3MSK Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>MXTD</td><td>MDIR</td><td>RESERVED</td><td colspan="5">MSK_28:0</td></tr>
<tr><td>R-1h</td><td>R-1h</td><td>R-1h</td><td colspan="5">0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">MSK_28:0</td></tr>
<tr><td colspan="8">0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">MSK_28:0</td></tr>
<tr><td colspan="8">0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">MSK_28:0</td></tr>
<tr><td colspan="8">0h</td></tr>
</table>

Table 25-59. DCAN_IF3MSK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MXTD	R	1h	Mask Extended Identifier. When 11 bit (standard) identifiers are used for a message object, the identifiers of received data frames are written into bits ID28 to ID18. For acceptance filtering, only these bits together with mask bits Msk28 to Msk18 are considered. 0h (R/W) = The extended identifier bit (IDE) has no effect on the acceptance filtering. 1h (R/W) = The extended identifier bit (IDE) is used for acceptance filtering.
30	MDIR	R	1h	Mask Message Direction 0h (R/W) = The message direction bit (Dir) has no effect on the acceptance filtering. 1h (R/W) = The message direction bit (Dir) is used for acceptance filtering.
29	RESERVED	R	1h	
28-0	MSK_28:0		0h	Identifier Mask 0h (R/W) = The corresponding bit in the identifier of the message object is not used for acceptance filtering (don't care). 1h (R/W) = The corresponding bit in the identifier of the message object is used for acceptance filtering.

25.4.1.47 DCAN_IF3ARB Register (offset = 148h) [reset = 0h]

DCAN_IF3ARB is shown in Figure 25-65 and described in Table 25-60.

Figure 25-65. DCAN_IF3ARB Register

31	30	29	28	27	26	25	24
MSGVAL	XTD	DIR	ID28_TO_ID0				
R-0h	R-0h	R-0h	R-0h				
23	22	21	20	19	18	17	16
ID28_TO_ID0							
R-0h							
15	14	13	12	11	10	9	8
ID28_TO_ID0							
R-0h							
7	6	5	4	3	2	1	0
ID28_TO_ID0							
R-0h							

Table 25-60. DCAN_IF3ARB Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MSGVAL	R	0h	Message Valid. The CPU should reset the MsgVal bit of all unused Messages Objects during the initialization before it resets bit Init in the CAN control register. This bit must also be reset before the identifier ID28 to ID0, the control bits Xtd, Dir or DLC3 to DLC0 are modified, or if the messages object is no longer required. 0h (R/W) = The message object is ignored by the message handler. 1h (R/W) = The message object is to be used by the message handler.
30	XTD	R	0h	Extended Identifier 0h (R/W) = The 11-bit (standard) Identifier is used for this message object. 1h (R/W) = The 29-bit (extended) Identifier is used for this message object.
29	DIR	R	0h	Message Direction 0h (R/W) = Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, this message is stored in this message object. 1h (R/W) = Direction = transmit: On TxRqst, the respective message object is transmitted as a data frame. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = 1).
28-0	ID28_TO_ID0	R	0h	Message Identifier. ID28 to ID0 is equal to 29 bit Identifier (extended frame). ID28 to ID18 is equal to 11 bit Identifier (standard frame).

25.4.1.48 DCAN_IF3MCTL Register (offset = 14Ch) [reset = 0h]

DCAN_IF3MCTL is shown in Figure 25-66 and described in Table 25-61.

Figure 25-66. DCAN_IF3MCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
NEWDAT	MSGLST	INTPND	UMASK	TXIE	RXIE	RMTEN	TXRQST
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
EOB	RESERVED			DLC			
R-0h	R-0h			R-0h			

Table 25-61. DCAN_IF3MCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NEWDAT	R	0h	New Data 0h (R/W) = No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. 1h (R/W) = The message handler or the CPU has written new data into the data portion of this message object.
14	MSGLST	R	0h	Message Lost (only valid for message objects with direction = receive) 0h (R/W) = No message lost since the last time when this bit was reset by the CPU. 1h (R/W) = The message handler stored a new message into this object when NewDat was still set, so the previous message has been overwritten.
13	INTPND	R	0h	Interrupt Pending 0h (R/W) = This message object is not the source of an interrupt. 1h (R/W) = This message object is the source of an interrupt. The Interrupt Identifier in the interrupt register will point to this message object if there is no other interrupt source with higher priority.
12	UMASK	R	0h	Use Acceptance Mask 0h (R/W) = Mask ignored 1h (R/W) = Use mask (Msk[28:0], MXtd, and MDir) for acceptance filtering. If the UMask bit is set to one, the message object's mask bits have to be programmed during initialization of the message object before MsgVal is set to one.
11	TXIE	R	0h	Transmit Interrupt enable 0h (R/W) = IntPnd will not be triggered after the successful transmission of a frame. 1h (R/W) = IntPnd will be triggered after the successful transmission of a frame.
10	RXIE	R	0h	Receive Interrupt enable 0h (R/W) = IntPnd will not be triggered after the successful reception of a frame. 1h (R/W) = IntPnd will be triggered after the successful reception of a frame.

Table 25-61. DCAN_IF3MCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	RMTEN	R	0h	Remote enable 0h (R/W) = At the reception of a remote frame, TxRqst is not changed. 1h (R/W) = At the reception of a remote frame, TxRqst is set.
8	TXRQST	R	0h	Transmit Request 0h (R/W) = This message object is not waiting for a transmission. 1h (R/W) = The transmission of this message object is requested and is not yet done.
7	EOB	R	0h	Data frame has 0 to 8 data bits. Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to one. 0h (R/W) = Data frame has 8 data bytes. 1h (R/W) = Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.
6-4	RESERVED	R	0h	
3-0	DLC	R	0h	Data Length Code. Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message. 0x 0-0x 8: Data frame has 0-8 data bytes. 0x 9-0x 15: Data frame has 8 data bytes.

25.4.1.49 DCAN_IF3DATA Register (offset = 150h) [reset = 0h]

DCAN_IF3DATA is shown in Figure 25-67 and described in Table 25-62.

The data bytes of CAN messages are stored in the IF3 registers in the following order. In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first.

Figure 25-67. DCAN_IF3DATA Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">DATA_3</td><td colspan="8">DATA_2</td><td colspan="8">DATA_1</td><td colspan="8">DATA_0</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="8">R-0h</td><td colspan="8">R-0h</td><td colspan="8">R-0h</td></tr>
</table>

Table 25-62. DCAN_IF3DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	DATA_3	R	0h	Data 3.
23-16	DATA_2	R	0h	Data 2.
15-8	DATA_1	R	0h	Data 1.
7-0	DATA_0	R	0h	Data 0.

25.4.1.50 DCAN_IF3DATB Register (offset = 154h) [reset = 0h]

DCAN_IF3DATB is shown in Figure 25-68 and described in Table 25-63.

The data bytes of CAN messages are stored in the IF3 registers in the following order. In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first.

Figure 25-68. DCAN_IF3DATB Register

31-24	23-16	15-8	7-0
DATA_7	DATA_6	DATA_5	DATA_4
R-0h	R-0h	R-0h	R-0h

Table 25-63. DCAN_IF3DATB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	DATA_7	R	0h	Data 7.
23-16	DATA_6	R	0h	Data 6.
15-8	DATA_5	R	0h	Data 5.
7-0	DATA_4	R	0h	Data 4.

25.4.1.51 DCAN_IF3UPD12 Register (offset = 160h) [reset = 0h]

DCAN_IF3UPD12 is shown in Figure 25-69 and described in Table 25-64.

The automatic update functionality of the IF3 register set can be configured for each message object. A message object is enabled for automatic IF3 update, if the dedicated IF3UpdEn flag is set. This means that an active NewDat flag of this message object (e.g due to reception of a CAN frame) will trigger an automatic copy of the whole message object to IF3 register set. IF3 Update enable should not be set for transmit objects.

Figure 25-69. DCAN_IF3UPD12 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">IF3UPDEN</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 25-64. DCAN_IF3UPD12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IF3UPDEN	R/W	0h	IF3 Update Enabled bits [31:0] (for all message objects) 0h (R/W) = Automatic IF3 update is disabled for this message object. 1h (R/W) = Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.

25.4.1.52 DCAN_IF3UPD34 Register (offset = 164h) [reset = 0h]

DCAN_IF3UPD34 is shown in Figure 25-70 and described in Table 25-65.

The automatic update functionality of the IF3 register set can be configured for each message object. A message object is enabled for automatic IF3 update, if the dedicated IF3UpdEn flag is set. This means that an active NewDat flag of this message object (e.g due to reception of a CAN frame) will trigger an automatic copy of the whole message object to IF3 register set. IF3 Update enable should not be set for transmit objects.

Figure 25-70. DCAN_IF3UPD34 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF3UPDEN																															
R/W-0h																															

Table 25-65. DCAN_IF3UPD34 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IF3UPDEN	R/W	0h	IF3 Update Enabled bits [32:63] (for all message objects) 0h (R/W) = Automatic IF3 update is disabled for this message object. 1h (R/W) = Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.

25.4.1.53 DCAN_IF3UPD56 Register (offset = 168h) [reset = 0h]

DCAN_IF3UPD56 is shown in Figure 25-71 and described in Table 25-66.

The automatic update functionality of the IF3 register set can be configured for each message object. A message object is enabled for automatic IF3 update, if the dedicated IF3UpdEn flag is set. This means that an active NewDat flag of this message object (e.g due to reception of a CAN frame) will trigger an automatic copy of the whole message object to IF3 register set. IF3 Update enable should not be set for transmit objects.

Figure 25-71. DCAN_IF3UPD56 Register

31–0
IF3UPDEN
R/W-0h

Table 25-66. DCAN_IF3UPD56 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IF3UPDEN	R/W	0h	IF3 Update Enabled bits [64:95] (for all message objects) 0h (R/W) = Automatic IF3 update is disabled for this message object. 1h (R/W) = Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.

25.4.1.54 DCAN_IF3UPD78 Register (offset = 16Ch) [reset = 0h]

DCAN_IF3UPD78 is shown in Figure 25-72 and described in Table 25-67.

The automatic update functionality of the IF3 register set can be configured for each message object. A message object is enabled for automatic IF3 update, if the dedicated IF3UpdEn flag is set. This means that an active NewDat flag of this message object (e.g due to reception of a CAN frame) will trigger an automatic copy of the whole message object to IF3 register set. IF3 Update enable should not be set for transmit objects.

Figure 25-72. DCAN_IF3UPD78 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF3UPDEN																															
R/W-0h																															

Table 25-67. DCAN_IF3UPD78 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IF3UPDEN	R/W	0h	IF3 Update Enabled bits [96:127] (for all message objects) 0h (R/W) = Automatic IF3 update is disabled for this message object. 1h (R/W) = Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.

25.4.1.55 DCAN_TIOC Register (offset = 1E0h) [reset = 0h]

DCAN_TIOC is shown in Figure 25-73 and described in Table 25-68.

The CAN_TX pin of the DCAN module can be used as general purpose IO pin if CAN function is not needed. The values of the IO control registers are only writable if Init bit of the CAN control register is set. The OD, Func, Dir and Out bits of the CAN TX IO control register are forced to certain values when Init bit of CAN control register is reset (see bit descriptions).

Figure 25-73. DCAN_TIOC Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="5">RESERVED</td><td>PU</td><td>PD</td><td>OD</td></tr>
<tr><td colspan="5">R-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td>FUNC</td><td>DIR</td><td>OUT</td><td>IN</td></tr>
<tr><td colspan="4">R-0h</td><td>0h</td><td>0h</td><td>0h</td><td>R-0h</td></tr>
</table>

Table 25-68. DCAN_TIOC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	
18	PU	R/W	0h	CAN_TX pull up/pull down select. This bit is only active when CAN_TX is configured to be an input. 0h (R/W) = CAN_TX pull down is selected, when pull logic is active (PD = 0). 1h (R/W) = CAN_TX pull up is selected, when pull logic is active (PD = 0).
17	PD	R/W	0h	CAN_TX pull disable. This bit is only active when CAN_TX is configured to be an input. 0h (R/W) = CAN_TX pull is active 1h (R/W) = CAN_TX pull is disabled
16	OD		0h	CAN_TX open drain enable. This bit is only active when CAN_TX is configured to be in GIO mode (TIOC.Func=0). Forced to '0' if Init bit of CAN control register is reset. 0h (R/W) = The CAN_TX pin is configured in push/pull mode. 1h (R/W) = The CAN_TX pin is configured in open drain mode.
15-4	RESERVED	R	0h	
3	FUNC		0h	CAN_TX function. This bit changes the function of the CAN_TX pin. Forced to '1' if Init bit of CAN control register is reset. 0h (R/W) = CAN_TX pin is in GIO mode. 1h (R/W) = CAN_TX pin is in functional mode (as an output to transmit CAN data).
2	DIR		0h	CAN_TX data direction. This bit controls the direction of the CAN_TX pin when it is configured to be in GIO mode only (TIOC.Func=0). Forced to '1' if Init bit of CAN control register is reset. 0h (R/W) = The CAN_TX pin is an input. 1h (R/W) = The CAN_TX pin is an output

Table 25-68. DCAN_TIOC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	OUT		0h	CAN_TX data out write. This bit is only active when CAN_TX pin is configured to be in GIO mode (TIOC.Func = 0) and configured to be an output pin (TIOC.Dir = 1). The value of this bit indicates the value to be output to the CAN_TX pin. Forced to Tx output of the CAN core, if Init bit of CAN control register is reset. 0h (R/W) = The CAN_TX pin is driven to logic low 1h (R/W) = The CAN_TX pin is driven to logic high
0	IN	R	0h	CAN_TX data in. Note: When CAN_TX pin is connected to a CAN transceiver, an external pullup resistor has to be used to ensure that the CAN bus will not be disturbed (e.g. while reset of the DCAN module). 0h (R/W) = The CAN_TX pin is at logic low 1h (R/W) = The CAN_TX pin is at logic high

25.4.1.56 DCAN_RIOC Register (offset = 1E4h) [reset = 0h]

DCAN_RIOC is shown in Figure 25-74 and described in Table 25-69.

The CAN_RX pin of the DCAN module can be used as general purpose IO pin if CAN function is not needed. The values of the IO control registers are writable only if Init bit of CAN control register is set. The OD, Func and Dir bits of the CAN RX IO control register are forced to certain values when the Init bit of CAN control register is reset (see bit descriptions).

Figure 25-74. DCAN_RIOC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED					PU	PD	OD
R-0h					R/W-0h	R/W-0h	0h

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED				FUNC	DIR	OUT	IN
R-0h				0h	0h	0h	R-0h

Table 25-69. DCAN_RIOC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	
18	PU	R/W	0h	CAN_RX pull up/pull down select. This bit is only active when CAN_RX is configured to be an input. 0h (R/W) = CAN_RX pull down is selected, when pull logic is active (PD = 0). 1h (R/W) = CAN_T=RX pull up is selected, when pull logic is active(PD = 0).
17	PD	R/W	0h	CAN_RX pull disable. This bit is only active when CAN_TX is configured to be an input. 0h (R/W) = CAN_RX pull is active 1h (R/W) = CAN_RX pull is disabled
16	OD		0h	CAN_RX open drain enable. This bit is only active when CAN_RX is configured to be in GIO mode (TIOC.Func=0). Forced to '0' if Init bit of CAN control register is reset. 0h (R/W) = The CAN_RX pin is configured in push/pull mode. 1h (R/W) = The CAN_RX pin is configured in open drain mode.
15-4	RESERVED	R	0h	
3	FUNC		0h	CAN_RX function. This bit changes the function of the CAN_RX pin. Forced to '1' if Init bit of CAN control register is reset. 0h (R/W) = CAN_RX pin is in GIO mode. 1h (R/W) = CAN_RX pin is in functional mode (as an output to transmit CAN data).
2	DIR		0h	CAN_RX data direction. This bit controls the direction of the CAN_RX pin when it is configured to be in GIO mode only (TIOC.Func=0). Forced to '1' if Init bit of CAN control register is reset. 0h (R/W) = The CAN_RX pin is an input. 1h (R/W) = The CAN_RX pin is an output

Table 25-69. DCAN_RIOC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	OUT		0h	CAN_RX data out write. This bit is only active when CAN_RX pin is configured to be in GIO mode (TIOC.Func = 0) and configured to be an output pin (TIOC.Dir = 1). The value of this bit indicates the value to be output to the CAN_RX pin. Forced to Tx output of the CAN core, if Init bit of CAN control register is reset. 0h (R/W) = The CAN_RX pin is driven to logic low 1h (R/W) = The CAN_RX pin is driven to logic high
0	IN	R	0h	CAN_RX data in. Note: When CAN_RX pin is connected to a CAN transceiver, an external pullup resistor has to be used to ensure that the CAN bus will not be disturbed (for example, while reset of the DCAN module). 0h (R/W) = The CAN_RX pin is at logic low 1h (R/W) = The CAN_RX pin is at logic high

Multichannel Serial Port Interface (McSPI)

This chapter describes the McSPI of the device.

Topic **Page**

26.1 Introduction

This document is intended to provide programmers with a functional presentation of the Master/Slave Multichannel Serial Port Interface (McSPI) module. It also provides a register description and a module configuration example.

McSPI is a general-purpose receive/transmit master/slave controller that can interface with up to four slave external devices or one single external master. It allows a duplex, synchronous, serial communication between a CPU and SPI compliant external devices (Slaves and Masters).

26.1.1 McSPI Features

The general features of the SPI controller are:

- Buffered receive/transmit data register per channel (1 word deep)
- Multiple SPI word access with one channel using a FIFO
- Two DMA requests per channel, one interrupt line
- Single interrupt line, for multiple interrupt source events
- Serial link interface supports:
  - Full duplex / Half duplex
  - Multi-channel master or single channel slave operations
  - Programmable 1-32 bit transmit/receive shift operations.
  - Wide selection of SPI word lengths continuous from 4 to 32 bits
- Up to four SPI channels
- SPI word Transmit / Receive slot assignment based on round robin arbitration
- SPI configuration per channel (clock definition, enable polarity and word width)
- Clock generation supports:
  - Programmable master clock generation (operating from fixed 48-MHz functional clock input)
  - Selectable clock phase and clock polarity per chip select.

26.1.2 Unsupported McSPI Features

This device supports only two chip selects per module. Module wakeup during slave mode operation is not supported, as noted in *McSPI Clock and Reset Management*.

Table 26-1. Unsupported McSPI Features

Feature	Reason
Slave mode wakeup	SWAKEUP not connected
Retention during power down	Module not synthesized with retention enabled
McSPI3-4	
Chip selects 2-3	Not pinned out

26.2 Integration

This device includes five instantiations of McSPI: SPI0–SPI4. The McSPI module is a general-purpose receive/transmit master/slave controller that can interface with either up to four slave external devices or one single external master. Figure 26-1 shows the example of a system with multiple external slave SPI compatible devices and Figure 26-2 shows the example of a system with an external master.

Figure 26-1. SPI Master Application

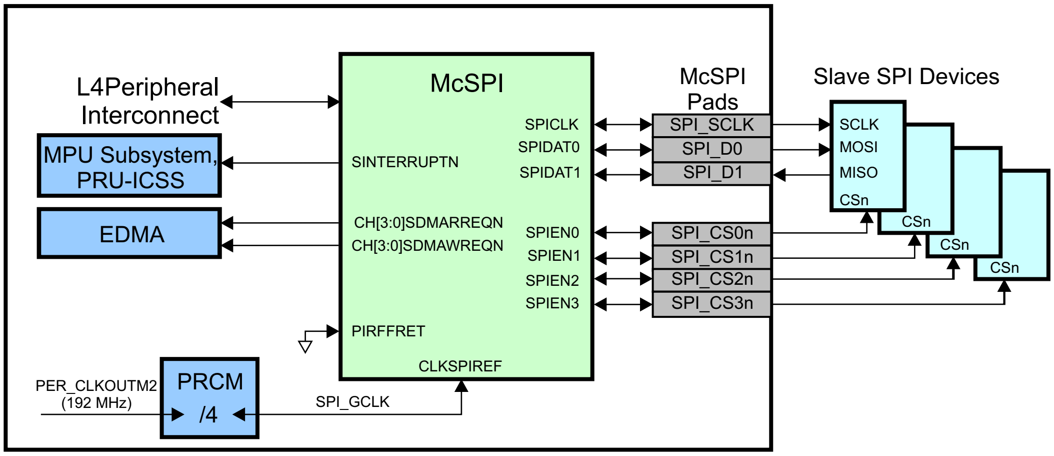
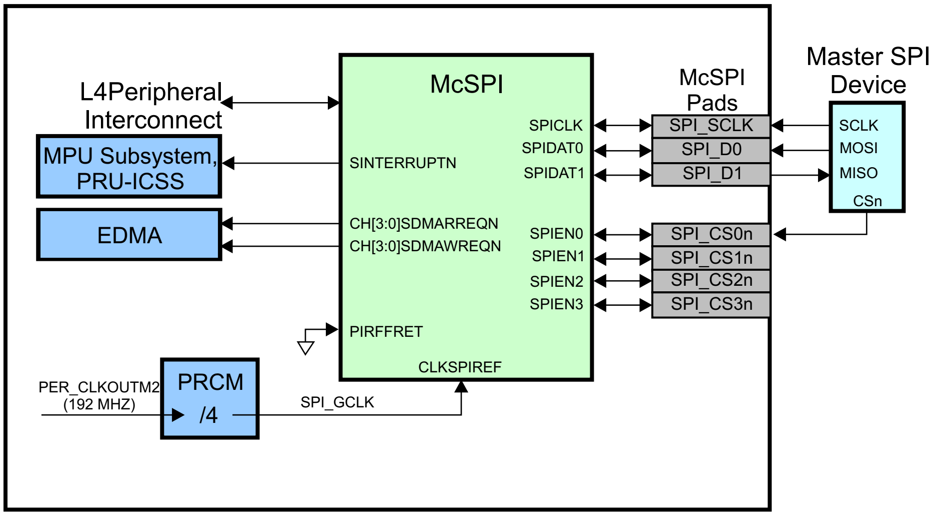


Figure 26-2. SPI Slave Application



26.2.1 McSPI Connectivity Attributes

The general connectivity attributes for the McSPI module are shown in Table 26-2.

Table 26-2. McSPI Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L4LS_GCLK (Interface/OCP) PD_PER_SPI_GCLK (Func)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	1 interrupt to MPU subsystem and PRU-ICSS (McSPI0INT) 1 interrupt per instance to MPU subsystem only (McSPI[1-4]INT)
DMA Requests	8 DMA requests per McSPI[0-2] instance, 4 DMA requests per McSPI[3-4] instance to EDMA • 1 RX request for CS0 (SPIREVT0) • 1 TX request for CS0 (SPIXEVT0) • 1 RX request for CS1 (SPIREVT1) • 1 TX request for CS1 (SPIXEVT1) • 1 RX request for CS2 (SPIREVT2), McSPI[0-2] only • 1 TX request for CS2 (SPIXEVT2), McSPI[0-2] only • 1 RX request for CS3 (SPIREVT3), McSPI[0-2] only • 1 TX request for CS3 (SPIXEVT3), McSPI[0-2] only
Physical Address	L4 Peripheral slave port

26.2.2 McSPI Clock and Reset Management

The SPI module clocks can be woken up in two manners: by the SPI module itself using the SWAKEUP signal, or directly from an external SPI master device by detecting an active low level on its chip select input pin (CS0n) using a GPIO attached to that device pin. Neither of these methods is supported on the device.

Table 26-3. McSPI Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
CLK Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk From PRCM
CLKSPIREF Functional clock	48 MHz	PER_CLKOUTM2 / 4	pd_per_spi_gclk From PRCM

26.2.3 McSPI Pin List

The McSPI interface pins are summarized in Table 26-4.

Table 26-4. McSPI Pin List

Pin	Type	Description
SPI*x*_SCLK	I/O(1)	SPI serial clock (output when master, input when slave)
SPI*x*_D0	I/O	Can be configured as either input or output (MOSI or MISO)
SPI*x*_D1	I/O	Can be configured as either input or output (MOSI or MISO)

(1) These signals are also used as inputs to re-time or sync data. The associated CONF_<module>_pin_RXACTIVE bit for these signals must be set to 1 to enable the inputs back to the module. It is also recommended to place a 33-ohm resistor in series (close to the processor) on each of these signals to avoid signal reflections.

Table 26-4. McSPI Pin List (continued)

Pin	Type	Description
SPIx_CS0	I/O	SPI chip select 0 output when master, input when slave (active low)
SPIx_CS1	I/O	SPI chip select 1 output when master, input when slave (active low)
SPIx_CS2	I/O	SPI chip select 2 output when master, input when slave (active low)
SPIx_CS3	I/O	SPI chip select 3 output when master, input when slave (active low)

26.3 Functional Description

26.3.1 SPI Transmission

This section describes the transmissions supported by McSPI. The SPI protocol is a synchronous protocol that allows a master device to initiate serial communication with a slave device. Data is exchanged between these devices. A slave select line (SPIEN) can be used to allow selection of an individual slave SPI device. Slave devices that are not selected do not interfere with SPI bus activities. Connected to multiple external devices, McSPI exchanges data with a single SPI device at a time through two main modes:

- Two data pins interface mode. (See Section 26.3.1.1)
- Single data pin interface mode (recommended for half-duplex transmission). (See Section 26.3.1.2)

The flexibility of McSPI allows exchanging data with several formats through programmable parameters described in Section 26.3.1.3.

26.3.1.1 Two Data Pins Interface Mode

The two data pins interface mode, allows a full duplex SPI transmission where data is transmitted (shifted out serially) and received (shifted in serially) simultaneously on separate data lines SPIDAT [0] and SPIDAT [1]. Data leaving the master exits on transmit serial data line also known as MOSI: MasterOutSlaveIn. Data leaving the slave exits on the receive data line also known as MISO: MasterInSlaveOut.

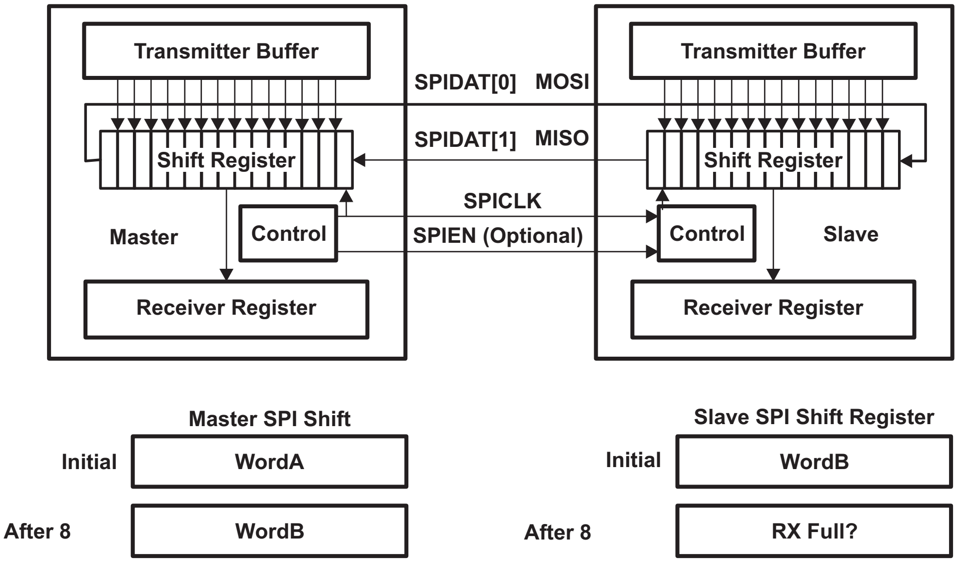
McSPI has a unified SPI port control: SPIDAT [1:0] can be independently configured as receive or transmit lines. The user has the responsibility to program which data line to use and in which direction (receive or transmit), according to the external slave/master connection.

The serial clock (SPICLK) synchronizes shifting and sampling of the information on the two serial data lines (SPIDAT [1:0]). Each time a bit is transferred out from the Master, one bit is transferred in from Slave.

Figure 26-3 shows an example of a full duplex system with a Master device on the left and a Slave device on the right. After 8 cycles of the serial clock SPICLK, the WordA has been transferred from the master to the slave. At the same time, the 8-bit WordB has been transferred from the slave to the master.

When referring to the master device, the control block transmits the clock SPICLK and the enable signal SPIEN (optional, see McSPI_MODULCTRL).

Figure 26-3. SPI Full-Duplex Transmission



26.3.1.2 Single Data Pin Interface Mode

In single data pin interface mode, under software control, a single data line is used to alternatively transmit and receive data (Half duplex transmission).

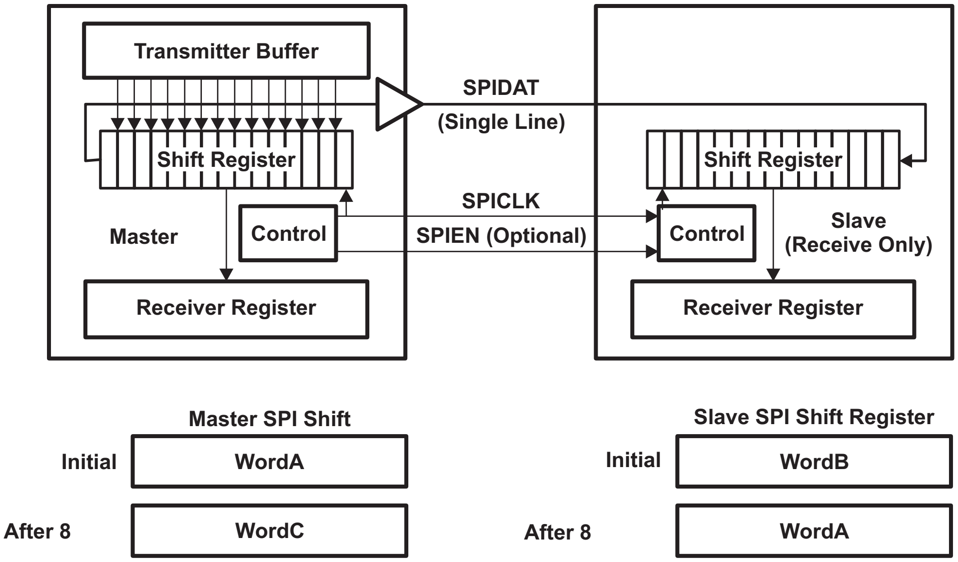
McSPI has a unified SPI port control: SPIDAT [1:0] can be independently configured as receive or transmit lines. The user has the responsibility to program which data line to use and in which direction (receive or transmit), according to the external slave/master connection.

As for a full duplex transmission, the serial clock (SPICLK) synchronizes shifting and sampling of the information on the single serial data line.

26.3.1.2.1 Example With a Receive-Only Slave

Figure 26-4 shows a half duplex system with a Master device on the left and a receive-only Slave device on the right. Each time a bit is transferred out from the Master, one bit is transferred in the Slave. After 8 cycles of the serial clock SPICLK, the 8-bit WordA has been transferred from the master to the slave.

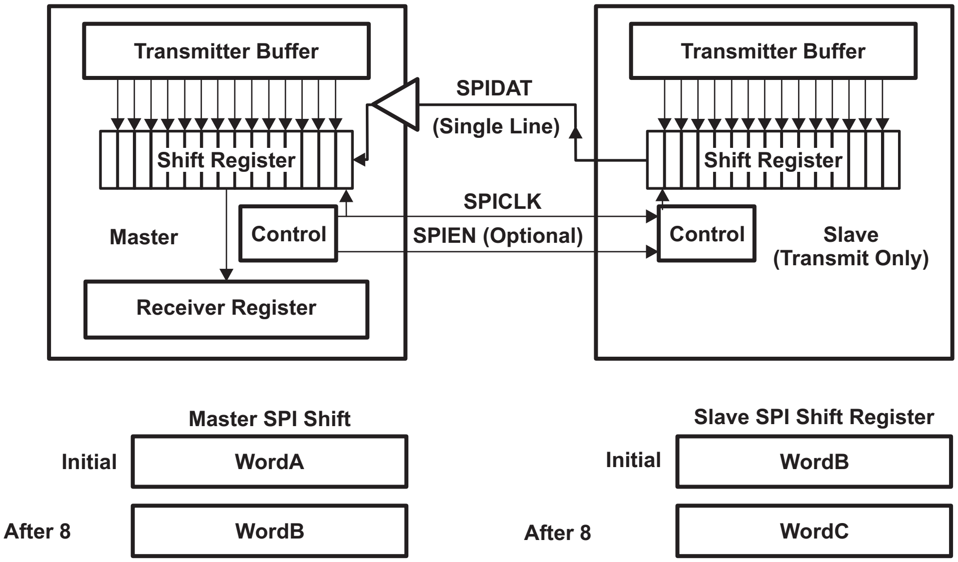
Figure 26-4. SPI Half-Duplex Transmission (Receive-only Slave)



26.3.1.2.2 Example With a Transmit-Only Slave

Figure 26-5 shows a half duplex system with a Master device on the left and a transmit-only Slave device on the right. Each time a bit is transferred out from the Slave, one bit is transferred in the Master. After 8 cycles of the serial clock SPICLK, the 8-bit WordA has been transferred from the slave to the master.

Figure 26-5. SPI Half-Duplex Transmission (Transmit-Only Slave)



26.3.1.3 Transfer Formats

This section describes the transfer formats supported by McSPI.

The flexibility of McSPI allows setting the parameters of the SPI transfer:

- SPI word length
- SPI enable generation programmable
- SPI enable assertion
- SPI enable polarity
- SPI clock frequency
- SPI clock phase
- SPI clock polarity

The consistency between SPI word length, clock phase and clock polarity of the master SPI device and the communicating slave device remains under software responsibility.

26.3.1.3.1 Programmable Word Length

McSPI supports any SPI word from 4 to 32 bits long.

The SPI word length can be changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

26.3.1.3.2 Programmable SPI Enable Generation

McSPI is able to generate or not generate SPI enable. If management of chip select is de-asserted, a point-to-point connection is mandatory. Only a single master of a slave device can be connected to the SPI bus.

26.3.1.3.3 Programmable SPI Enable (SPIEN)

The polarity of the SPIEN signals is programmable. SPIEN signals can be active high or low.

The assertion of the SPIEN signals is programmable. SPIEN signals can be manually asserted or automatically asserted.

Two consecutive words for two different slave devices may go along with active SPIEN signals with different polarity.

26.3.1.3.4 Programmable SPI Clock (SPICLK)

The phase and the polarity of the SPI serial clock are programmable when McSPI is a SPI master device or a SPI slave device. The baud rate of the SPI serial clock is programmable when McSPI is a SPI master.

When McSPI is operating as a slave, the serial clock SPICLK is an input from the master.

26.3.1.3.5 Bit Rate

In Master Mode, an internal reference clock CLKSPIREF is used as an input of a programmable divider to generate bit rate of the serial clock SPICLK. Granularity of this clock divider can be changed.

26.3.1.3.6 Polarity and Phase

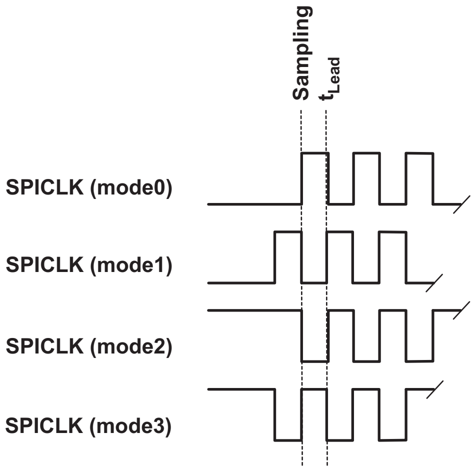
McSPI supports four sub-modes of the SPI format transfer that depend on the polarity (POL) and the phase (PHA) of the SPI serial clock (SPICLK). Table 26-5 and Figure 26-6 show a summary of the four sub-modes. Software selects one of four combinations of serial clock phase and polarity.

Two consecutive SPI words for two different slave devices may go along with active SPICLK signal with different phase and polarity.

Table 26-5. Phase and Polarity Combinations

Polarity (POL)	Phase (PHA)	SPI Mode	Comments
0	0	mode0	SPICLK active high and sampling occurs on the rising edge.
0	1	mode1	SPICLK active high and sampling occurs on the falling edge.
1	0	mode2	SPICLK active low and sampling occurs on the falling edge.
1	1	mode3	SPICLK active low and sampling occurs on the rising edge.

Figure 26-6. Phase and Polarity Combinations



26.3.1.3.7 Transfer Format With PHA = 0

This section describes the concept of a SPI transmission with the SPI mode0 and the SPI mode2.

In the transfer format with PHA = 0, SPIEN is activated a half cycle of SPICLK ahead of the first SPICLK edge.

In both master and slave modes, McSPI drives the data lines at the time of SPIEN is asserted.

Each data frame is transmitted starting with the MSB. At the extremity of both SPI data lines, the first bit of SPI word is valid a half-cycle of SPICLK after the SPIEN assertion.

Therefore, the first edge of the SPICLK line is used by the master to sample the first data bit sent by the slave. On the same edge, the first data bit sent by the master is sampled by the slave.

On the next SPICLK edge, the received data bit is shifted into the shift register, and a new data bit is transmitted on the serial data line.

This process continues for a total of pulses on the SPICLK line defined by the SPI word length programmed in the master device, with data being latched on odd numbered edges and shifted on even numbered edges.

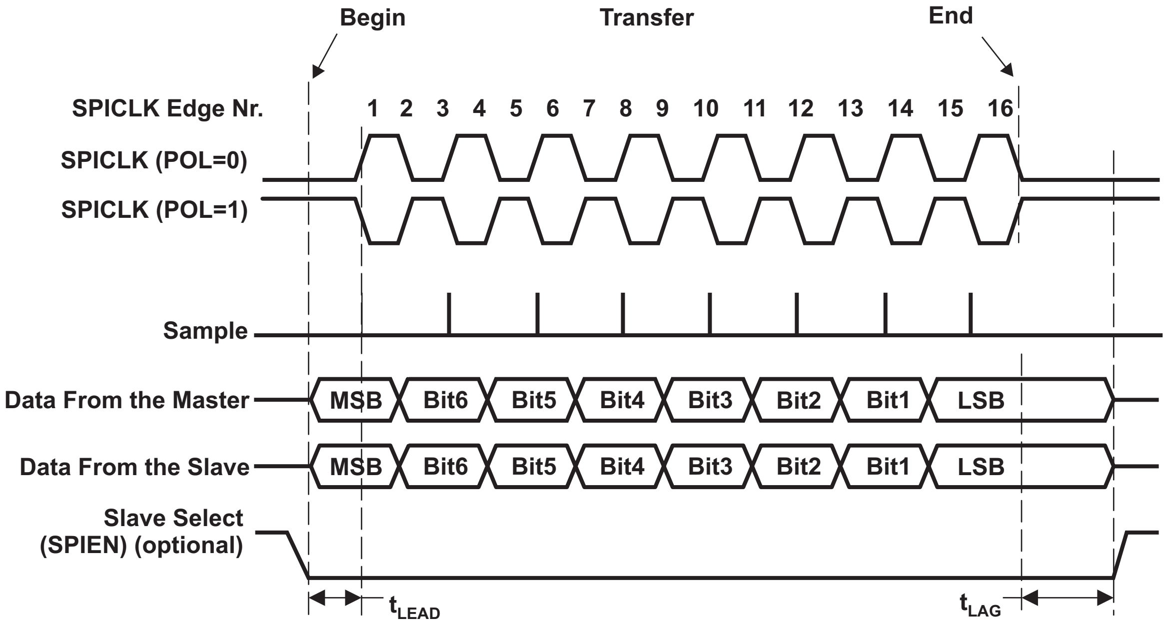
Figure 26-7 is a timing diagram of a SPI transfer for the SPI mode0 and the SPI mode2, when McSPI is master or slave, with the frequency of SPICLK equals to the frequency of CLKSPIREF. It should not be used as a replacement for SPI timing information and requirements detailed in the data manual.

When McSPI is in slave mode, if the SPIEN line is not de-asserted between successive transmissions then the content of the Transmitter register is not transmitted, instead the last received SPI word is transmitted.

In master mode, the SPIEN line must be negated and reasserted between each successive SPI word. This is because the slave select pin freezes the data in its shift register and does not allow it to be altered if PHA bit equals 0.

In 3-pin mode without using the SPIEN signal, the controller provides the same waveform but with SPIEN forced to low state. In slave mode, SPIEN is useless.

Figure 26-7. Full Duplex Single Transfer Format with PHA = 0



26.3.1.3.8 Transfer Format With PHA = 1

This section describes SPI full duplex transmission with the SPI mode1 and the SPI mode3.

In the transfer format with PHA = 1, SPIEN is activated a delay (t_{Lead}) ahead of the first SPICLK edge.

In both master and slave modes, McSPI drives the data lines on the first SPICLK edge.

Each data frame is transmitted starting with the MSB. At the extremity of both SPI data lines, the first bit of SPI word is valid on the next SPICLK edge, a half-cycle later of SPICLK. It is the sampling edge for both the master and slave.

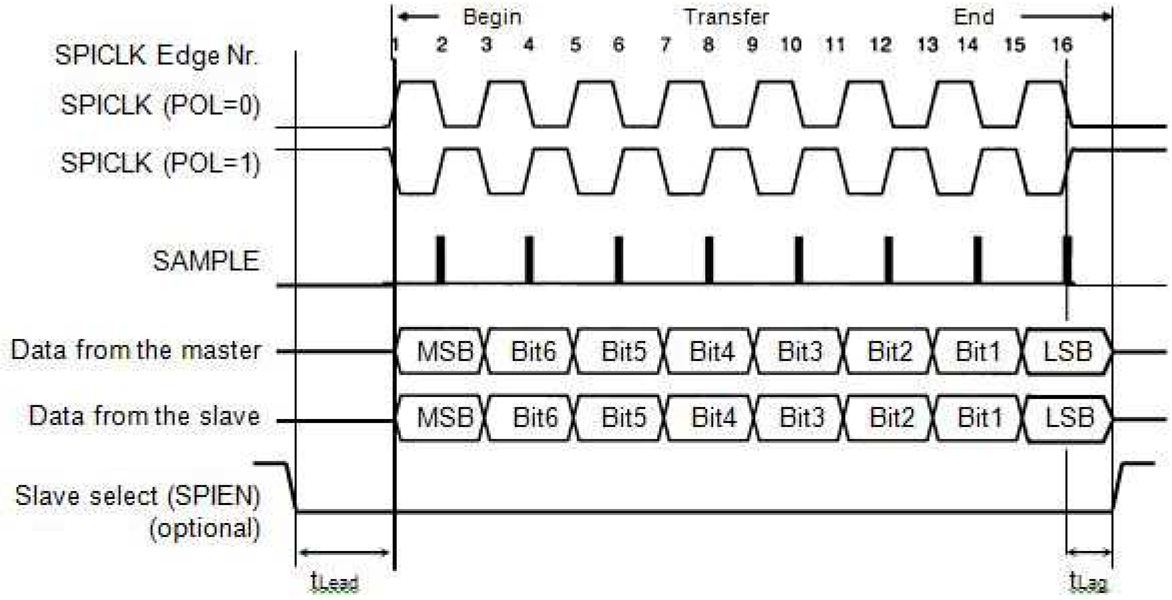
When the third edge occurs, the received data bit is shifted into the shift register. The next data bit of the master is provided to the serial input pin of the slave.

This process continues for a total of pulses on the SPICLK line defined by the word length programmed in the master device, with data being latched on even numbered edges and shifted on odd numbered edges.

Figure 26-8 is a timing diagram of a SPI transfer for the SPI mode1 and the SPI mode3, when McSPI is master or slave, with the frequency of SPICLK equals to the frequency of CLKSPIREF. It should not be used as a replacement for SPI timing information and requirements detailed in the data manual.

The SPIEN line may remain active between successive transfers. In 3-pin mode without using the SPIEN signal, the controller provides the same waveform but with SPIEN forced to low state. In slave mode SPIEN is useless.

Figure 26-8. Full Duplex Single Transfer Format With PHA = 1



26.3.2 Master Mode

McSPI is in master mode when the bit MS of the register MCSPI_MODULCTRL is cleared.

In master mode McSPI supports multi-channel communication with up to 4 independent SPI communication channel contexts. McSPI initiates a data transfer on the data lines (SPIDAT [1:0]) and generates clock (SPICLK) and control signals (SPIEN) to a single SPI slave device at a time.

26.3.2.1 Dedicated Resources Per Channel

In the following sections, the letter “i” indicates the channel number that can be 0, 1, 2 or 3. Each channel has the following dedicated resources:

- Its own channel enable, programmable with the bit EN of the register MCSPI_CH(i)CTRL. Disabling the channel, outside data word transmission, remains under user responsibility.
- Its own transmitter register MCSPI_TX on top of the common shift register. If the transmitter register is empty, the status bit TXS of the register MCSPI_CH(i)STAT is set.
- Its own receiver register MCSPI_RX on top of the common shift register. If the receiver register is full, the status bit RXS of the register MCSPI_CH(i)STAT is set.
- A fixed SPI ENABLE line allocation (SPIEN[i] port for channel "i"), SPI enable management is optional.
- Its own communication configuration with the following parameters via the register MCSPI_CH(i)CONF
  - Transmit/Receive modes, programmable with the bit TRM.
  - Interface mode (Two data pins or Single data pin) and data pins assignment, both programmable with the bits IS and DPE.
  - SPI word length, programmable with the bits WL.
  - SPIEN polarity, programmable with the bit EPOL.
  - SPIEN kept active between words, programmable with the bit FORCE.
  - Turbo mode, programmable with the bit TURBO.
  - SPICLK frequency, programmable with the bit CLKD, the granularity of clock division can be changed using CLKG bit, the clock ratio is then concatenated with MCSPI_CH(i)CTRL[EXTCLK] value.
  - SPICLK polarity, programmable with the bit POL.
  - SPICLK phase, programmable with the bit PHA.
  - Start bit polarity, programmable with the bit SBPOL
  - Use a FIFO Buffer or not (see the following note), programmable with FFER and FFEW, depending on transfer mode, (MCSPI_CH(i)CONF[TRM]).
- Two DMA requests events, read and write, to synchronize read/write accesses of the DMA controller with the activity of McSPI. The DMA requests are enabled with the bits DMAR and DMAW.
- Three interrupts events

Note: When more than one channel has an FIFO enable bit field (FFER or FFEW) set, the FIFO will not be used on any channel. Software must ensure that only one enabled channel is configured to use the FIFO buffer.

The transfers will use the latest loaded parameters of the register MCSPI_CH(i)CONF.

The configuration parameters SPIEN polarity, Turbo mode, SPICLK phase and SPICLK polarity can be loaded in the MCSPI_CH(i)CONF register only when the channel is disabled. The user has the responsibility to change the other parameters of the MCSPI_CH(i)CONF register when no transfer occurs on the SPI interface.

26.3.2.2 Interrupt Events in Master Mode

In master mode, the interrupt events related to the transmitter register state are TX_empty and TX_underflow. The interrupt event related to the receiver register state is RX_full.

26.3.2.2.1 TX_empty

The event TX_empty is activated when a channel is enabled and its transmitter register becomes empty (transient event). Enabling channel automatically raises this event, except for the Master receive only mode. (See Section 26.3.2.5). When the FIFO buffer is enabled (MCSPI_CH(i)CONF[FFEW] set to 1), the TX_empty is asserted as soon as there is enough space in the buffer to write a number of bytes defined by MCSPI_XFERLEVEL[AEL].

Transmitter register must be loaded to remove the source of the interrupt and the TX_empty interrupt status bit must be cleared for interrupt line de-assertion (if event enabled as interrupt source) . (See Section 26.3.4).

When FIFO is enabled, no new TX_empty event will be asserted as soon as CPU has not performed the number of writes into the transmitter register defined by MCSPI_XFERLEVEL[AEL]. It is the responsibility of CPU to perform the right number of writes.

26.3.2.2.2 TX_underflow

The event TX_underflow is activated when the channel is enabled and if the transmitter register or FIFO is empty (not updated with new data) at the time of shift register assignment.

The TX_underflow is a harmless warning in master mode.

To avoid having TX_underflow event at the beginning of a transmission, the event TX_underflow is not activated when no data has been loaded into the transmitter register since channel has been enabled.

To avoid having a TX_underflow event, the Transmit Register (MCSPI_TX(i)) should be loaded as infrequently as possible.

TX_underflow interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

Note: When more than one channel has an FIFO enable bit field (FFER or FFEW) set, the FIFO will not be used on any channel. Software must ensure that only one enabled channel is configured to use the FIFO buffer.

26.3.2.2.3 RX_ full

The event RX_full is activated when channel is enabled and receiver register becomes filled (transient event). When FIFO buffer is enabled (MCSPI_CH(i)CONF[FFER] set to 1), the RX_full is asserted when the number of bytes in the buffer equals the level defined by MCSPI_XFERLEVEL[AFL].

Receiver register must be read to remove source of interrupt and RX_full interrupt status bit must be cleared for interrupt line de-assertion (if event enabled as interrupt source).

When the FIFO is enabled, no new RX_FULL event will be asserted once the CPU has read the number of bytes defined by MCSPI_XFERLEVEL[AFL]. It is the responsibility of the CPU to perform the correct number of read operations.

26.3.2.2.4 End of Word Count

The event end of word (EOW) count is activated when channel is enabled and configured to use the built-in FIFO. This interrupt is raised when the controller had performed the number of transfers defined in the MCSPI_XFERLEVEL[WCNT] register. If the value was programmed to 0000h, the counter is not enabled and this interrupt is not generated.

The EOW count interrupt also indicates that the SPI transfer has halted on the channel using the FIFO buffer.

The EOW interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

26.3.2.3 Master Transmit and Receive Mode

This mode is programmable per channel (bit TRM of register MCSPI_CH(i)CONF).

The channel access to the shift registers, for transmission/reception, is based on its transmitter and receiver register state and round robin arbitration.

The channel that meets the rules below is included in the round robin list of already active channels scheduled for transmission and/or reception. The arbiter skips the channel that does not meet the rules and search for the next following enabled channel, in rotation.

Rule 1: Only enabled channels (bit EN of the register MCSPI_CH(i)CTRL), can be scheduled for transmission and/or reception.

Rule 2: An enabled channel can be scheduled if its transmitter register is not empty (bit TXS of the register MCSPI_CH(i)STAT) or its FIFO is not empty when the buffer is used for the corresponding channel (bit FFE of the register MCSPI_CH(i)STAT) at the time of shift register assignment. If the transmitter register or FIFO is empty, at the time of shift register assignment, the event TX_underflow is activated and the next enabled channel with new data to transmit is scheduled. (See also transmit only mode).

Rule 3: An enabled channel can be scheduled if its receive register is not full (bit RXS of the register MCSPI_CH(i)STAT)) or its FIFO is not full when the buffer is used for the corresponding channel (bit FFF of the register MCSPI_CH(i)STAT) at the time of shift register assignment. (See also receive only mode). Therefore the receiver register of FIFO cannot be overwritten. The RX_overflow bit, in the MCSPI_IRQSTS register is never set in this mode.

On completion of SPI word transfer (bit EOT of the register MCSPI_CH(i)STAT is set) the updated transmitter register for the next scheduled channel is loaded into the shift register. This bit is meaningless when using the Buffer for this channel. The serialization (transmit and receive) starts according to the channel communication configuration. On serialization completion the received data is transferred to the channel receive register.

The built-in FIFO is available in this mode and if configured in one data direction, transmit or receive, then the FIFO is seen as a unique 64-byte buffer. If configured in both data directions, transmit and receive, then the FIFO is split into two separate 32-byte buffers with their own address space management. In this last case, the definition of AEL and AFL levels is based on 32 bytes and is under CPU responsibility.

26.3.2.4 Master Transmit-Only Mode

This mode eliminates the need for the CPU to read the receiver register (minimizing data movement) when only transmission is meaningful.

The master transmit only mode is programmable per channel (bits TRM of the register MCSPI_CH(i)CONF).

In master transmit only mode, transmission starts after data is loaded into the transmitter register.

Rule 1 and **Rule 2**, defined above, are applicable in this mode.

Rule 3, defined above, is not applicable: In master transmit only mode, the receiver register or FIFO state “full” does not prevent transmission, and the receiver register is always overwritten with the new SPI word. This event in the receiver register is not significant when only transmission is meaningful. So, the RX_overflow bit, in the MCSPI_IRQSTS register is never set in this mode.

The McSPI module automatically disables the RX_full interrupt status. The corresponding interrupt request and DMA Read request are not generated in master transmit only mode.

The status of the serialization completion is given by the bit EOT of the register MCSPI_CH(i)STAT. This bit is meaningless when using the Buffer for this channel.

The built-in FIFO is available in this mode and can be configured with FFEW bit field in the MCSPI_CH(i)CONF register, then the FIFO is seen as a unique 64-byte buffer.

26.3.2.5 Master Receive-Only Mode

This mode eliminates the need for the CPU to refill the transmitter register (minimizing data movement) when only reception is meaningful.

The master receive mode is programmable per channel (bits TRM of the register MCSPI_CH(i)CONF).

The master receive only mode enables channel scheduling only on empty state of the receiver register.

Rule 1 and **Rule 3**, defined above, are applicable in this mode.

Rule 2, defined above, is not applicable: In master receive only mode, after the first loading of the transmitter register of the enabled channel, the transmitter register state is maintained as full. The content of the transmitter register is always loaded into the shift register, at the time of shift register assignment. So, after the first loading of the transmitter register, the bits TX_empty and TX_underflow, in the MCSPI_IRQSTS register are never set in this mode.

The status of the serialization completion is given by the bit EOT of the register MCSPI_CH(i)STAT. The bit RX_full in the MCSPI_IRQSTS register is set when a received data is loaded from the shift register to the receiver register. This bit is meaningless when using the Buffer for this channel.

The built-in FIFO is available in this mode and can be configured with FFER bit field in the MCSPI_CH(i)CONF register, then the FIFO is seen as a unique 64-byte buffer.

26.3.2.6 Single-Channel Master Mode

When the SPI is configured as a master device with a single enabled channel, the assertion of the SPIM_CSX signal can be controlled in two different ways:

- In 3 pin mode : MCSPI_MODULCTRL[1] PIN34 and MCSPI_MODULCTRL[0] SINGLE bit are set to 1, the controller transmit SPI word as soon as transmit register or FIFO is not empty.
- In 4 pin mode : MCSPI_MODULCTRL[1] PIN34 bit is cleared to 0 and MCSPI_MODULCTRL[0] SINGLE bit is set to 1, SPIEN assertion/deassertion controlled by Software. (See Section 26.3.2.6.1) using the MCSPI_CH(i)CONF[20] FORCE bit.

26.3.2.6.1 Programming Tips When Switching to Another Channel

When a single channel is enabled and data transfer is ongoing:

- Wait for completion of the SPI word transfer (bit EOT of the register MCSPI_CH(i)STAT is set) before disabling the current channel and enabling a different channel.
- Disable the current channel first, and then enable the other channel.

26.3.2.6.2 Keep SPIEN Active Mode (Force SPIEN)

Continuous transfers are manually allowed by keeping the SPIEN signal active for successive SPI words transfer. Several sequences (configuration/enable/disable of the channel) can be run without deactivating the SPIEN line. This mode is supported by all channels and any master sequence can be used (transmit-receive, transmit-only, receive-only).

Keeping the SPIEN active mode is supported when:

- A single channel is used (bit MCSPI_MODULCTRL[Single] is set to 1).
- Transfer parameters of the transfer are loaded in the configuration register (MCSPI_CH(i)CONF) in the appropriate channel.

  The state of the SPIEN signal is programmable.

  - Writing 1 into the bit FORCE of the register MCSPI_CH(i)CONF drives high the SPIEN line when MCSPI_CH(i)CONF[EPOL] is set to zero, and drives it low when MCSPI_CH(i)CONF[EPOL] is set.
  - Writing 0 into the bit FORCE of the register MCSPI_CH(i)CONF drives low the SPIEN line when MCSPI_CH(i)CONF[EPOL] is set to zero, and drives it high when MCSPI_CH(i)CONF[EPOL] is set.
- A single channel is enabled (MCSPI_CH(i)CTRL[En] set to 1) . The first enabled channel activates the SPIEN line.

Once the channel is enabled, the SPIEN signal is activated with the programmed polarity.

As in multi-channel master mode, the start of the transfer depends on the status of the transmitter register, the status of the receiver register and the mode defined by the bits TRM in the configuration register (transmit only, receive only or transmit and receive) of the enabled channel.

The status of the serialization completion of each SPI word is given by the bit EOT of the register MCSPI_CH(i)STAT. The bit RX_full in the MCSPI_IRQSTS register is set when a received data is loaded from the shift register to the receiver register.

A change in the configuration parameters is propagated directly on the SPI interface. If the SPIEN signal is activated the user must insure that the configuration is changed only between SPI words, in order to avoid corrupting the current transfer.

NOTE: The SPIEN polarity, the SPICLK phase and SPICLK polarity must not be modified when the SPIEN signal is activated. The Transmit/Receive mode, programmable with the bit TRM can be modified only when the channel is disabled. The channel can be disabled and enabled while the SPIEN signal is activated.

The delay between SPI words that requires the connected SPI slave device to switch from one configuration (transmit only for instance) to another (receive only for instance) must be handled under software responsibility.

At the end of the last SPI word, the channel must be deactivated (MCSPI_CH(i)CTRL[En] is cleared to 0) and the SPIEN can be forced to its inactive state (MCSPI_CH(i)CONF[Force]).

Figure 26-9 and Figure 26-10 show successive transfers with SPIEN kept active low with a different configuration for each SPI word in respectively single data pin interface mode and two data pins interface mode. The arrows indicate when the channel is disabled before a change in the configuration parameters and enabled again.

Figure 26-9. Continuous Transfers With SPIEN Maintained Active (Single-Data-Pin Interface Mode)

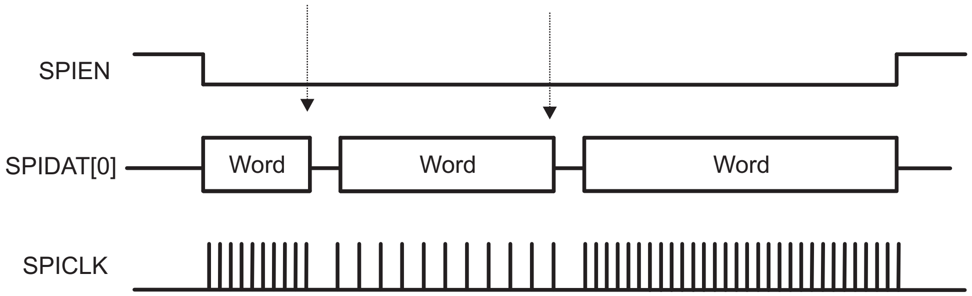
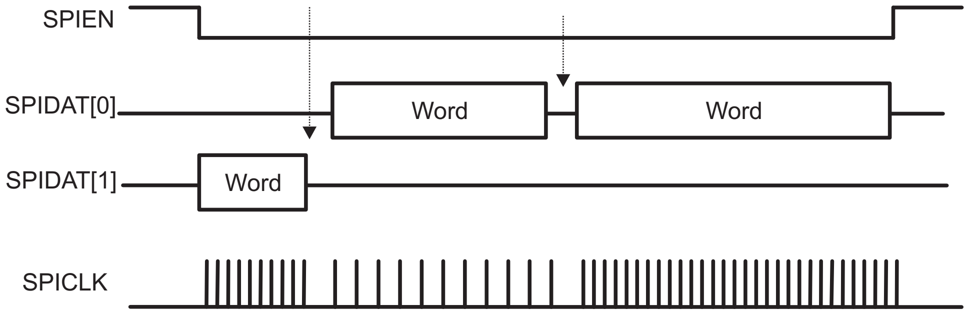


Figure 26-10. Continuous Transfers With SPIEN Maintained Active (Dual-Data-Pin Interface Mode)



NOTE: The turbo mode is also supported for the Keep SPIEN active mode when the following conditions are met:

- A single channel will be explicitly used (bit MCSPI_MODULCTRL[Single] is set to 1).
- The turbo mode is enabled in the configuration of the channel (bit Turbo of the register MCSPI_CH(i)CONF).

26.3.2.6.3 Turbo Mode

The purpose of the Turbo mode is to improve the throughput of the SPI interface when a single channel is enabled, by allowing transfers until the shift register and the receiver register are full.

This mode is programmable per channel (bit Turbo of the register MCSPI_CH(i)CONF). When several channels are enabled, the bit Turbo of the registers MCSPI_CH(i)CONF has no effect, and the channel access to the shift registers remains as described in Section 26.3.2.3.

In Turbo mode, **Rule 1** and **Rule 2** defined in Section 26.3.2.3 are applicable but Rule 3 is not applicable. An enabled channel can be scheduled if its receive register is full (bit RXS of the register MCSPI_CH(i)STAT) at the time of shift register assignment until the shift register is full.

In Turbo mode, **Rule 1** and **Rule 2** defined in Section 26.3.2.3 are applicable but Rule 3 is not applicable. An enabled channel can be scheduled if its receive register is full (bit RXS of the register MCSPI_CH(i)STAT) at the time of shift register assignment until the shift register is full.

The receiver register cannot be overwritten in Turbo mode. In consequence the RX_overflow bit, in MCSPI_IRQSTS register is never set in this mode.

26.3.2.7 Start Bit Mode

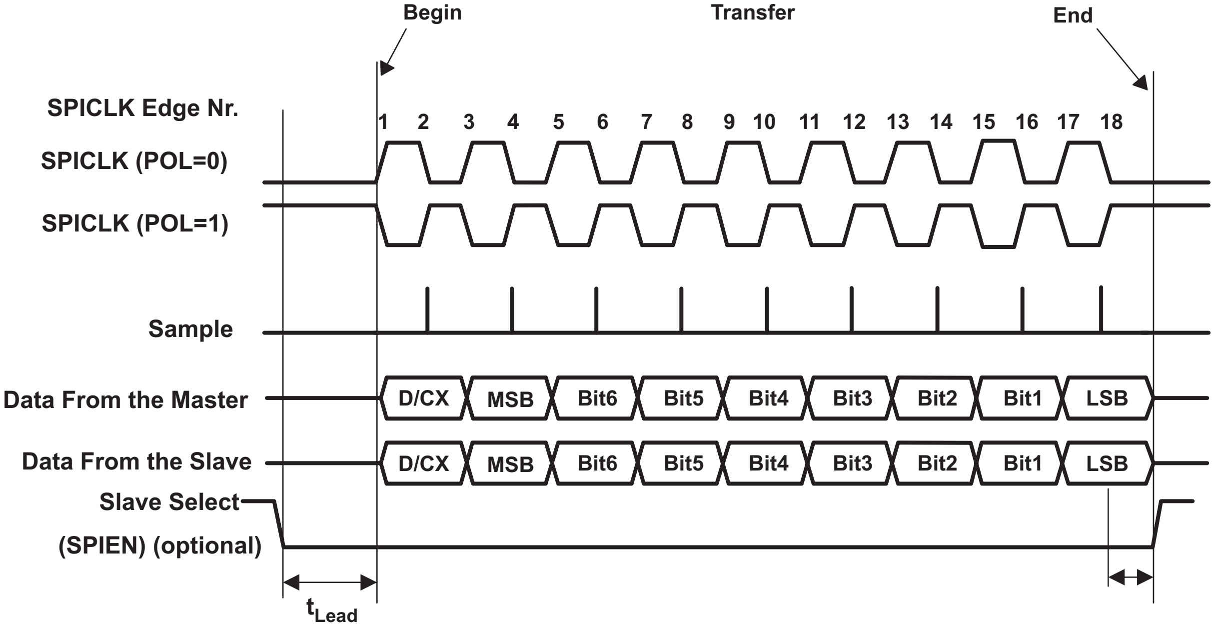
The purpose of the start bit mode is to add an extended bit before the SPI word transmission specified by word length WL. This feature is only available in master mode.

This mode is programmable per channel using the start bit enable (SBE) bit of the register MCSPI_CH(i)CONF).

The polarity of the extended bit is programmable per channel and it indicates whether the next SPI word must be handled as a command when SBPOL is cleared to 0 or as a data or a parameter when SBPOL is set to 1. Moreover start bit polarity SBPOL can be changed dynamically during start bit mode transfer without disabling the channel for reconfiguration, in this case you have the responsibility to configure the SBPOL bit before writing the SPI word to be transmitted in TX register.

The start bit mode could be used at the same time as turbo mode and/or manual chip select mode. In this case only one channel could be used, no round-robin arbitration is possible.

Figure 26-11. Extended SPI Transfer With Start Bit PHA = 1

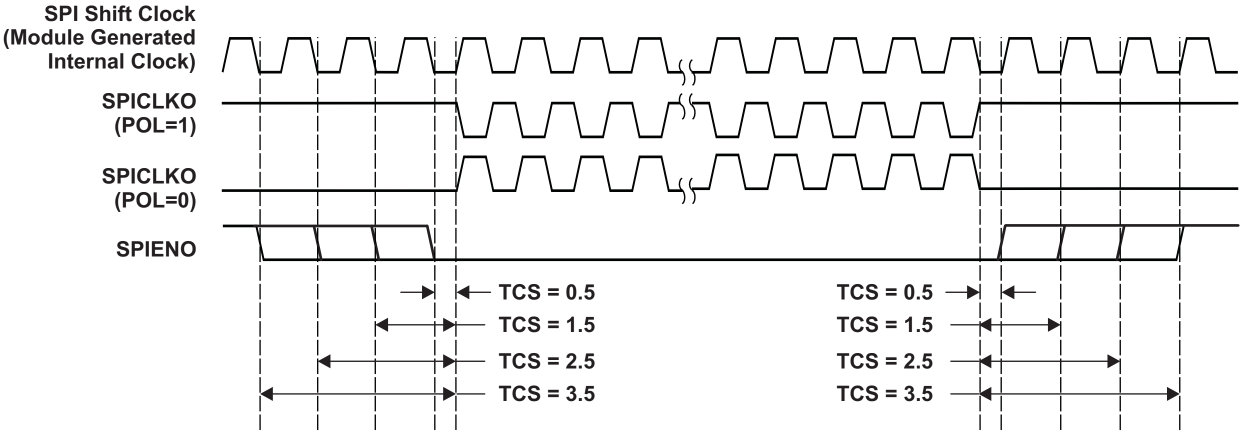


26.3.2.8 Chip-Select Timing Control

The chip select timing control is only available in master mode with automatic chip select generation (FORCE bit field is cleared to 0), to add a programmable delay between chip select assertion and first clock edge or chip select removal and last clock edge. The option is available only in 4 pin mode MCSPI_MODULCTRL[1] PIN34 is cleared to 0.

This mode is programmable per channel (bit TCS of the register MCSPI_CH(i)CONF). Figure 26-12 shows the chip-select SPIEN timing control.

Figure 26-12. Chip-Select SPIEN Timing Controls



NOTE: Because of the design implementation for transfers using a clock divider ratio set to 1 (clock bypassed), a half cycle must be added to the value between chip-select assertion and the first clock edge with PHA = 1 or between chip-select removal and the last clock edge with PHA = 0.

With an odd clock divider ratio which occurs when granularity is one clock cycle, that means that MCSPI_CH(i)CONF[CLKG] is set to 1 and MCSPI_CH(i)CONF[CLKD] has an even value, the clock duty cycle is not 50%, then one of the high level or low level duration is selected to be added to TCS delay.

Table 26-6 summarizes all delays between chip select and first (setup) or last (hold) clock edge.

In 3-pin mode this option is useless, the chip select SPIEN is forced to low state.

Table 26-6. Chip Select ↔ Clock Edge Delay Depending on Configuration

Clock Ratio F_{ratio}	Clock Phase PHA	Chip Select ↔ Clock Edge Delay	
		Setup	Hold
1	0	T_ref × (TCS + ½)	T_ref × (TCS + 1)
	1	T_ref × (TCS + 1)	T_ref × (TCS + ½)
Even ≥ 2	x	T_ref × F_{ratio} × (TCS + ½)	T_ref × F_{ratio} × (TCS + ½)
Odd ≥ 3 (only with MCSPI_CH(i)CONF[CLKG] set to 1	0	T_ref × [{F_{ratio} × TCS) + (F_{ratio} + ½)]	T_ref × [{F_{ratio} × TCS) + (F_{ratio} + ½)]
	1	T_ref × [{F_{ratio} × TCS) + (F_{ratio} - ½)]	T_ref × [{F_{ratio} × TCS) + (F_{ratio} - ½)]

T_ref = CLKSPIREF period in ns. F_{ratio} = SPI clock division ratio

The clock divider ratio depends on divider granularity MCSPI_CH(i)CONF[CLKG]:

- MCSPI_CH(i)CONF[CLKG] = 0 : granularity is power of two.
  $F_{ratio} = 2^{MCSPI\_CH(i)CONF[CLKD]}$
- MCSPI_CH(i)CONF[CLKG] = 1 : granularity is one cycle.
  F_{ratio} = MCSPI_CH(i)CTRL[EXTCLK] × MCSPI_CH(i)CONF[CLKD] + 1

26.3.2.9 Clock Ratio Granularity

By default the clock division ratio is defined by the register MCSPI_CH(i)CONF[CLKD] with power of two granularity leading to a clock division in range 1 to 32768, in this case the duty cycle is always 50%. With bit MCSPI_CH(i)CONF[CLKG] the clock division granularity can be changed to one clock cycle, in that case the register MCSPI_CH(i)CTRL[EXTCLK] is concatenated with MCSPI_CH(i)CONF[CLKD] to give a 12-bit width division ratio in range 1 to 4096.

When granularity is one clock cycle (MCSPI_CH(i)CONF[CLKG] set to 1), for odd value of clock ratio the clock duty cycle is kept to 50-50 using falling edge of clock reference CLKSPIREF.

Table 26-7. CLKSPIO High/Low Time Computation

Clock Ratio F_{ratio}	CLKSPIO High Time	CLKSPIO Low Time
1	T_{high}_ref	T_{low}_ref
Even ≥ 2	t_ref × (F_{ratio}/2)	t_ref × (F_{ratio}/2)
Odd ≥ 3	t_ref × (F_{ratio}/2)	t_ref × (F_{ratio}/2)

T_ref = CLKSPIREF period in ns. T_{high}_ref = CLKSPIREF high Time period in ns. T_{low}_ref = CLKSPIREF low Time period in ns. F_{ratio} = SPI clock division ratio

F_{ratio} = MCSPI_CH(i)CTRL[EXTCLK] × MCSPI_CH(i)CONF[CLKD] + 1

For odd ratio value the duty cycle is calculated as below:

Duty_cycle = ½

Granularity examples: With a clock source frequency of 48 MHz:

Table 26-8. Clock Granularity Examples

MCSPI_CH (i)CTRL	MCSPI_CH (i)CONF	MCSPI_CH (i)CONF		MCSPI_CH (i)CONF	MCSPI_CH (i)CONF					
EXTCLK	CLKD	CLKG	F_{ratio}	PHA	POL	Thigh (ns)	Tlow (ns)	Tperiod (ns)	Duty Cycle	Fout (MHz)
X	0	0	1	X	X	10.4	10.4	20.8	50-50	48
X	1	0	2	X	X	20.8	20.8	41.6	50-50	24
X	2	0	4	X	X	41.6	41.6	83.2	50-50	12
X	3	0	8	X	X	83.2	83.2	166.4	50-50	6
0	0	1	1	X	X	10.4	10.4	20.8	50-50	48
0	1	1	2	X	X	20.8	20.8	41.6	50-50	24
0	2	1	3	1	0	31,2	31,2	62.4	50-50	16
0	2	1	3	1	1	31,2	31,2	62.4	50-50	16
0	3	1	4	X	X	41.6	41.6	83.2	50-50	12
5	0	1	81	1	0	842,4	842,4	1684.8	50-50	0.592
5	7	1	88	X	X	915.2	915.2	1830.4	50-50	0.545

26.3.2.10 FIFO Buffer Management

The McSPI controller has a built-in 64-byte buffer in order to unload DMA or interrupt handler and improve data throughput.

This buffer can be used by only one channel and is selected by setting MCSPI_CH(i)CONF[FFER] and/or MCSPI_CH(i)CONF[FFEW] to 1.

If several channels are selected and several FIFO enable bit fields set to 1, the controller forces the buffer to be disabled for all channels. It is the responsibility of the driver to enable the buffer for only one channel.

The buffer can be used in the modes defined below:

- Master or Slave mode.
- Transmit only, Receive only or Transmit/Receive mode.
- Single channel or turbo mode, or in normal round robin mode. In round robin mode the buffer is used by only one channel.
- All word length MCSPI_CH(i)CONF[WL] are supported.

Two levels AEL and AFL located in MCSPI_XFERLEVEL register rule the buffer management. The granularity of these levels is one byte, then it is not aligned with SPI word length. It is the responsibility of the driver to set these values as a multiple of SPI word length defined in MCSPI_CH(i)CONF[WL]. The number of byte written in the FIFO depends on word length (see Table 26-9).

Table 26-9. FIFO Writes, Word Length Relationship

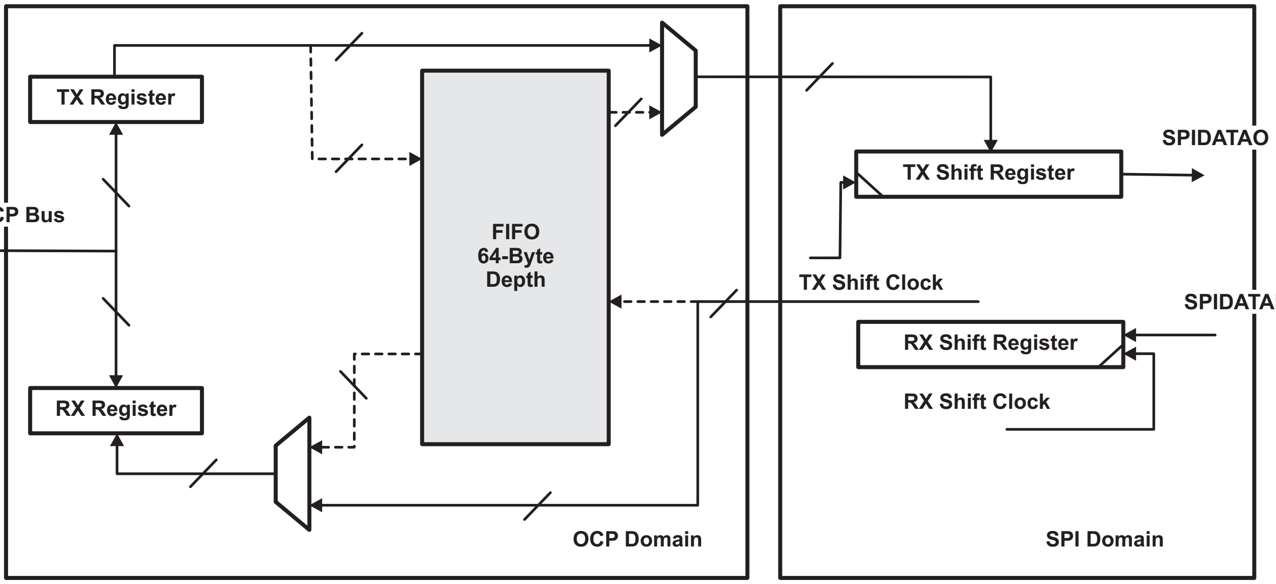
	SPI Word Length WL		
	3 ≤ WL ≤ 7	**8 ≤ WL ≤ 15**	**16 ≤ WL ≤ 31**
Number of byte written in the FIFO	1 byte	2 bytes	4 bytes

26.3.2.10.1 Split FIFO

The FIFO can be split into two part when module is configured in transmit/receive mode MCSPI_CH(i)CONF[TRM] is cleared to 0 and MCSPI_CH(i)CONF[FFER] and MCSPI_CH(i)CONF[FFEW] asserted. Then system can access a 32-byte depth FIFO per direction.

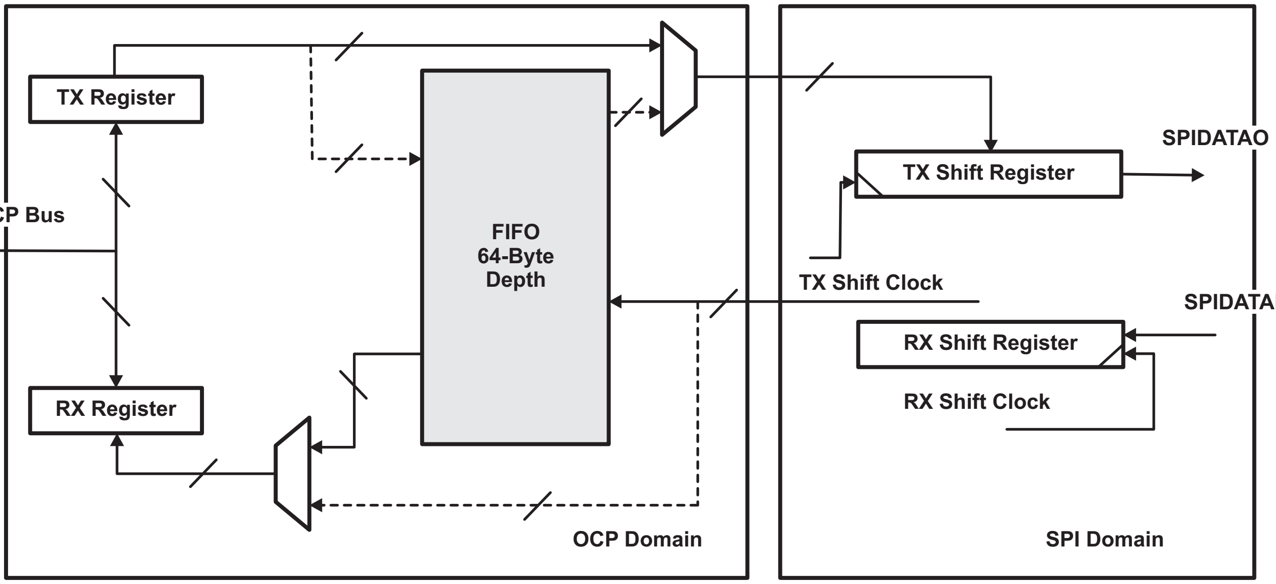
The FIFO buffer pointers are reset when the corresponding channel is enabled or FIFO configuration changes.

Figure 26-13. Transmit/Receive Mode With No FIFO Used



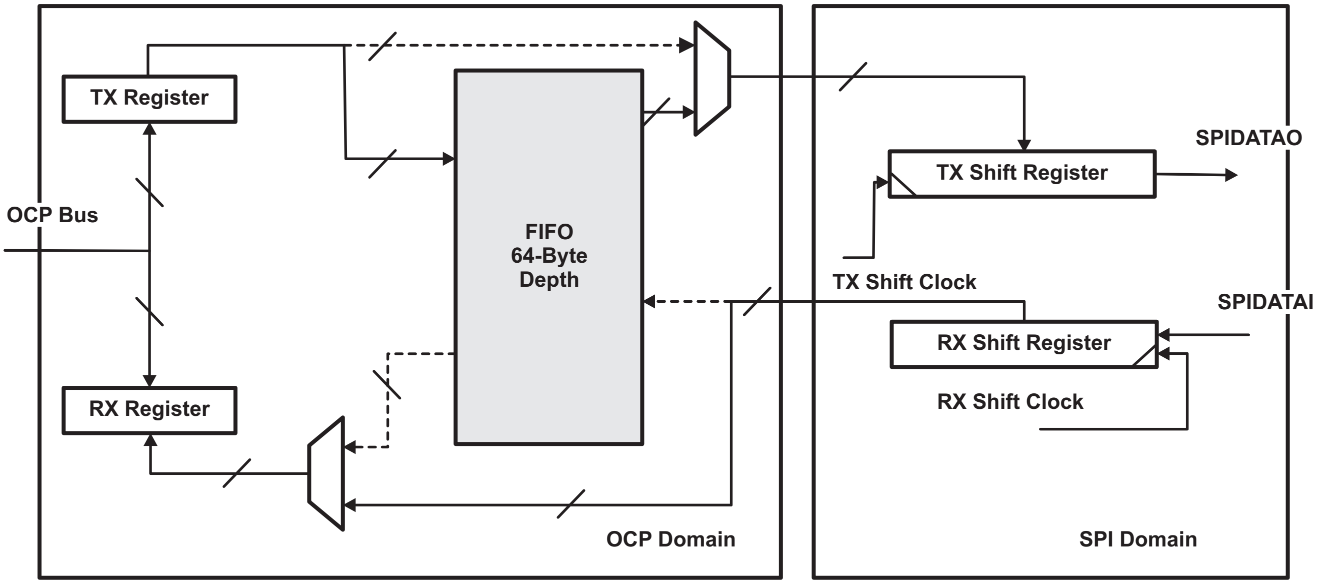
Configuration:
MCSPI_CH(i)CONF[TRM]=0x0 Transmit/receive mode
MCSPI_CH(i)CONF[FFRE]=0x0 FIFO disabled on receive path
MCSPI_CH(i)CONF[FFWE]=0x0 FIFO disabled on transmit path

Figure 26-14. Transmit/Receive Mode With Only Receive FIFO Enabled



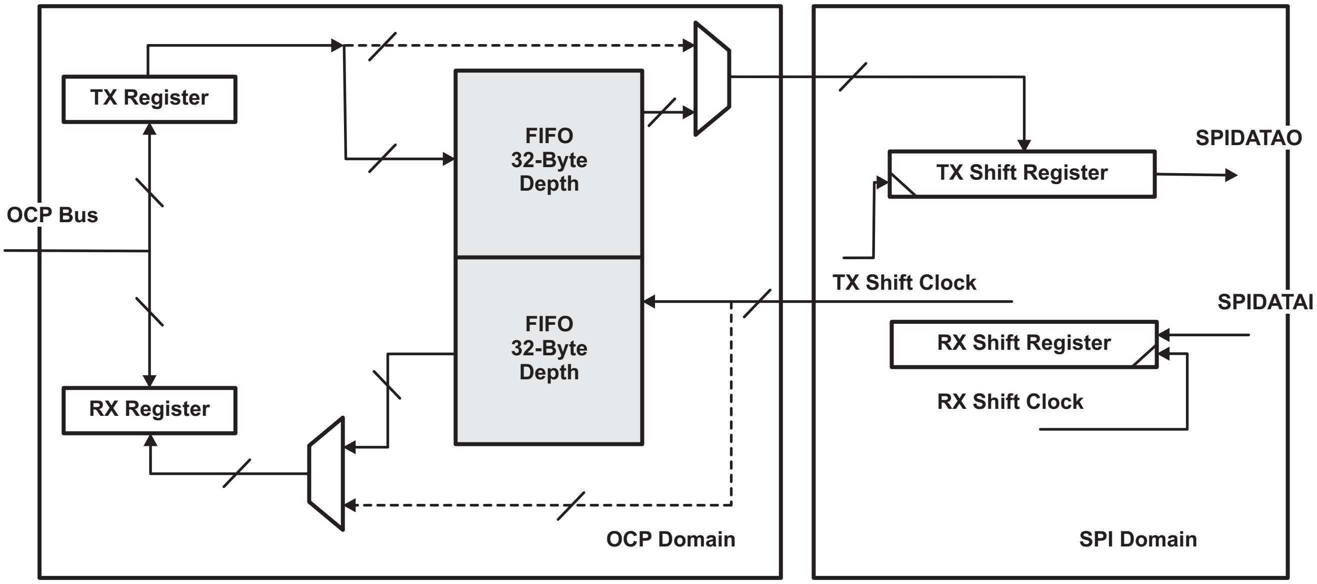
Configuration:
MCSPI_CH(i)CONF[TRM]=0x0 Transmit/receive mode
MCSPI_CH(i)CONF[FFRE]=0x1 FIFO enabled on receive path
MCSPI_CH(i)CONF[FFWE]=0x0 FIFO disabled on transmit path

Figure 26-15. Transmit/Receive Mode With Only Transmit FIFO Used



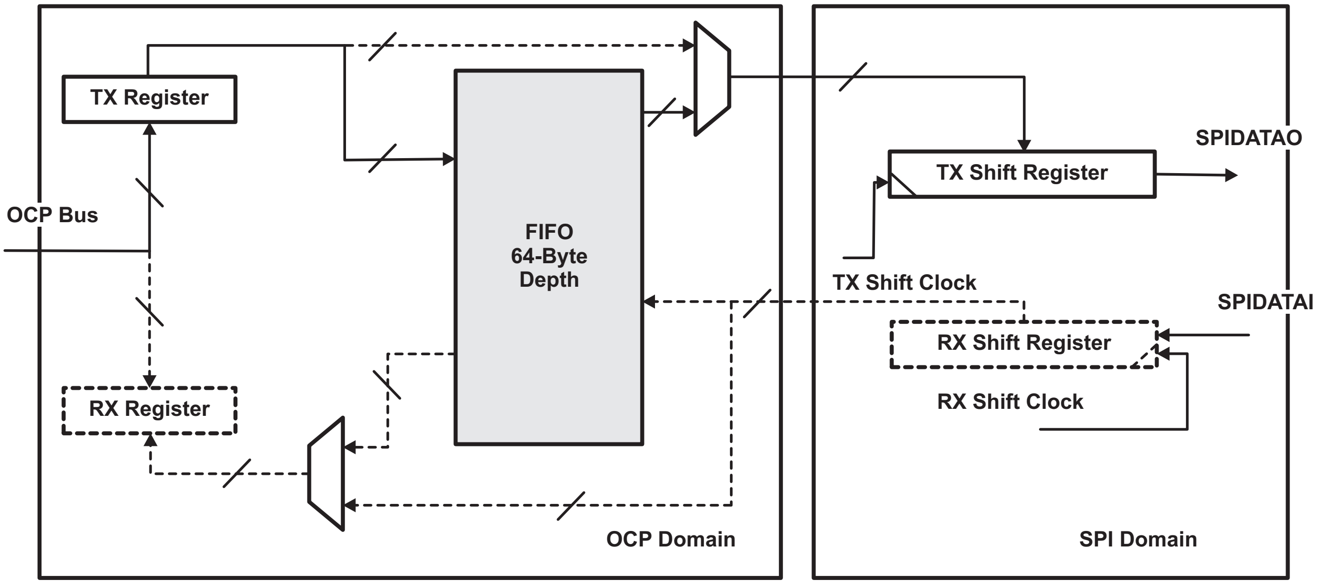
Configuration:
MCSPI_CH(i)CONF[TRM]=0x0 Transmit/receive mode
MCSPI_CH(i)CONF[FFRE]=0x0 FIFO disabled on receive path
MCSPI_CH(i)CONF[FFWE]=0x1 FIFO enabled on transmit path

Figure 26-16. Transmit/Receive Mode With Both FIFO Direction Used



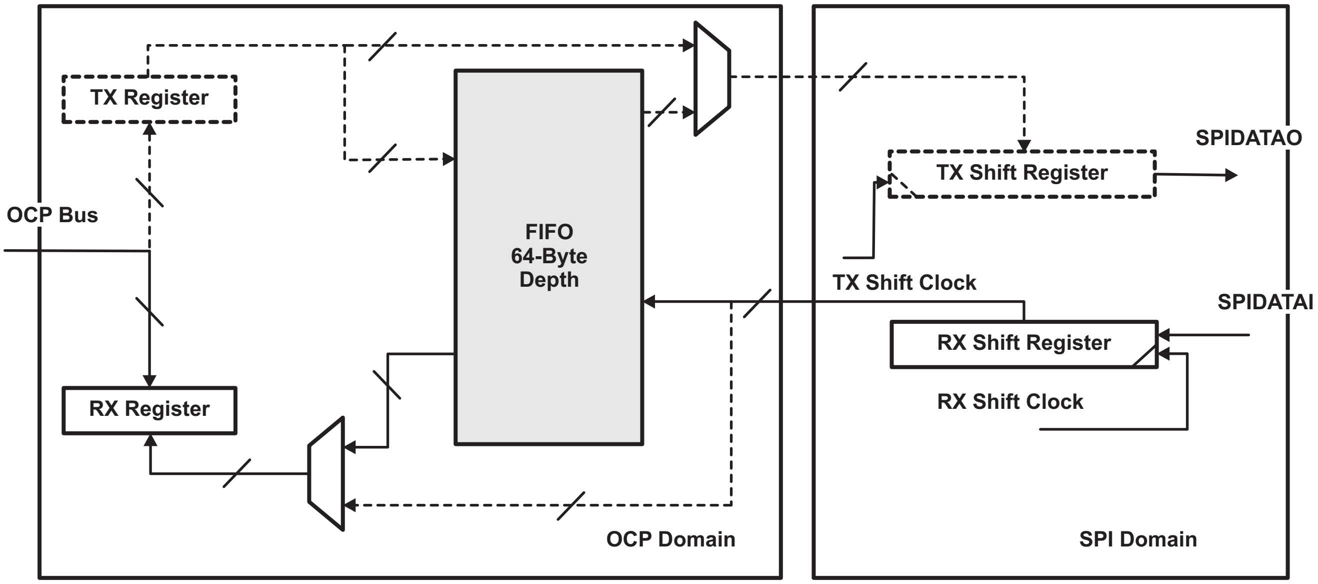
Configuration:
MCSPI_CH(i)CONF[TRM]=0x0 Transmit/receive mode
MCSPI_CH(i)CONF[FFRE]=0x1 FIFO enabled on receive path
MCSPI_CH(i)CONF[FFWE]=0x0 FIFO disabled on transmit path

Figure 26-17. Transmit-Only Mode With FIFO Used



Configuration:
MCSPI_CH(i)CONF[TRM]=0x2 Transmit only mode
MCSPI_CH(i)CONF[FFRE]=0x1 FIFO enabled on transmit path
MCSPI_CH(i)CONF[FFWE] not applicable

Figure 26-18. Receive-Only Mode With FIFO Used



Configuration:
MCSPI_CH(i)CONF[TRM]=012 Receive only mode
MCSPI_CH(i)CONF[FFRE]=0x1 FIFO enabled on receive path
MCSPI_CH(i)CONF[FFWE] not applicable

26.3.2.10.2 Buffer Almost Full

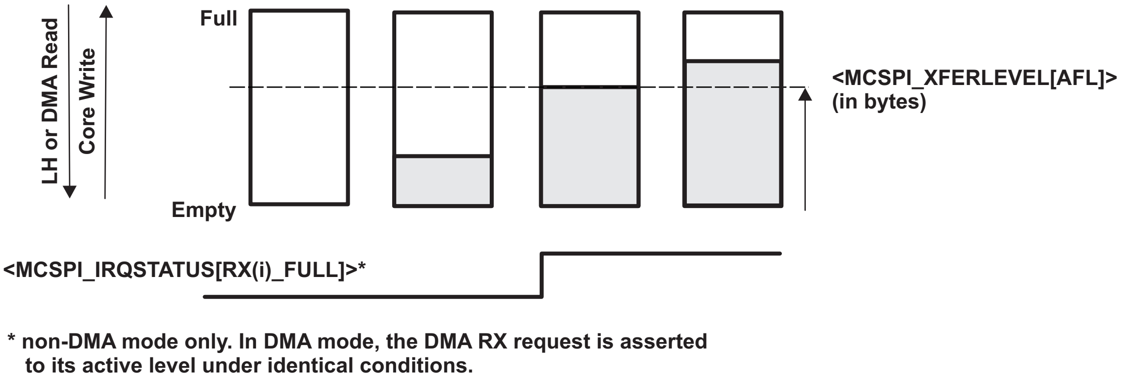
The bit field MCSPI_XFERLEVEL[AFL] is needed when the buffer is used to receive SPI word from a slave (MCSPI_CH(i)CONF[FFER] must be set to 1). It defines the almost full buffer status.

When FIFO pointer reaches this level an interrupt or a DMA request is sent to the CPU to enable system to read AFL+1 bytes from receive register. Be careful AFL+1 must correspond to a multiple value of MCSPI_CH(i)CONF[WL].

When DMA is used, the request is de-asserted after the first receive register read.

No new request will be asserted until the system has performed the correct number of read operations from the buffer.

Figure 26-19. Buffer Almost Full Level (AFL)



.

NOTE: SPI_IRQSTS register bits are not available in DMA mode. In DMA mode, the SPIm_DMA_RX*n* request is asserted on the same conditions as the SPI_IRQSTS RX*n*_FULL flag.

26.3.2.10.3 Buffer Almost Empty

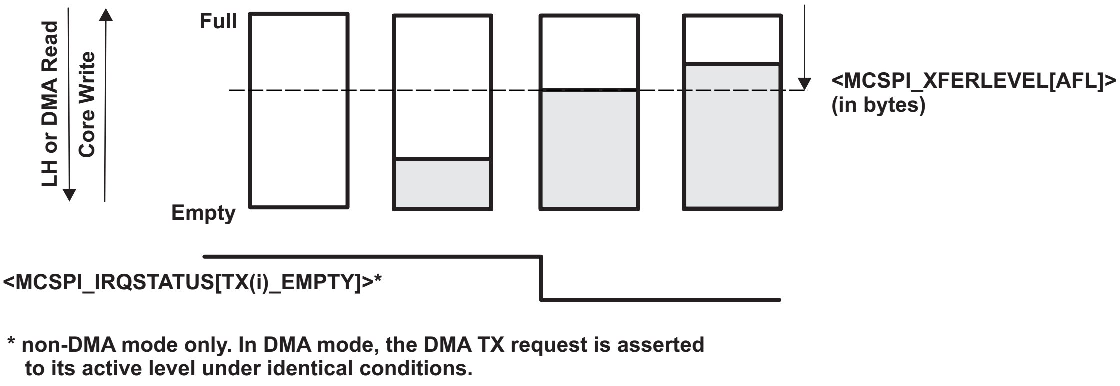
The bitfield MCSPI_XFERLEVEL[AEL] is needed when the buffer is used to transmit SPI word to a slave (MCSPI_CH(i)CONF[FFEW]must be set to 1). It defines the almost empty buffer status.

When FIFO pointer has not reached this level an interrupt or a DMA request is sent to the CPU to enable system to write AEL+1 bytes to transmit register. Be careful AEL+1 must correspond to a multiple value of MCSPI_CH(i)CONF[WL].

When DMA is used, the request is de-asserted after the first transmit register write.

No new request will be asserted until the system has performed the correct number of write operations.

Figure 26-20. Buffer Almost Empty Level (AEL)



.

26.3.2.10.4 End of Transfer Management

When the FIFO buffer is enabled for a channel, the user should configure the MCSPI_XFERLEVEL register, the AEL and AFL levels, and, especially, the WCNT bit field to define the number of SPI word to be transferred using the FIFO. This should be done before enabling the channel.

This counter allows the controller to stop the transfer correctly after a defined number of SPI words have been transferred. If WNCT is cleared to 0, the counter is not used and the user must stop the transfer manually by disabling the channel, in this case the user doesn’t know how many SPI transfers have been done. For receive transfer, software shall poll the corresponding FFE bit field and read the Receive register to empty the FIFO buffer.

When End Of Word count interrupt is generated, the user can disable the channel and poll on MCSPI_CH(i)STAT[FFE] register to know if SPI word is still there in FIFO buffer and read last words.

26.3.2.10.5 Multiple SPI Word Access

The CPU has the ability to perform multiple SPI word access to the receive or transmit registers within a single 32-bit OCP access by setting the bit field MCSPI_MODULCTRL[MOA] to ‘1’ under specific conditions:

- The channel selected has the FIFO enable.
- Only FIFO sense enabled support the kind of access.
- The bit field MCSPI_MODULCTRL[MOA] is set to 1
- Only 32-bit OCP access and data width can be performed to receive or transmit registers, for other kind of access the CPU must de-assert MCSPI_MODULCTRL[MOA] bit fields.
- The Level MCSPI_XFERLEVEL[AEL] and MCSPI_XFERLEVEL[AFL] must be 32-bit aligned , it means that AEL[0] = AEL[1] = 1 or AFL[0] = AFL[1] = 1.
- If MCSPI_XFERLEVEL[WCNT] is used it must be configured according to SPI word length.
- The word length of SPI words allows to perform multiple SPI access, that means that MCSPI_CH(i)CONF[WL] < 16.

Number of SPI word access depending on SPI word length:

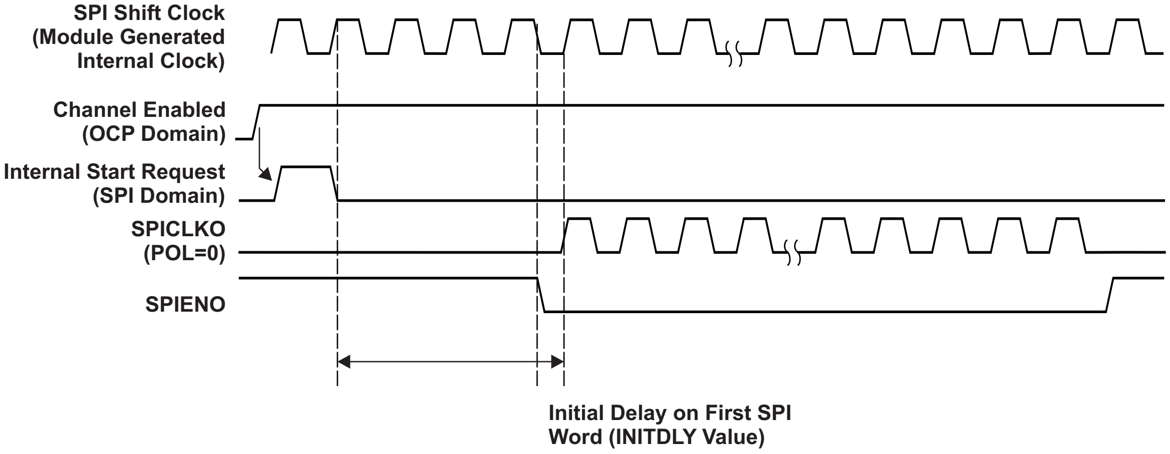
- 3 ≤ WL ≤ 7, SPI word length smaller or equal to byte length, four SPI words accessed per 32-bit OCP read/write. If word count is used (MCSPI_XFERLEVEL[WCNT]), set the bit field to WCNT[0]=WCNT[1]=0.
- 8 ≤ WL ≤ 15, SPI word length greater than byte or equal to 16-bit length, two SPI words accessed per 32-bit OCP read/write. If word count is used (MCSPI_XFERLEVEL[WCNT]), set the bit field to WCNT[0]= 0.
- 16 ≤ WL multiple SPI word access not applicable.

26.3.2.11 First SPI Word Delayed

The McSPI controller has the ability to delay the first SPI word transfer to give time for system to complete some parallel processes or fill the FIFO in order to improve transfer bandwidth. This delay is applied only on first SPI word after SPI channel enabled and first write in Transmit register. It is based on output clock frequency.

This option is meaningful in master mode and single channel mode, MCSPI_MODULCTRL[SINGLE] = 1.

Figure 26-21. Master Single Channel Initial Delay



Few delay values are available: No delay, 4/8/16/32 SPI cycles.

Its accuracy is half cycle in clock bypass mode and depends on clock polarity and phase.

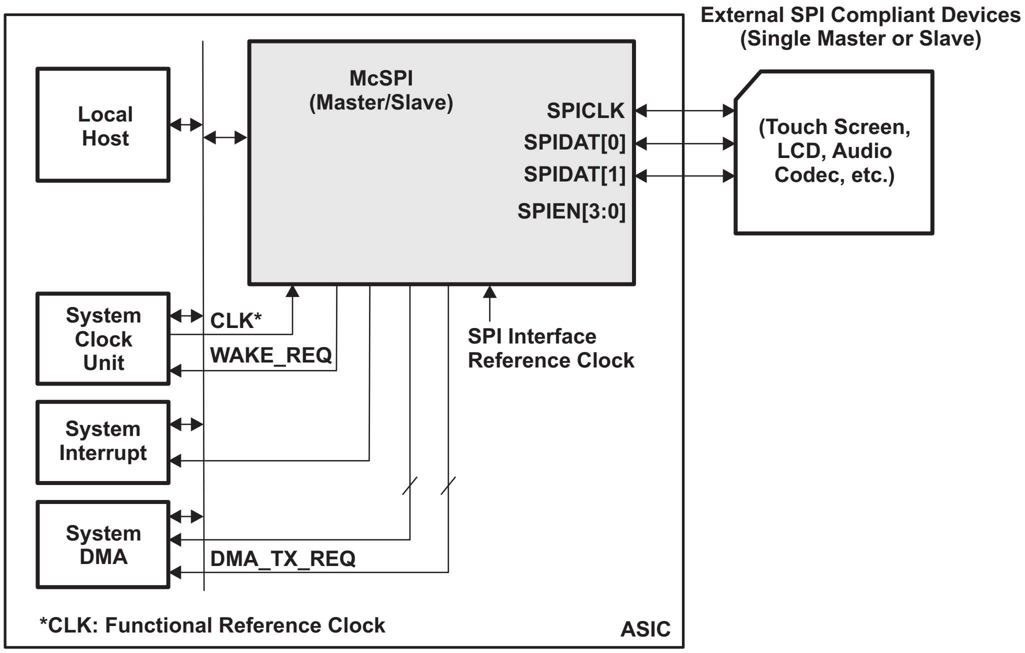
26.3.2.12 3- or 4-Pin Mode

External SPI bus interface can be configured to use a restricted set of pins using the bit field MCSPI_MODULCTRL[PIN34] and depending on targeted application:

- If MCSPI_MODULCTRL[PIN34] is cleared to 0 (default value) the controller is in 4-pin mode using the SPI pins SPICLK, SOMI, SIMO and chip enable CS.
- If MCSPI_MODULCTRL[PIN34] is set to 1 the controller is in 3-pin mode using the SPI pins SPICLK, SOMI and SIMO.

In 3-pin mode it is mandatory to put the controller in single channel master mode (MCSPI_MODULCTRL[SINGLE] asserted) and to connect only one SPI device on the bus.

Figure 26-22. 3-Pin Mode System Overview



In 3-pin mode all options related to chip select management are useless:

- MCSPI_CHxCONF[EPOL]
- MCSPI_CHxCONF[TCS0]
- MCSPI_CHxCONF[FORCE]

The chip select pin SPIEN is forced to ‘0’ in this mode.

26.3.3 Slave Mode

McSPI is in slave mode when the bit MS of the register MCSPI_MODULCTRL is set.

In slave mode, McSPI can be connected to up to 4 external SPI master devices. McSPI handles transactions with a single SPI master device at a time.

In slave mode, McSPI initiates data transfer on the data lines (SPIDAT[1;0]) when it receives an SPI clock (SPICLK) from the external SPI master device.

The controller is able to work with or without a chip select SPIEN depending on MCSPI_MODULCTRL[PIN34] bit setting. It also supports transfers without a dead cycle between two successive words.

26.3.3.1 Dedicated Resources

In slave mode, enabling a channel that is not channel 0 has no effect. Only channel 0 can be enabled. The channel 0, in slave mode has the following resources:

- Its own channel enable, programmable with the bit EN of the register MCSPI_CH0CTRL. This channel should be enabled before transmission and reception. Disabling the channel, outside data word transmission, remains under user responsibility.
- Any of the 4 ports SPIEN[3:0] can be used as a slave SPI device enable. This is programmable with the bits SPIENSLV of the register MCSPI_CH0CONF.
- Its own transmitter register MCSPI_TX on top of the common shift register. If the transmitter register is empty, the status bit TXS of the register MCSPI_CH0STAT is set. When McSPI is selected by an external master (active signal on the SPIEN port assigned to channel 0), the transmitter register content of channel0 is always loaded in shift register whether it has been updated or not. The transmitter register should be loaded before McSPI is selected by a master.
- Its own receiver register MCSPI_RX on top of the common shift register. If the receiver register is full, the status bit RXS of the register MCSPI_CH0STAT is set.

NOTE: The transmitter register and receiver registers of the other channels are not used. Read from or Write in the registers of a channel other than 0 has no effect.

- Its own communication configuration with the following parameters via the register MCSPI_CH0CONF:
  - Transmit/Receive modes, programmable with the bit TRM.
  - Interface mode (Two data pins or Single data pin) and data pins assignment, both programmable with the bits IS and DPE.
  - SPI word length, programmable with the bits WL.
  - SPIEN polarity, programmable with the bit EPOL.
  - SPICLK polarity, programmable with the bit POL.
  - SPICLK phase, programmable with the bit PHA.
  - Use a FIFO buffer or not, programmable with FFER and FFEW, depending on transfer mode TRM.

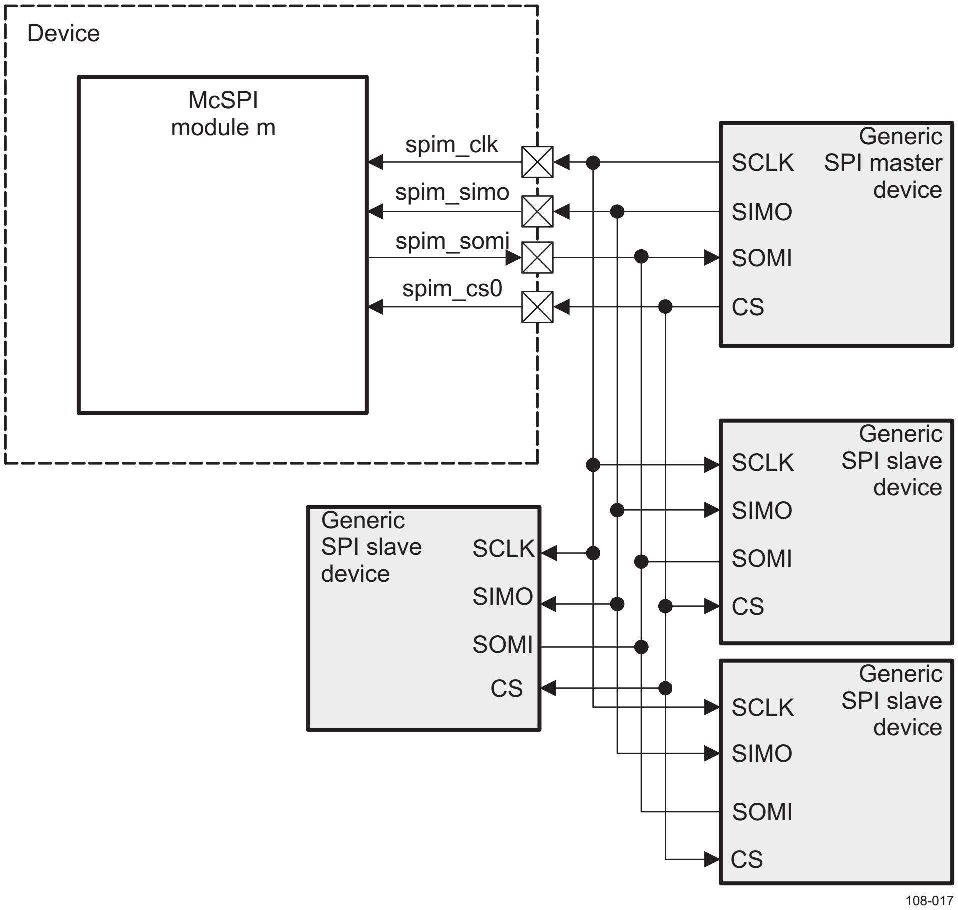
The SPICLK frequency of a transfer is controlled by the external SPI master connected to McSPI. The bits CLKD0 of the MCSPI_CH0CONF register are not used in slave mode.

NOTE: The configuration of the channel can be loaded in the MCSPI_CH0CONF register only when the channel is disabled.

- Two DMA requests events, read and write, to synchronize read/write accesses of the DMA controller with the activity of McSPI. The DMA requests are enabled with the bits DMAR and DMAW of the MCSPI_CH0CONF register.
- Four interrupts events.

Figure 26-23 shows an example of four slaves wired on a single master device.

Figure 26-23. Example of SPI Slave with One Master and Multiple Slave Devices on Channel 0



26.3.3.2 Interrupt Events in Slave Mode

The interrupt events related to the transmitter register state are TX_empty and TX_underflow. The interrupt events related to the receiver register state are RX_full and RX_overflow.

26.3.3.2.1 TX_EMPTY

The event TX_empty is activated when the channel is enabled and its transmitter register becomes empty. Enabling channel automatically raises this event. When FIFO buffer is enabled (MCSPI_CH(i)CONF[FFEW] set to 1), the TX_empty is asserted as soon as there is enough space in buffer to write a number of byte defined by MCSPI_XFERLEVEL[AEL].

Transmitter register must be load to remove source of interrupt and TX_empty interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

When FIFO is enabled, no new TX_empty event will be asserted unless the host performs the number of writes to the transmitter register defined by MCSPI_XFERLEVEL[AEL]. It is the responsibility of the Local Host to perform the right number of writes.

26.3.3.2.2 TX_UNDERFLOW

The event TX_underflow is activated when channel is enabled and if the transmitter register or FIFO (if use of buffer is enabled) is empty (not updated with new data) when an external master device starts a data transfer with McSPI (transmit and receive).

When the FIFO is enabled, the data read while the underflow flag is set will not be the last word written to the FIFO.

The TX_underflow indicates an error (data loss) in slave mode.

To avoid having TX_underflow event at the beginning of a transmission, the event TX_underflow is not activated when no data has been loaded into the transmitter register since channel has been enabled.

TX_underflow interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

26.3.3.2.3 RX_FULL

The event RX_FULL is activated when channel is enabled and receiver becomes filled (transient event). When FIFO buffer is enabled (MCSPI_CH(i)CONF[FFER] set to 1), the RX_FULL is asserted as soon as there is a number of bytes holds in buffer to read defined by MCSPI_XFERLEVEL[AFL].

Receiver register must be read to remove source of interrupt and RX_full interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

When FIFO is enabled, no new RX_FULL event will be asserted unless the host has performed the number of reads from the receive register defined by MCSPI_XFERLEVEL[AFL]. It is the responsibility of Local Host to perform the right number of reads.

26.3.3.2.4 RX_OVERFLOW

The RX0_OVERFLOW event is activated in slave mode in either transmit-and-receive or receive-only mode, when a channel is enabled and the SPI_RX*n* register or FIFO is full when a new SPI word is received. The SPI_RX*n* register is always overwritten with the new SPI word. If the FIFO is enabled, data within the FIFO is overwritten, it must be considered as corrupted. The RX0_OVERFLOW event should not appear in slave mode using the FIFO.

The RX0_OVERFLOW indicates an error (data loss) in slave mode.

The SPI_IRQSTS[3] RX0_OVERFLOW interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

26.3.3.2.5 End of Word Count

The event end of word (EOW) count is activated when channel is enabled and configured to use the built-in FIFO. This interrupt is raised when the controller had performed the number of transfer defined in MCSPI_XFERLEVEL[WCNT] register. If the value was programmed to 0000h, the counter is not enabled and this interrupt is not generated.

The EOW count interrupt also indicates that the SPI transfer has halted on the channel using the FIFO buffer.

The EOW interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

26.3.3.3 Slave Transmit-and-Receive Mode

The slave transmit and receive mode is programmable (TRM bit cleared to 0 in the register MCSPI_CH(i)CONF).

After the channel is enabled, transmission and reception proceeds with interrupt and DMA request events.

In slave transmit and receive mode, transmitter register should be loaded before McSPI is selected by an external SPI master device.

Transmitter register or FIFO (if enabled) content is always loaded into the shift register whether it has been updated or not. The event TX_underflow is activated accordingly, and does not prevent transmission.

On completion of SPI word transfer (bit EOT of the register MCSPI_CH(i)STAT is set) the received data is transferred to the channel receive register. This bit is meaningless when using the Buffer for this channel.

The built-in FIFO is available in this mode and can be configured in one data direction, transmit or receive, then the FIFO is seen as a unique 64-byte buffer. It can also be configured in both data directions, transmit and receive, then the FIFO is split into two separate 32-byte buffers with their own address space management.

26.3.3.4 Slave Receive-Only Mode

The slave receive-only mode is programmable (MCSPI_CH(i)CONF[TRM] set to 01).

In receive-only mode, the transmitter register should be loaded before McSPI is selected by an external SPI master device. Transmitter register or FIFO (if enabled) content is always loaded into the shift register whether it has been updated or not. The event TX_underflow is activated accordingly, and does not prevent transmission.

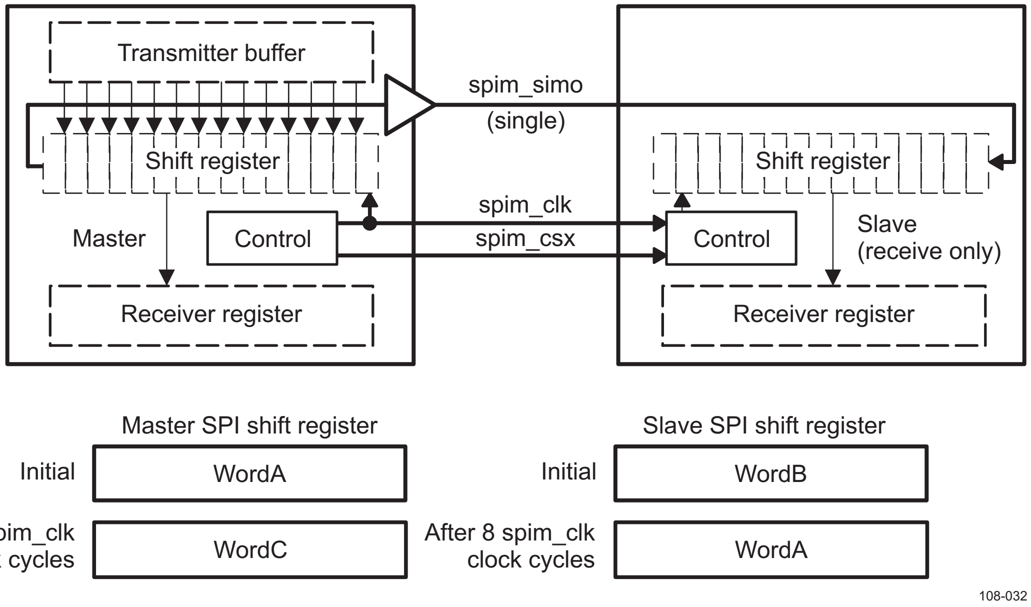
When an SPI word transfer completes (the MCSPI_CH(i)STAT[EOT] bit (with *I* = 0) is set to 1), the received data is transferred to the channel receive register.

To use McSPI as a slave receive-only device with MCSPI_CH(i)CONF[TRM]=00, the user has the responsibility to disable the TX_empty and TX_underflow interrupts and DMA write requests due to the transmitter register state.

On completion of SPI word transfer (bit EOT of the register MCSPI_CH(i)STAT is set) the received data is transferred to the channel receive register. This bit is meaningless when using the Buffer for this channel. The built-in FIFO is available in this mode and can be configured with FFER bit field in the MCSPI_CH(i)CONF register, then the FIFO is seen as a unique 64-byte buffer.

Figure 26-24 shows an example of a half-duplex system with a master device on the left and a receive-only slave device on the right. Each time one bit transfers out from the master, one bit transfers in to the slave. If WordA is 8 bits, then after eight cycles of the serial clock spim_clk, WordA transfers from the master to the slave.

Figure 26-24. SPI Half-Duplex Transmission (Receive-Only Slave)



26.3.3.5 Slave Transmit-Only Mode

The slave transmit-only mode is programmable (MCSPI_CH(i)CONF[TRM] set to 10). This mode eliminates the need for the CPU to read the receiver register (minimizing data movement) when only transmission is meaningful.

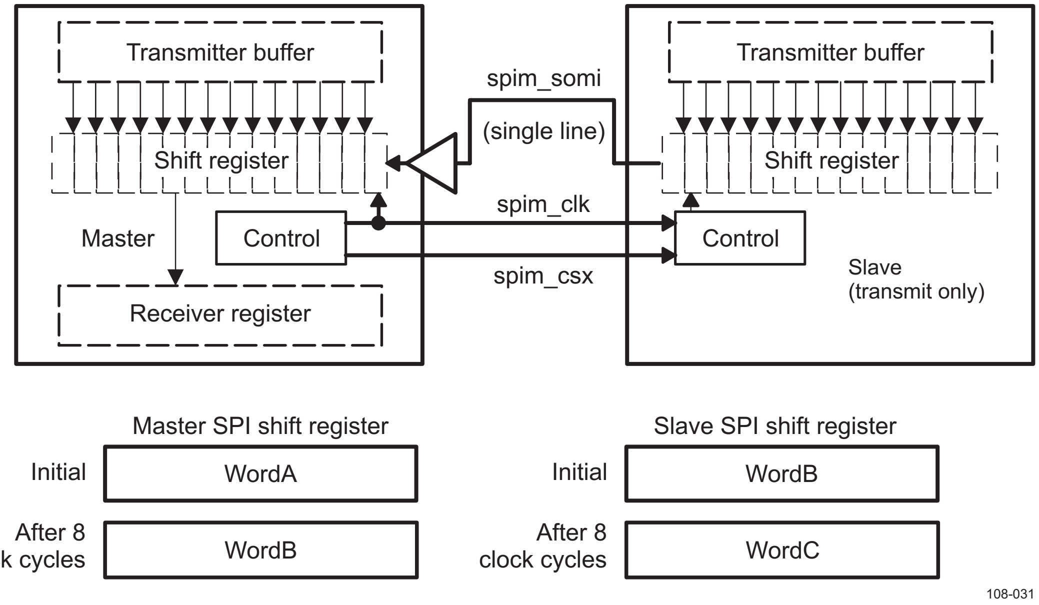
To use McSPI as a slave transmit-only device with MCSPI_CH(i)CONF[TRM]=10, the user should disable the RX_full and RX_overflow interrupts and DMA read requests due to the receiver register state.

On completion of SPI word transfer the bit EOT of the register MCSPI_CH(i)STAT is set. This bit is meaningless when using the Buffer for this channel.

The built-in FIFO is available in this mode and can be configured with FFER bit field in the MCSPI_CH(i)CONF register, then the FIFO is seen as a unique 64-byte buffer.

Figure 26-25 shows a half-duplex system with a master device on the left and a transmit-only slave device on the right. Each time a bit transfers out from the slave device, one bit transfers in the master. If WordB is 8-bits, then after eight cycles of the serial clock spim_clk, WordB transfers from the slave to the master.

Figure 26-25. SPI Half-Duplex Transmission (Transmit-Only Slave)



26.3.4 Interrupts

According to its transmitter register state and its receiver register state each channel can issue interrupt events if they are enabled.

The interrupt events are listed in the Section 26.3.2.2 and in Section 26.3.3.2.

Each interrupt event has a status bit, in the MCSPI_IRQSTS register, which indicates service is required, and an interrupt enable bit, in the MCSPI_IRQEN register, which enables the status to generate hardware interrupt requests.

When an interrupt occurs and it is later masked (IRQEN), the interrupt line is not asserted again even if the interrupt source has not been serviced.

McSPI supports interrupt driven operation and polling.

26.3.4.1 Interrupt-Driven Operation

Alternatively, an interrupt enable bit, in the MCSPI_IRQEN register, can be set to enable each of the events to generate interrupt requests when the corresponding event occurs. Status bits are automatically set by hardware logic conditions.

When an event occurs (the single interrupt line is asserted), the CPU must:

- Read the MCSPI_IRQSTS register to identify which event occurred,
- Read the receiver register that corresponds to the event in order to remove the source of an RX_full event, or write into the transmitter register that corresponds to the event in order to remove the source of a TX_empty event. No action is needed to remove the source of the events TX_underflow and RX_overflow.
- Write a 1 into the corresponding bit of MCSPI_IRQSTS register to clear the interrupt status, and release the interrupt line.

The interrupt status bit should always be reset after the channel is enabled and before the event is enabled as an interrupt source.

26.3.4.2 Polling

When the interrupt capability of an event is disabled in the MCSPI_IRQEN register, the interrupt line is not asserted and:

- The status bits in the MCSPI_IRQSTS register can be polled by software to detect when the corresponding event occurs.
- Once the expected event occurs, CPU must read the receiver register that corresponds to the event in order to remove the source of an RX_full event, or write into the transmitter register that corresponds to the event in order to remove the source of a TX_empty event. No action is needed to remove the source of the events TX_underflow and RX_overflow.
- Writing a 1 into the corresponding bit of MCSPI_IRQSTS register clears the interrupt status and does not affect the interrupt line state.

26.3.5 DMA Requests

McSPI can be interfaced with a DMA controller. At system level, the advantage is to free the local host of the data transfers.

According to its transmitter register state, its receiver register state or FIFO level (if use of buffer for the channel) each channel can issue DMA requests if they are enabled.

The DMA requests need to be disabled in order to get TX and RX interrupts, in order to define either the end of the transfer or the transfer of the last words for the modes listed below:

- Master transmit-only
- Master normal receive-only mode
- Master turbo receive-only mode
- Slave transmit-only

There are two DMA request lines per channel. The management of DMA requests differ according to use of FIFO buffer or not.

26.3.5.1 FIFO Buffer Disabled

The DMA Read request line is asserted when the channel is enabled and a new data is available in the receive register of the channel. DMA Read request can be individually masked with the bit DMAR of the register MCSPI_CH(i)CONF. The DMA Read request line is de-asserted on read completion of the receive register of the channel.

The DMA Write request line is asserted when the channel is enabled and the transmitter register of the channel is empty. DMA Write request can be individually masked with the bit DMAW of the register MCSPI_CH(i)CONF. The DMA Write request line is de-asserted on load completion of the transmitter register of the channel.

Only one SPI word can be transmitted/received per OCP bus access to write/read the transmit or receive register.

26.3.5.2 FIFO Buffer Enabled

The DMA Read request line is asserted when the channel is enabled and a number of bytes defined in MCSPI_XFERLEVEL[AFL] bit field is hold in FIFO buffer for the receive register of the channel. DMA Read request can be individually masked with the bit DMAR of the register MCSPI_CH(i)CONF. The DMA Read request line is de-asserted on the first SPI word read completion of the receive register of the channel. No new DMA request will be asserted again as soon as user has not performed the right number of read accesses defined by MCSPI_XFERLEVEL[AFL] it is under user responsibility.

The DMA Write request line is asserted when the channel is enabled and the number of bytes hold in FIFO buffer is below the level defined by the MCSPI_XFERLEVEL[AEL] bit field. DMA Write request can be individually masked with the bit DMAW of the register MCSPI_CH(i)CONF. The DMA Write request line is de-asserted on load completion of the first SPI word in the transmitter register of the channel. No new DMA request will be asserted again as soon as user has not performed the right number of write accesses defined by MCSPI_XFERLEVEL[AEL] it is under user responsibility.

Only one SPI word can be transmitted/received per OCP bus access to write/read the transmit or receive FIFO.

26.3.5.3 DMA 256-Bit Aligned Addresses

The controller has two registers, MCSPI_DAFTX and MCSPI_DAFRX, used only with an enabled channel which manages the FIFO to be compliant the a DMA handler providing only 256-bit aligned addresses.

This features is activated when the bit field MCSPI_MODULCTRL[FDAA] is set to ‘1’ and only one enabled channel have its bit field MCSPI_CH(i)CONF[FFEW] or MCSPI_CH(i)CONF[FFER] enabled.

In this case the registers MCSPI_TX(i) and MCSPI_RX(i) are not used and data is managed through registers MCSPI_DAFTX and MCSPI_DAFRX.

26.3.6 Emulation Mode

The MReqDebug input differentiates a regular access of a processor (application access), from an emulator access.

Application access: MReqDebug = 0

In functional mode, the consequences of a read of a receiver register MCSPI_RX(i) are the following:

- The source of an RX(i)_Full event in the MCSPI_IRQSTS register is removed, if it was enabled in the MCSPI_IRQEN register.
- The RX(i)S status bit in the MCSPI_IRQSTS register is cleared.
- In master mode, depending on the round robin arbitration, and the transmitter register state, the channel may access to the shift register for transmission/reception.

Emulator access: MReqDebug = 1

In emulation mode, McSPI behavior is the same as in functional mode but a read of a receiver register MCSPI_RX(i) is not intrusive:

- MCSPI_RX(i) is still considered as not read. When the FIFO buffer is enabled, pointers are not updated.
- The source of an RX(i)_Full event in the MCSPI_IRQSTS register is not removed. The RX(i)S status bit in the MCSPI_CH(i)STAT register is held steady.

In emulation mode, as in functional mode, based on the ongoing data transfers, the status bits of the MCSPI_CH(i)STAT register may be optionally updated, the interrupt and DMA request lines may be optionally asserted.

26.3.7 Power Saving Management

Independently of the module operational modes (Transmit and/or Receive), two modes of operations are defined from a power management perspective: normal and idle modes.

The two modes are temporally fully exclusive.

26.3.7.1 Normal Mode

Both the Interface, or OCP, clock and SPI clock (CLKSPIREF) provided to McSPI must be active for both master and slave modes. The auto-gating of the module OCP clock and SPI clock occurs when the following conditions are met:

- The bit AutoIdle of the register MCSPI_SYSCONFIG is set.
- In master mode, there is no data to transmit or receive in all channels.
- In slave mode, the SPI is not selected by the external SPI master device and no OCP accesses.

Autogating of the module OCP clock and SPI clock stops when the following conditions are met:

- In master mode, an OCP access occurs.
- In slave mode, an OCP access occurs or McSPI is selected by an external SPI master device.

26.3.7.2 Idle Mode

The OCP clock and SPI clock provided to McSPI may be switched off on system power manager request and switched back on module request.

McSPI is compliant with the power management handshaking protocol: idle request from the system power manager, idle acknowledgement from McSPI.

The idle acknowledgement in response to an idle request from the system power manager varies according to a programmable mode in the MCSPI_SYSCONFIG register: No idle mode, force idle mode, and smart idle mode.

- When programmed for no idle mode (the bit SIdleMode of the register MCSPI_SYSCONFIG is set to “01”), the module ignores the system power manager request, and behaves normally, as if the request was not asserted.
- When programmed for smart idle mode (the bit SIdleMode of the register MCSPI_SYSCONFIG is set to “10”), the module acknowledges the system power manager request according to its internal state.
- When programmed for force idle mode (the bit SIdleMode of the register MCSPI_SYSCONFIG is set to “00”), the module acknowledges the system power manager request unconditionally.

The OCP clock will be optionally switched off, during the smart idle mode period, if the bit ClockActivity of the register MCSPI_SYSCONFIG is set.

The SPI clock will be optionally switched off, during the smart idle mode period, if the second bit ClockActivity of the register MCSPI_SYSCONFIG is set.

McSPI assumes that both clocks may be switched off whatever the value set in the field ClockActivity of the register MCSPI_SYSCONFIG.

26.3.7.2.1 Transitions from Normal Mode to Smart-Idle Mode

The module detects an idle request when the synchronous signal IdleReq is asserted.

When IdleReq is asserted, any access to the module will generate an error as long as the OCP clock is alive.

When configured as a slave device, McSPI responds to the idle request by asserting the SIdleAck signal (idle acknowledgement) only after completion of the current transfer (SPIEN slave selection signal deasserted by the external master) and if interrupt or DMA request lines are not asserted.

As a master device, McSPI responds to the idle request by asserting the SIdleAck signal (idle acknowledgement) only after completion of all the channel data transfers and if interrupt or DMA request lines are not asserted.

As long as SIdleAck is not asserted, if an event occurs, the module can still generate an interrupt or a DMA request after IdleReq assertion. In this case, the module ignores the idle request and SIdleAck will not get asserted: The system power manager will abort the power mode transition procedure. It is then the responsibility of the system to de-assert IdleReq before attempting to access the module.

When SIdleAck is asserted, the module does not assert any new interrupt or DMA request.

26.3.7.2.2 Transition From Smart-Idle Mode to Normal mode

McSPI detects the end of the idle period when the idle request signal (IdleReq) is deasserted.

Upon IdleReq de-assertion, the module switches back to normal mode and de-asserts SIdleAck signal. The module is fully operational.

26.3.7.2.3 Force-Idle Mode

Force-idle mode is enabled as follows:

- The bit SIdleMode of the register MCSPI_SYSCONFIG is cleared to “00” (Force Idle).

  The force idle mode is an idle mode where McSPI responds unconditionally to the idle request by asserting the SIdleAck signal and by deasserting unconditionally the interrupt and DMA request lines if asserted.

  The transition from normal mode to idle mode does not affect the interrupt event bits of the MCSPI_IRQSTS register.

  In force-idle mode, the module is supposed to be disabled at that time, so the interrupt and DMA request lines are likely deasserted. OCP clock and SPI clock provided to McSPI can be switched off.

  An idle request during an SPI data transfer can lead to an unexpected and unpredictable result, and is under software responsibility.

  Any access to the module in force idle mode will generate an error as long as the OCP clock is alive and IdleReq is asserted.

The module exits the force idle mode when:

- The idle request signal (IdleReq) is de-asserted.

  Upon IdleReq de-assertion, the module switches back to normal mode and de-asserts SIdleAck signal. The module is fully operational. The interrupt and DMA request lines are optionally asserted a clock cycle later.

26.3.8 System Test Mode

McSPI is in system test mode (SYSTEST) when the bit System_Test of the register MCSPI_MODULCTRL is set.

The SYSTEST mode is used to check in a very simple manner the correctness of the system interconnect either internally to interrupt handler, or power manager, or externally to SPI I/Os.

I/O verification can be performed in SYSTEST mode by toggling the outputs and capturing the logic state of the inputs. (See MCSPI_SYST register definition in Section 26.4.1)

26.3.9 Reset

26.3.9.1 Internal Reset Monitoring

The module is reset by the hardware when an active-low reset signal, synchronous to the OCP interface clock is asserted on the input pin RESETN.

This hardware reset signal has a global reset action on the module. All configuration registers and all state machines are reset, in all clock domains.

Additionally, the module can be reset by software through the bit SoftReset of the register MCSPI_SYSCONFIG. This bit has exactly the same action on the module logic as the hardware RESETN signal. The register MCSPI_SYSCONFIG is not sensitive to software reset. The SoftReset control bit is active high. The bit is automatically reset to 0 by the hardware.

A global ResetDone status bit is provided in the status register MCSPI_SYSSTS. This bit is set to 1 when all the different clock domains resets (OCP domain and SPI domains) have been released (logical AND).

The global ResetDone status bit can be monitored by the software to check if the module is ready-to-use following a reset (either hardware or software).

The clock CLKSPIREF must be provided to the module, in order to allow the ResetDone status bit to be set.

When used in slave mode, the clock CLKSPIREF is needed only during the reset phase. The clock CLKSPIREF can be switched off after the ResetDone status is set.

26.3.9.2 Reset Values of Registers

The reset values of registers and signals are described in .

26.3.10 Access to Data Registers

This section details the supported data accesses (read or write) from/to the data receiver registers MCSPI_RX(i) and data transmitter registers MCSPI_TX(i).

Supported access:

McSPI supports only one SPI word per register (receiver or transmitter) and does not support successive 8-bit or 16-bit accesses for a single SPI word.

The SPI word received is always right justified on LSbit of the 32bit register MCSPI_RX(i), and the SPI word to transmit is always right justified on LSbit of the 32bit register MCSPI_TX(i).

The upper bits, above SPI word length, are ignored and the content of the data registers is not reset between the SPI data transfers.

The coherence between the number of bits of the SPI Word, the number of bits of the access and the enabled byte remains under the user’s responsibility. Only aligned accesses are supported.

In Master mode, data should not be written in the transmit register when the channel is disabled.

26.3.11 Programming Aid

26.3.11.1 Module Initialization

- Hard or soft reset.
- Read MCSPI_SYSSTS.
- Check if reset is done.
- Module configuration: (a) Write into MCSPI_MODULCTRL (b) Write into MCSPI_SYSCONFIG.
- Before the ResetDone bit is set, the clocks CLK and CLKSPIREF must be provided to the module.
- To avoid hazardous behavior, it is advised to reset the module before changing from MASTER mode to SLAVE mode or from SLAVE mode to MASTER mode.

26.3.11.2 Common Transfer Sequence

McSPI module allows the transfer of one or several words, according to different modes:

- MASTER, MASTER Turbo, SLAVE
- TRANSMIT - RECEIVE, TRANSMIT ONLY, RECEIVE ONLY
- Write and Read requests: Interrupts, DMA
- SPIEN lines assertion/deassertion: automatic, manual

For all these flows, the host process contains the main process and the interrupt routines. The interrupt routines are called on the interrupt signals or by an internal call if the module is used in polling mode.

In multi-channel master mode, the flows of different channels can be run simultaneously.

26.3.11.3 Main Program

- Interrupt Initialization: (a) Reset status bits in MCSPI_IRQSTS (b) Enable interrupts in MCSPI_IRQENA.
- Channel Configuration: Write MCSPI_CH(i)CONF.
- Start the channel: Write 0000 0001h in MCSPI_CH(i)CTRL.
- First write request: TX empty - Generate DMA write event/ polling TX empty flag by CPU to write First transmit word into MCSPI_TX(i).
- End of transfer: Stop the channel by writing 0000 0000h in MCSPI_CH(i)CTRL

The end of transfer depends on the transfer mode.

In multi-channel master mode, be careful not to overwrite the bits of other channels when initializing MCSPI_IRQSTS and MCSPI_IRQEN.

26.3.12 Interrupt and DMA Events

McSPI has two DMA requests (Rx and Tx) per channel. It also has one interrupt line for all the interrupt requests.

26.4 McSPI Registers

26.4.1 MCSPI Registers

Table 26-10 lists the memory-mapped registers for the MCSPI. All register offset addresses not listed in Table 26-10 should be considered as reserved locations and the register contents should not be modified.

Table 26-10. MCSPI Registers

Offset	Acronym	Register Name	Section
0h	MCSPI_HL_REV		Section 26.4.1.1
4h	MCSPI_HL_HWINFO		Section 26.4.1.2
10h	MCSPI_HL_SYSCONFIG		Section 26.4.1.3
100h	MCSPI_REVISION		Section 26.4.1.4
110h	MCSPI_SYSCONFIG		Section 26.4.1.5
114h	MCSPI_SYSSTS		Section 26.4.1.6
118h	MCSPI_IRQSTS		Section 26.4.1.7
11Ch	MCSPI_IRQEN		Section 26.4.1.8
120h	MCSPI_WAKEUPEN		Section 26.4.1.9
124h	MCSPI_SYST		Section 26.4.1.10
128h	MCSPI_MODULCTRL		Section 26.4.1.11
12Ch	MCSPI_CH0CONF		Section 26.4.1.12
130h	MCSPI_CH0STAT		Section 26.4.1.13
134h	MCSPI_CH0CTRL		Section 26.4.1.14
138h	MCSPI_TX0		Section 26.4.1.15
13Ch	MCSPI_RX0		Section 26.4.1.16
140h	MCSPI_CH1CONF		Section 26.4.1.17
144h	MCSPI_CH1STAT		Section 26.4.1.18
148h	MCSPI_CH1CTRL		Section 26.4.1.19
14Ch	MCSPI_TX1		Section 26.4.1.20
150h	MCSPI_RX1		Section 26.4.1.21
154h	MCSPI_CH2CONF		Section 26.4.1.22
158h	MCSPI_CH2STAT		Section 26.4.1.23
15Ch	MCSPI_CH2CTRL		Section 26.4.1.24
160h	MCSPI_TX2		Section 26.4.1.25

Table 26-10. MCSPI Registers (continued)

Offset	Acronym	Register Name	Section
164h	MCSPI_RX2		Section 26.4.1.26
168h	MCSPI_CH3CONF		Section 26.4.1.27
16Ch	MCSPI_CH3STAT		Section 26.4.1.28
170h	MCSPI_CH3CTRL		Section 26.4.1.29
174h	MCSPI_TX3		Section 26.4.1.30
178h	MCSPI_RX3		Section 26.4.1.31
17Ch	MCSPI_XFERLEVEL		Section 26.4.1.32
180h	MCSPI_DAFTX		Section 26.4.1.33
1A0h	MCSPI_DAFRX		Section 26.4.1.34

26.4.1.1 MCSPI_HL_REV Register (Offset = 0h) [reset = 80300000h]

Register mask: FFFFFFFFh

MCSPI_HL_REV is shown in Figure 26-26 and described in Table 26-11.

Return to Summary Table.

IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility

Figure 26-26. MCSPI_HL_REV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCHEME		RSVD		FUNC											
R-2h		R-0h		R-30h											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R_RTL					X_MAJOR			CUSTOM		Y_MINOR					
R-0h					R-0h			R-0h		R-0h					

Table 26-11. MCSPI_HL_REV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	2h	
29-28	RSVD	R	0h	
27-16	FUNC	R	30h	
15-11	R_RTL	R	0h	
10-8	X_MAJOR	R	0h	
7-6	CUSTOM	R	0h	
5-0	Y_MINOR	R	0h	

26.4.1.2 MCSPI_HL_HWINFO Register (Offset = 4h) [reset = 9h]

Register mask: FFFFFFFFh

MCSPI_HL_HWINFO is shown in Figure 26-27 and described in Table 26-12.

Return to Summary Table.

Information about the IP module's hardware configuration, i.e. typically the module's HDL generics (if any). Actual field format and encoding is up to the module's designer to decide.

Figure 26-27. MCSPI_HL_HWINFO Register

31	30	29	28	27	26	25	24
RSVD							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RSVD							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RSVD							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RSVD	RETMODE	FFNBYTE					USEFIFO
R-0h	R-0h	R-4h					R-1h

Table 26-12. MCSPI_HL_HWINFO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RSVD	R	0h	Reserved These bits are initialized to zero, and writes to them are ignored.
6	RETMODE	R	0h	This bit field indicates whether the retention mode is supported using the pin PIRFFRET. 0h (R) = Retention mode disabled 1h (R) = Retention mode enabled
5-1	FFNBYTE	R	4h	FIFO number of byte generic parameter This register defines the value of FFNBYTE generic parameter, only MSB bits from 8 down to 4 are taken into account. 1h (R) = FIFO 16 bytes depth 2h (R) = FIFO 32 bytes depth 4h (R) = FIFO 64 bytes depth 8h (R) = FIFO 128 bytes depth 10h (R) = FIFO 256 bytes depth
0	USEFIFO	R	1h	Use of a FIFO enable: This bit field indicates if a FIFO is integrated within controller design with its management. 0h (R) = FIFO not implemented in design 1h (R) = FIFO and its management implemented in design with depth defined by FFNBYTE generic.

26.4.1.3 MCSPI_HL_SYSCONFIG Register (Offset = 10h) [reset = 8h]

Register mask: FFFFFFFFh

MCSPI_HL_SYSCONFIG is shown in Figure 26-28 and described in Table 26-13.

Return to Summary Table.

Clock management configuration

Figure 26-28. MCSPI_HL_SYSCONFIG Register

31	30	29	28	27	26	25	24
RSVD							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RSVD							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RSVD							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RSVD				IDLEMODE		FREEEMU	SOFTRESET
R-0h				R/W-2h		R/W-0h	R/W-0h

Table 26-13. MCSPI_HL_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RSVD	R	0h	
3-2	IDLEMODE	R/W	2h	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state. 0h (R/W) = Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements.Backup mode, for debug only. 1h (R/W) = No-idle mode: local target never enters idle state.Backup mode, for debug only. 2h (R/W) = Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements.IP module shall not generate (IRQ- or DMA-request-related) wakeup events. 3h (R/W) = Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements.IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state.Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented.
1	FREEEMU	R/W	0h	Sensitivity to emulation (debug) suspend input signal. 0h (R/W) = IP module is sensitive to emulation suspend 1h (R/W) = IP module is not sensitive to emulation suspend
0	SOFTRESET	R/W	0h	Software reset. (Optional) 0h (W) = No action 0h (R) = Reset done, no pending action 1h (R) = Reset (software or other) ongoing 1h (W) = Initiate software reset

26.4.1.4 MCSPI_REVISION Register (Offset = 100h) [reset = X]

MCSPI_REVISION is shown in Figure 26-29 and described in Table 26-14.

Return to Summary Table.

This register contains the hard coded RTL revision number.

Figure 26-29. MCSPI_REVISION Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="24">RESERVED</td><td colspan="8">REV</td></tr>
<tr><td colspan="24">R-0h</td><td colspan="8">R-X</td></tr>
</table>

Table 26-14. MCSPI_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reads returns 0
7-0	REV	R	X	IP revision [7:4] Major revision [3:0] Minor revision Examples: 0x10 for 1.0, 0x21 for 2.1

26.4.1.5 MCSPI_SYSCONFIG Register (Offset = 110h) [reset = 15h]

Register mask: FFFFFFFFh

MCSPI_SYSCONFIG is shown in Figure 26-30 and described in Table 26-15.

Return to Summary Table.

This register allows controlling various parameters of the OCP interface.

Figure 26-30. MCSPI_SYSCONFIG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">CLOCKACTIVITY</td></tr>
<tr><td colspan="6">R/W-0h</td><td colspan="2">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">RESERVED</td><td colspan="2">SIDLEMODE</td><td>ENAWAKEUP</td><td>SOFTRESET</td><td>AUTOIDLE</td></tr>
<tr><td colspan="3">R/W-0h</td><td colspan="2">R/W-2h</td><td>R/W-1h</td><td>R/W-0h</td><td>R/W-1h</td></tr>
</table>

Table 26-15. MCSPI_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	0h	Reads returns 0
9-8	CLOCKACTIVITY	R/W	0h	Clocks activity during wake up mode period 0h (R/W) = OCP and Functional clocks may be switched off. 1h (R/W) = OCP clock is maintained. Functional clock may be switched-off. 2h (R/W) = Functional clock is maintained. OCP clock may be switched-off. 3h (R/W) = OCP and Functional clocks are maintained.
7-5	RESERVED	R/W	0h	Reads returns 0
4-3	SIDLEMODE	R/W	2h	Power management 0h (R/W) = If an idle request is detected, the McSPI acknowledges it unconditionally and goes in Inactive mode. Interrupt, DMA requests and wake up lines are unconditionally de-asserted and the module wakeup capability is deactivated even if the bit MCSPI_SYSCONFIG[EnaWakeUp] is set. 1h (R/W) = If an idle request is detected, the request is ignored and the module does not switch to wake up mode, and keeps on behaving normally. 2h (R/W) = If an idle request is detected, the module will switch to idle mode based on its internal activity. The wake up capability cannot be used. 3h (R/W) = If an idle request is detected, the module will switch to idle mode based on its internal activity, and the wake up capability can be used if the bit MCSPI_SYSCONFIG[EnaWakeUp] is set.
2	ENAWAKEUP	R/W	1h	WakeUp feature control 0h (R/W) = WakeUp capability is disabled 1h (R/W) = WakeUp capability is enabled
1	SOFTRESET	R/W	0h	Software reset. During reads it always returns 0. 0h (R/W) = (write) Normal mode 1h (R/W) = (write) Set this bit to 1 to trigger a module reset.The bit is automatically reset by the hardware.

Table 26-15. MCSPI_SYSCONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	AUTOIDLE	R/W	1h	Internal OCP Clock gating strategy 0h (R/W) = OCP clock is free-running 1h (R/W) = Automatic OCP clock gating strategy is applied, based on the OCP interface activity

26.4.1.6 MCSPI_SYSSTS Register (Offset = 114h) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_SYSSTS is shown in Figure 26-31 and described in Table 26-16.

Return to Summary Table.

This register provides status information about the module excluding the interrupt status information

Figure 26-31. MCSPI_SYSSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							RESETDONE
R-0h							R-0h

Table 26-16. MCSPI_SYSSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved for module specific status information. Read returns 0
0	RESETDONE	R	0h	Internal Reset Monitoring 0h (R) = Internal module reset is on-going 1h (R) = Reset completed

26.4.1.7 MCSPI_IRQSTS Register (Offset = 118h) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_IRQSTS is shown in Figure 26-32 and described in Table 26-17.

Return to Summary Table.

The interrupt status regroups all the status of the module internal events that can generate an interrupt

Figure 26-32. MCSPI_IRQSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							

23	22	21	20	19	18	17	16
RESERVED						EOW	WKS
R/W-0h						R/W1C-0h	R/W1C-0h

15	14	13	12	11	10	9	8
RESERVED	RX3_FULL	TX3_UNDERFLOW	TX3_EMPTY	RESERVED	RX2_FULL	TX2_UNDERFLOW	TX2_EMPTY
R/W-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

7	6	5	4	3	2	1	0
RESERVED	RX1_FULL	TX1_UNDERFLOW	TX1_EMPTY	RX0_OVERFLOW	RX0_FULL	TX0_UNDERFLOW	TX0_EMPTY
R/W-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

Table 26-17. MCSPI_IRQSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	Reads returns 0
17	EOW	R/W1C	0h	End of word count event when a channel is enabled using the FIFO buffer and the channel had sent the number of SPI word defined by MCSPI_XFERLEVEL[WCNT]. 0h (W) = Event status bit unchanged 0h (R) = Event false 1h (W) = Event status bit is reset 1h (R) = Event is pending
16	WKS	R/W1C	0h	Wake Up event in slave mode when an active control signal is detected on the SPIEN line programmed in the field MCSPI_CH0CONF[SPIENSLV] 0h (W) = Event status bit unchanged 0h (R) = Event false 1h (W) = Event status bit is reset 1h (R) = Event is pending
15	RESERVED	R/W	0h	Reads returns 0
14	RX3_FULL	R/W1C	0h	Receiver register is full or almost full. Only when Channel 3 is enabled 0h (W) = Event status bit unchanged 0h (R) = Event false 1h (W) = Event status bit is reset 1h (R) = Event is pending

Table 26-17. MCSPI_IRQSTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	TX3_UNDERFLOW	R/W1C	0h	Transmitter register underflow. Only when Channel 3 is enabled. The transmitter register is empty (not updated by Host or DMA with new data) before its time slot assignment. Exception: No TX_underflow event when no data has been loaded into the transmitter register since channel has been enabled. 0h (W) = Event status bit unchanged 0h (R) = Event false 1h (W) = Event status bit is reset 1h (R) = Event is pending
12	TX3_EMPTY	R/W1C	0h	Transmitter register is empty or almost empty. Note: Enabling the channel automatically rises this event. 0h (W) = Event status bit unchanged 0h (R) = Event false 1h (W) = Event status bit is reset 1h (R) = Event is pending
11	RESERVED	R/W	0h	Reads returns 0
10	RX2_FULL	R/W1C	0h	Receiver register full or almost full. Channel 2 0h (W) = Event status bit unchanged 0h (R) = Event false 1h (W) = Event status bit is reset 1h (R) = Event is pending
9	TX2_UNDERFLOW	R/W1C	0h	Transmitter register underflow. Channel 2 0h (W) = Event status bit unchanged 0h (R) = Event false 1h (W) = Event status bit is reset 1h (R) = Event is pending
8	TX2_EMPTY	R/W1C	0h	Transmitter register empty or almost empty. Channel 2 0h (W) = Event status bit unchanged 0h (R) = Event false 1h (W) = Event status bit is reset 1h (R) = Event is pending
7	RESERVED	R/W	0h	Reads returns 0
6	RX1_FULL	R/W1C	0h	Receiver register full or almost full. Channel 1 0h (W) = Event status bit unchanged 0h (R) = Event false 1h (W) = Event status bit is reset 1h (R) = Event is pending
5	TX1_UNDERFLOW	R/W1C	0h	Transmitter register underflow. Channel 1 0h (W) = Event status bit unchanged 0h (R) = Event false 1h (W) = Event status bit is reset 1h (R) = Event is pending
4	TX1_EMPTY	R/W1C	0h	Transmitter register empty or almost empty. Channel 1 0h (W) = Event status bit unchanged 0h (R) = Event false 1h (W) = Event status bit is reset 1h (R) = Event is pending

Table 26-17. MCSPI_IRQSTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	RX0_OVERFLOW	R/W1C	0h	Receiver register overflow (slave mode only). Channel 0 0h (W) = Event status bit unchanged 0h (R) = Event false 1h (W) = Event status bit is reset 1h (R) = Event is pending
2	RX0_FULL	R/W1C	0h	Receiver register full or almost full. Channel 0 0h (W) = Event status bit unchanged 0h (R) = Event false 1h (W) = Event status bit is reset 1h (R) = Event is pending
1	TX0_UNDERFLOW	R/W1C	0h	Transmitter register underflow. Channel 0 0h (W) = Event status bit unchanged 0h (R) = Event false 1h (W) = Event status bit is reset 1h (R) = Event is pending
0	TX0_EMPTY	R/W1C	0h	Transmitter register empty or almost empty. Channel 0 0h (W) = Event status bit unchanged 0h (R) = Event false 1h (W) = Event status bit is reset 1h (R) = Event is pending

26.4.1.8 MCSPI_IRQEN Register (Offset = 11Ch) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_IRQEN is shown in Figure 26-33 and described in Table 26-18.

Return to Summary Table.

This register allows to enable/disable the module internal sources of interrupt, on an event-by-event basis.

Figure 26-33. MCSPI_IRQEN Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED						EOW_EN	WKE
R/W-0h						R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RX3_FULL_EN	TX3_UNDERFLOW_EN	TX3_EMPTY_EN	RESERVED	RX2_FULL_EN	TX2_UNDERFLOW_EN	TX2_EMPTY_EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	RX1_FULL_EN	TX1_UNDERFLOW_EN	TX1_EMPTY_EN	RX0_OVERFLOW_EN	RX0_FULL_EN	TX0_UNDERFLOW_EN	TX0_EMPTY_EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 26-18. MCSPI_IRQEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	Reads return 0
17	EOW_EN	R/W	0h	End of Word count Interrupt Enable. 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled
16	WKE	R/W	0h	Wake Up event interrupt Enable in slave mode when an active control signal is detected on the SPIEN line programmed in the field MCSPI_CH0CONF[SPIENSLV] 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled
15	RESERVED	R/W	0h	Reads returns 0
14	RX3_FULL_EN	R/W	0h	Receiver register Full Interrupt Enable. Ch 3 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled
13	TX3_UNDERFLOW_EN	R/W	0h	Transmitter register Underflow Interrupt Enable. Ch 3 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled
12	TX3_EMPTY_EN	R/W	0h	Transmitter register Empty Interrupt Enable. Ch3 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled
11	RESERVED	R/W	0h	Reads return 0
10	RX2_FULL_EN	R/W	0h	Receiver register Full Interrupt Enable. Ch 2 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled

Table 26-18. MCSPI_IRQEN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	TX2_UNDERFLOW_EN	R/W	0h	Transmitter register Underflow Interrupt Enable. Ch 2 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled
8	TX2_EMPTY_EN	R/W	0h	Transmitter register Empty Interrupt Enable. Ch 2 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled
7	RESERVED	R/W	0h	Reads return 0
6	RX1_FULL_EN	R/W	0h	Receiver register Full Interrupt Enable. Ch 1 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled
5	TX1_UNDERFLOW_EN	R/W	0h	Transmitter register Underflow Interrupt Enable. Ch 1 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled
4	TX1_EMPTY_EN	R/W	0h	Transmitter register Empty Interrupt Enable. Ch 1 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled
3	RX0_OVERFLOW_EN	R/W	0h	Receiver register Overflow Interrupt Enable. Ch 0 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled
2	RX0_FULL_EN	R/W	0h	Receiver register Full Interrupt Enable. Ch 0 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled
1	TX0_UNDERFLOW_EN	R/W	0h	Transmitter register Underflow Interrupt Enable. Ch 0 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled
0	TX0_EMPTY_EN	R/W	0h	Transmitter register Empty Interrupt Enable. Ch 0 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled

26.4.1.9 MCSPI_WAKEUPEN Register (Offset = 120h) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_WAKEUPEN is shown in Figure 26-34 and described in Table 26-19.

Return to Summary Table.

The wakeup enable register allows to enable/disable the module internal sources of wakeup on event-by-event basis.

Figure 26-34. MCSPI_WAKEUPEN Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED							WKEN
R/W-0h							R/W-0h

Table 26-19. MCSPI_WAKEUPEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	Reads returns 0
0	WKEN	R/W	0h	WakeUp functionality in slave mode when an active control signal is detected on the SPIEN line programmed in the field MCSPI_CH0CONF[SPIENSLV] 0h (R/W) = The event is not allowed to wakeup the system, even if the global control bit MCSPI_SYSCONF[EnaWakeUp] is set. 1h (R/W) = The event is allowed to wakeup the system if the global control bit MCSPI_SYSCONF[EnaWakeUp] is set.

26.4.1.10 MCSPI_SYST Register (Offset = 124h) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_SYST is shown in Figure 26-35 and described in Table 26-20.

Return to Summary Table.

This register is used to check the correctness of the system interconnect either internally to peripheral bus, or externally to device IO pads, when the module is configured in system test (SYSTEST) mode.

Figure 26-35. MCSPI_SYST Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED				SSB	SPIENDIR	SPIDATDIR1	SPIDATDIR0
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
WAKD	SPICLK	SPIDAT_1	SPIDAT_0	SPIEN_3	SPIEN_2	SPIEN_1	SPIEN_0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 26-20. MCSPI_SYST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	Reads returns 0
11	SSB	R/W	0h	Set status bit 0h (R/W) = No action. Writing 0 does not clear already set status bits; This bit must be cleared prior attempting to clear a status bit of the <MCSPI_IRQSTATUS> register. 1h (R/W) = Force to 1 all status bits of MCSPI_IRQSTATUS register. Writing 1 into this bit sets to 1 all status bits contained in the <MCSPI_IRQSTATUS> register.
10	SPIENDIR	R/W	0h	Set the direction of the SPIEN [3:0] lines and SPICLK line 0h (R/W) = output (as in master mode) 1h (R/W) = input (as in slave mode)
9	SPIDATDIR1	R/W	0h	Set the direction of the SPIDAT[1] 0h (R/W) = output 1h (R/W) = input
8	SPIDATDIR0	R/W	0h	Set the direction of the SPIDAT[0] 0h (R/W) = output 1h (R/W) = input
7	WAKD	R/W	0h	SWAKEUP output (signal data value of internal signal to system). The signal is driven high or low according to the value written into this register bit. 0h (R/W) = The pin is driven low. 1h (R/W) = The pin is driven high.
6	SPICLK	R/W	0h	SPICLK line (signal data value) If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the CLKSPI line (high or low), and a write into this bit has no effect. If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the CLKSPI line is driven high or low according to the value written into this register.

Table 26-20. MCSPI_SYST Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	SPIDAT_1	R/W	0h	SPIDAT[1] line (signal data value) If MCSPI_SYST[SPIDATDIR1] = 0 (output mode direction), the SPIDAT[1] line is driven high or low according to the value written into this register. If MCSPI_SYST[SPIDATDIR1] = 1 (input mode direction), this bit returns the value on the SPIDAT[1] line (high or low), and a write into this bit has no effect.
4	SPIDAT_0	R/W	0h	SPIDAT[0] line (signal data value) If MCSPI_SYST[SPIDATDIR0] = 0 (output mode direction), the SPIDAT[0] line is driven high or low according to the value written into this register. If MCSPI_SYST[SPIDATDIR0] = 1 (input mode direction), this bit returns the value on the SPIDAT[0] line (high or low), and a write into this bit has no effect.
3	SPIEN_3	R/W	0h	SPIEN[3] line (signal data value) If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the SPIENT[3] line is driven high or low according to the value written into this register. If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the SPIEN[3] line (high or low), and a write into this bit has no effect.
2	SPIEN_2	R/W	0h	SPIEN[2] line (signal data value) If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the SPIENT[2] line is driven high or low according to the value written into this register. If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the SPIEN[2] line (high or low), and a write into this bit has no effect.
1	SPIEN_1	R/W	0h	SPIEN[1] line (signal data value) If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the SPIENT[1] line is driven high or low according to the value written into this register. If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the SPIEN[1] line (high or low), and a write into this bit has no effect.
0	SPIEN_0	R/W	0h	SPIEN[0] line (signal data value) If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the SPIENT[0] line is driven high or low according to the value written into this register. If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the SPIEN[0] line (high or low), and a write into this bit has no effect.

26.4.1.11 MCSPI_MODULCTRL Register (Offset = 128h) [reset = 4h]

Register mask: FFFFFFFFh

MCSPI_MODULCTRL is shown in Figure 26-36 and described in Table 26-21.

Return to Summary Table.

This register is dedicated to the configuration of the serial port interface.

Figure 26-36. MCSPI_MODULCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							FDAA
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
MOA	INITDLY			SYSTEM_TEST	MS	PIN34	SINGLE
R/W-0h	R/W-0h			R/W-0h	R/W-1h	R/W-0h	R/W-0h

Table 26-21. MCSPI_MODULCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	Reads returns 0
8	FDAA	R/W	0h	FIFO DMA Address 256-bit aligned This register is used when a FIFO is managed by the module and DMA connected to the controller provides only 256 bit aligned address. If this bit is set the enabled channel which uses the FIFO has its datas managed through MCSPI_DAFTX and MCSPI_DAFRX registers instead of MCSPI_TX(i) and MCSPI_RX(i) registers. 0h (R/W) = FIFO data managed by MCSPI_TX(i) and MCSPI_RX(i) registers. 1h (R/W) = FIFO data managed by MCSPI_DAFTX and MCSPI_DAFRX registers.
7	MOA	R/W	0h	Multiple word ocp access: This register can only be used when a channel is enabled using a FIFO. It allows the system to perform multiple SPI word access for a single 32-bit OCP word access. This is possible for WL < 16. 0h (R/W) = Multiple word access disabled 1h (R/W) = Multiple word access enabled with FIFO
6-4	INITDLY	R/W	0h	Initial spi delay for first transfer: This register is an option only available in SINGLE master mode, The controller waits for a delay to transmit the first spi word after channel enabled and corresponding TX register filled. This Delay is based on SPI output frequency clock, No clock output provided to the boundary and chip select is not active in 4 pin mode within this period. 0h (R/W) = No delay for first spi transfer. 1h (R/W) = 4ClkDly : The controller wait 4 spi bus clock 2h (R/W) = 8ClkDly : The controller wait 8 spi bus clock 3h (R/W) = 16ClkDly : The controller wait 16 spi bus clock 4h (R/W) = 32ClkDly : The controller wait 32 spi bus clock

Table 26-21. MCSPI_MODULCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	SYSTEM_TEST	R/W	0h	Enables the system test mode 0h (R/W) = Functional mode 1h (R/W) = System test mode (SYSTEST)
2	MS	R/W	1h	Master/ Slave 0h (R/W) = Master - The module generates the SPICLK and SPIEN[3:0] 1h (R/W) = Slave - The module receives the SPICLK and SPIEN[3:0]
1	PIN34	R/W	0h	Pin mode selection: This register is used to configure the SPI pin mode, in master or slave mode. If asserted the controller only use SIMO,SOMI and SPICLK clock pin for spi transfers. 0h (R/W) = 4PinMode : SPIEN is used as a chip select. 1h (R/W) = 3PinMode : SPEIN is not used. In this mode all related option to chip select have no meaning.
0	SINGLE	R/W	0h	Single channel / Multi Channel (master mode only) 0h (R/W) = More than one channel will be used in master mode. 1h (R/W) = Only one channel will be used in master mode. This bit must be set in Force SPIEN mode.

26.4.1.12 MCSPI_CH0CONF Register (Offset = 12Ch) [reset = 00060000h]

Register mask: FFFFFFFFh

MCSPI_CH0CONF is shown in Figure 26-37 and described in Table 26-22.

Return to Summary Table.

This register is dedicated to the configuration of the channel 0

Figure 26-37. MCSPI_CH0CONF Register

31	30	29	28	27	26	25	24
RESERVED		CLKG	FFER	FFEW	TCS0		SBPOL
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SBE	SPIENSLV		FORCE	TURBO	IS	DPE1	DPE0
R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAR	DMAW	TRM		WL			
R/W-0h	R/W-0h	R/W-0h		R/W-0h			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WL	EPOL	CLKD				POL	PHA
R/W-0h	R/W-0h	R/W-0h				R/W-0h	R/W-0h

Table 26-22. MCSPI_CH0CONF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	read returns 0
29	CLKG	R/W	0h	Clock divider granularity This register defines the granularity of channel clock divider: power of two or one clock cycle granularity. When this bit is set the register MCSPI_CHCTRL[EXTCLK] must be configured to reach a maximum of 4096 clock divider ratio. Then The clock divider ratio is a concatenation of MCSPI_CHCONF[CLKD] and MCSPI_CHCTRL[EXTCLK] values 0h (R/W) = Clock granularity of power of two 1h (R/W) = One clock cycle ganularity
28	FFER	R/W	0h	FIFO enabled for receive:Only one channel can have this bit field set. 0h (R/W) = The buffer is not used to receive data. 1h (R/W) = The buffer is used to receive data.
27	FFEW	R/W	0h	FIFO enabled for Transmit:Only one channel can have this bit field set. 0h (R/W) = The buffer is not used to transmit data. 1h (R/W) = The buffer is used to transmit data.
26-25	TCS0	R/W	0h	Chip Select Time Control This 2-bits field defines the number of interface clock cycles between CS toggling and first or last edge of SPI clock. 0h (R/W) = 0.5 clock cycle 1h (R/W) = 1.5 clock cycle 2h (R/W) = 2.5 clock cycle 3h (R/W) = 3.5 clock cycle
24	SBPOL	R/W	0h	Start bit polarity 0h (R/W) = Start bit polarity is held to 0 during SPI transfer. 1h (R/W) = Start bit polarity is held to 1 during SPI transfer.
23	SBE	R/W	0h	Start bit enable for SPI transfer 0h (R/W) = Default SPI transfer length as specified by WL bit field 1h (R/W) = Start bit D/CX added before SPI transfer polarity is defined by MCSPI_CH0CONF[SBPOL]

Table 26-22. MCSPI_CH0CONF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22-21	SPIENSLV	R/W	0h	Channel 0 only and slave mode only: SPI slave select signal detection. Reserved bits for other cases. 0h (R/W) = Detection enabled only on SPIEN[0] 1h (R/W) = Detection enabled only on SPIEN[1] 2h (R/W) = Detection enabled only on SPIEN[2] 3h (R/W) = Detection enabled only on SPIEN[3]
20	FORCE	R/W	0h	Manual SPIEN assertion to keep SPIEN active between SPI words. (single channel master mode only) 0h (R/W) = Writing 0 into this bit drives low the SPIEN line when MCSPI_CHCONF(i)[EPOL]=0, and drives it high when MCSPI_CHCONF(i)[EPOL]=1. 1h (R/W) = Writing 1 into this bit drives high the SPIEN line when MCSPI_CHCONF(i)[EPOL]=0, and drives it low when MCSPI_CHCONF(i)[EPOL]=1
19	TURBO	R/W	0h	Turbo mode 0h (R/W) = Turbo is deactivated (recommended for single SPI word transfer) 1h (R/W) = Turbo is activated to maximize the throughput for multi SPI words transfer.
18	IS	R/W	1h	Input Select 0h (R/W) = Data Line0 (SPIDAT[0]) selected for reception. 1h (R/W) = Data Line1 (SPIDAT[1]) selected for reception
17	DPE1	R/W	1h	Transmission Enable for data line 1 (SPIDATAGZEN[1]) 0h (R/W) = Data Line1 (SPIDAT[1]) selected for transmission 1h (R/W) = No transmission on Data Line1 (SPIDAT[1])
16	DPE0	R/W	0h	Transmission Enable for data line 0 (SPIDATAGZEN[0]) 0h (R/W) = Data Line0 (SPIDAT[0]) selected for transmission 1h (R/W) = No transmission on Data Line0 (SPIDAT[0])
15	DMAR	R/W	0h	DMA Read request The DMA Read request line is asserted when the channel is enabled and a new data is available in the receive register of the channel. The DMA Read request line is deasserted on read completion of the receive register of the channel. 0h (R/W) = DMA Read Request disabled 1h (R/W) = DMA Read Request enabled
14	DMAW	R/W	0h	DMA Write request. The DMA Write request line is asserted when The channel is enabled and the transmitter register of the channel is empty. The DMA Write request line is deasserted on load completion of the transmitter register of the channel. 0h (R/W) = DMA Write Request disabled 1h (R/W) = DMA Write Request enabled
13-12	TRM	R/W	0h	Transmit/Receive modes 0h (R/W) = Transmit and Receive mode 1h (R/W) = Receive only mode 2h (R/W) = Transmit only mode 3h (R/W) = Reserved

Table 26-22. MCSPI_CH0CONF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-7	WL	R/W	0h	SPI word length 0h (R/W) = Reserved 1h (R/W) = Reserved 2h (R/W) = Reserved 3h (R/W) = 4bits : The SPI word is 4-bits long 4h (R/W) = 5bits : The SPI word is 5-bits long 5h (R/W) = 6bits : The SPI word is 6-bits long 6h (R/W) = 7bits : The SPI word is 7-bits long 7h (R/W) = 8bits : The SPI word is 8-bits long 8h (R/W) = 9bits : The SPI word is 9-bits long 9h (R/W) = 10bits : The SPI word is 10-bits long Ah (R/W) = 11bits : The SPI word is 11-bits long Bh (R/W) = 12bits : The SPI word is 12-bits long Ch (R/W) = 13bits : The SPI word is 13-bits long Dh (R/W) = 14bits : The SPI word is 14-bits long Eh (R/W) = 15bits : The SPI word is 15-bits long Fh (R/W) = 16bits : The SPI word is 16-bits long 10h (R/W) = 17bits : The SPI word is 17-bits long 11h (R/W) = 18bits : The SPI word is 18-bits long 12h (R/W) = 19bits : The SPI word is 19-bits long 13h (R/W) = 20bits : The SPI word is 20-bits long 14h (R/W) = 21bits : The SPI word is 21-bits long 15h (R/W) = 22bits : The SPI word is 22-bits long 16h (R/W) = 23bits : The SPI word is 23-bits long 17h (R/W) = 24bits : The SPI word is 24-bits long 18h (R/W) = 25bits : The SPI word is 25-bits long 19h (R/W) = 26bits : The SPI word is 26-bits long 1Ah (R/W) = 27bits : The SPI word is 27-bits long 1Bh (R/W) = 28bits : The SPI word is 28-bits long 1Ch (R/W) = 29bits : The SPI word is 29-bits long 1Dh (R/W) = 30bits : The SPI word is 30-bits long 1Eh (R/W) = 31bits : The SPI word is 31-bits long 1Fh (R/W) = 32bits : The SPI word is 32-bits long
6	EPOL	R/W	0h	SPIEN polarity 0h (R/W) = SPIEN is held high during the active state. 1h (R/W) = SPIEN is held low during the active state.

Table 26-22. MCSPI_CH0CONF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-2	CLKD	R/W	0h	Frequency divider for SPICLK. (only when the module is a Master SPI device). A programmable clock divider divides the SPI reference clock (CLKSPIREF) with a 4-bit value, and results in a new clock SPICLK available to shift-in and shift-out data. By default the clock divider ratio has a power of two granularity when MCSPI_CHCONF[CLKG] is cleared, Otherwise this register is the 4 LSB bit of a 12-bit register concatenated with clock divider extension MCSPI_CHCTRL[EXTCLK] register.The value description below defines the clock ratio when MCSPI_CHCONF[CLKG] is set to 0. 0h (R/W) = 1 1h (R/W) = 2 2h (R/W) = 4 3h (R/W) = 8 4h (R/W) = 16 5h (R/W) = 32 6h (R/W) = 64 7h (R/W) = 128 8h (R/W) = 256 9h (R/W) = 512 Ah (R/W) = 1024 Bh (R/W) = 2048 Ch (R/W) = 4096 Dh (R/W) = 8192 Eh (R/W) = 16384 Fh (R/W) = 32768
1	POL	R/W	0h	SPICLK polarity 0h (R/W) = SPICLK is held high during the active state 1h (R/W) = SPICLK is held low during the active state
0	PHA	R/W	0h	SPICLK phase 0h (R/W) = Data are latched on odd numbered edges of SPICLK. 1h (R/W) = Data are latched on even numbered edges of SPICLK.

26.4.1.13 MCSPI_CH0STAT Register (Offset = 130h) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_CH0STAT is shown in Figure 26-38 and described in Table 26-23.

Return to Summary Table.

This register provides status information about transmitter and receiver registers of channel 0

Figure 26-38. MCSPI_CH0STAT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>RESERVED</td><td>RXFFF</td><td>RXFFE</td><td>TXFFF</td><td>TXFFE</td><td>EOT</td><td>TXS</td><td>RXS</td></tr>
<tr><td>R-0h</td><td>R-0h</td><td>R-0h</td><td>R-0h</td><td>R-0h</td><td>R-0h</td><td>R-0h</td><td>R-0h</td></tr>
</table>

Table 26-23. MCSPI_CH0STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Read returns 0
6	RXFFF	R	0h	Channel "i" FIFO Receive Buffer Full Status 0h (R) = FIFO Receive Buffer is not full 1h (R) = FIFO Receive Buffer is full
5	RXFFE	R	0h	Channel "i" FIFO Receive Buffer Empty Status 0h (R) = FIFO Receive Buffer is not empty 1h (R) = FIFO Receive Buffer is empty
4	TXFFF	R	0h	Channel "i" FIFO Transmit Buffer Full Status 0h (R) = FIFO Transmit Buffer is not full 1h (R) = FIFO Transmit Buffer is full
3	TXFFE	R	0h	Channel "i" FIFO Transmit Buffer Empty Status 0h (R) = FIFO Transmit Buffer is not empty 1h (R) = FIFO Transmit Buffer is empty
2	EOT	R	0h	Channel "i" End of transfer Status. The definitions of beginning and end of transfer vary with master versus slave and the transfer format (Transmit/Receive modes, Turbo mode). See dedicated chapters for details. 0h (R) = This flag is automatically cleared when the shift register is loaded with the data from the transmitter register (beginning of transfer). 1h (R) = This flag is automatically set to one at the end of an SPI transfer.
1	TXS	R	0h	Channel "i" Transmitter Register Status 0h (R) = Register is full 1h (R) = Register is empty
0	RXS	R	0h	Channel "i" Receiver Register Status 0h (R) = Register is empty 1h (R) = Register is full

26.4.1.14 MCSPI_CH0CTRL Register (Offset = 134h) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_CH0CTRL is shown in Figure 26-39 and described in Table 26-24.

Return to Summary Table.

This register is dedicated to enable the channel 0

Figure 26-39. MCSPI_CH0CTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTCLK								RESERVED							EN
R/W-0h								R/W-0h							R/W-0h

Table 26-24. MCSPI_CH0CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	Read returns 0
15-8	EXTCLK	R/W	0h	Clock ratio extension: This register is used to concatenate with MCSPI_CHCONF[CLKD] register for clock ratio only when granularity is one clock cycle (MCSPI_CHCONF[CLKG] set to 1). Then the max value reached is 4096 clock divider ratio. 0h (R/W) = Clock ratio is CLKD + 1 1h (R/W) = Clock ratio is CLKD + 1 + 16 FFh (R/W) = Clock ratio is CLKD + 1 + 4080
7-1	RESERVED	R/W	0h	Read returns 0
0	EN	R/W	0h	Channel Enable 0h (R/W) = Channel "i" is not active 1h (R/W) = Channel "i" is active

26.4.1.15 MCSPI_TX0 Register (Offset = 138h) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_TX0 is shown in Figure 26-40 and described in Table 26-25.

Return to Summary Table.

The McSPI channel FIFO transmit buffer register (MCSPI_TXx) contains a single McSPI word to transmit through the serial link, whatever the SPI word length is. Little endian host access SPI 8 bit word on 0; big endian host accesses on 3h. SPI words are transferred with MSB first. Refer to section 26.3.10 "Access to Data Registers" for more information.

Figure 26-40. MCSPI_TX0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDATA																															
R/W-0h																															

Table 26-25. MCSPI_TX0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TDATA	R/W	0h	Channel 0 Data to transmit

26.4.1.16 MCSPI_RX0 Register (Offset = 13Ch) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_RX0 is shown in Figure 26-41 and described in Table 26-26.

Return to Summary Table.

The McSPI channel FIFO receive buffer register (MCSPI_RXx) contains a single McSPI word received through the serial link, whatever the SPI word length is. Little endian host access SPI 8 bit word on 0h; big endian host accesses on 3h. Refer to section 26.3.10 "Access to Data Registers" for more information.

Figure 26-41. MCSPI_RX0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDATA																															
R-0h																															

Table 26-26. MCSPI_RX0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RDATA	R	0h	Channel 0 Received Data

26.4.1.17 MCSPI_CH1CONF Register (Offset = 140h) [reset = 00060000h]

Register mask: FFFFFFFFh

MCSPI_CH1CONF is shown in Figure 26-42 and described in Table 26-27.

Return to Summary Table.

This register is dedicated to the configuration of the channel.

Figure 26-42. MCSPI_CH1CONF Register

31	30	29	28	27	26	25	24
RESERVED		CLKG	FFER	FFEW	TCS1		SBPOL
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h

23	22	21	20	19	18	17	16
SBE	RESERVED		FORCE	TURBO	IS	DPE1	DPE0
R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-1h	R/W-1h	R/W-0h

15	14	13	12	11	10	9	8
DMAR	DMAW	TRM		WL			
R/W-0h	R/W-0h	R/W-0h		R/W-0h			

7	6	5	4	3	2	1	0
WL	EPOL	CLKD				POL	PHA
R/W-0h	R/W-0h	R/W-0h				R/W-0h	R/W-0h

Table 26-27. MCSPI_CH1CONF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	read returns 0
29	CLKG	R/W	0h	Clock divider granularity This register defines the granularity of channel clock divider: power of two or one clock cycle granularity. When this bit is set the register MCSPI_CHCTRL[EXTCLK] must be configured to reach a maximum of 4096 clock divider ratio. Then The clock divider ratio is a concatenation of MCSPI_CHCONF[CLKD] and MCSPI_CHCTRL[EXTCLK] values 0h (R/W) = Clock granularity of power of two 1h (R/W) = One clock cycle ganularity
28	FFER	R/W	0h	FIFO enabled for receive:Only one channel can have this bit field set. 0h (R/W) = The buffer is not used to receive data. 1h (R/W) = The buffer is used to receive data.
27	FFEW	R/W	0h	FIFO enabled for Transmit:Only one channel can have this bit field set. 0h (R/W) = The buffer is not used to transmit data. 1h (R/W) = The buffer is used to transmit data.
26-25	TCS1	R/W	0h	Chip Select Time Control This 2-bits field defines the number of interface clock cycles between CS toggling and first or last edge of SPI clock. 0h (R/W) = 0.5 clock cycle 1h (R/W) = 1.5 clock cycle 2h (R/W) = 2.5 clock cycle 3h (R/W) = 3.5 clock cycle
24	SBPOL	R/W	0h	Start bit polarity 0h (R/W) = Start bit polarity is held to 0 during SPI transfer. 1h (R/W) = Start bit polarity is held to 1 during SPI transfer.
23	SBE	R/W	0h	Start bit enable for SPI transfer 0h (R/W) = Default SPI transfer length as specified by WL bit field 1h (R/W) = Start bit D/CX added before SPI transfer polarity is defined by MCSPI_CH1CONF[SBPOL]

Table 26-27. MCSPI_CH1CONF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22-21	RESERVED	R/W	0h	read returns 0
20	FORCE	R/W	0h	Manual SPIEN assertion to keep SPIEN active between SPI words. (single channel master mode only) 0h (R/W) = Writing 0 into this bit drives low the SPIEN line when MCSPI_CHCONF(i)[EPOL]=0, and drives it high when MCSPI_CHCONF(i)[EPOL]=1. 1h (R/W) = Writing 1 into this bit drives high the SPIEN line when MCSPI_CHCONF(i)[EPOL]=0, and drives it low when MCSPI_CHCONF(i)[EPOL]=1
19	TURBO	R/W	0h	Turbo mode 0h (R/W) = Turbo is deactivated (recommended for single SPI word transfer) 1h (R/W) = Turbo is activated to maximize the throughput for multi SPI words transfer.
18	IS	R/W	1h	Input Select 0h (R/W) = Data Line0 (SPIDAT[0]) selected for reception. 1h (R/W) = Data Line1 (SPIDAT[1]) selected for reception
17	DPE1	R/W	1h	Transmission Enable for data line 1 (SPIDATAGZEN[1]) 0h (R/W) = Data Line1 (SPIDAT[1]) selected for transmission 1h (R/W) = No transmission on Data Line1 (SPIDAT[1])
16	DPE0	R/W	0h	Transmission Enable for data line 0 (SPIDATAGZEN[0]) 0h (R/W) = Data Line0 (SPIDAT[0]) selected for transmission 1h (R/W) = No transmission on Data Line0 (SPIDAT[0])
15	DMAR	R/W	0h	DMA Read request The DMA Read request line is asserted when the channel is enabled and a new data is available in the receive register of the channel. The DMA Read request line is deasserted on read completion of the receive register of the channel. 0h (R/W) = DMA Read Request disabled 1h (R/W) = DMA Read Request enabled
14	DMAW	R/W	0h	DMA Write request. The DMA Write request line is asserted when The channel is enabled and the transmitter register of the channel is empty. The DMA Write request line is deasserted on load completion of the transmitter register of the channel. 0h (R/W) = DMA Write Request disabled 1h (R/W) = DMA Write Request enabled
13-12	TRM	R/W	0h	Transmit/Receive modes 0h (R/W) = Transmit and Receive mode 1h (R/W) = Receive only mode 2h (R/W) = Transmit only mode 3h (R/W) = Reserved

Table 26-27. MCSPI_CH1CONF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-7	WL	R/W	0h	SPI word length 0h (R/W) = Reserved 1h (R/W) = Reserved 2h (R/W) = Reserved 3h (R/W) = 4bits : The SPI word is 4-bits long 4h (R/W) = 5bits : The SPI word is 5-bits long 5h (R/W) = 6bits : The SPI word is 6-bits long 6h (R/W) = 7bits : The SPI word is 7-bits long 7h (R/W) = 8bits : The SPI word is 8-bits long 8h (R/W) = 9bits : The SPI word is 9-bits long 9h (R/W) = 10bits : The SPI word is 10-bits long Ah (R/W) = 11bits : The SPI word is 11-bits long Bh (R/W) = 12bits : The SPI word is 12-bits long Ch (R/W) = 13bits : The SPI word is 13-bits long Dh (R/W) = 14bits : The SPI word is 14-bits long Eh (R/W) = 15bits : The SPI word is 15-bits long Fh (R/W) = 16bits : The SPI word is 16-bits long 10h (R/W) = 17bits : The SPI word is 17-bits long 11h (R/W) = 18bits : The SPI word is 18-bits long 12h (R/W) = 19bits : The SPI word is 19-bits long 13h (R/W) = 20bits : The SPI word is 20-bits long 14h (R/W) = 21bits : The SPI word is 21-bits long 15h (R/W) = 22bits : The SPI word is 22-bits long 16h (R/W) = 23bits : The SPI word is 23-bits long 17h (R/W) = 24bits : The SPI word is 24-bits long 18h (R/W) = 25bits : The SPI word is 25-bits long 19h (R/W) = 26bits : The SPI word is 26-bits long 1Ah (R/W) = 27bits : The SPI word is 27-bits long 1Bh (R/W) = 28bits : The SPI word is 28-bits long 1Ch (R/W) = 29bits : The SPI word is 29-bits long 1Dh (R/W) = 30bits : The SPI word is 30-bits long 1Eh (R/W) = 31bits : The SPI word is 31-bits long 1Fh (R/W) = 32bits : The SPI word is 32-bits long
6	EPOL	R/W	0h	SPIEN polarity 0h (R/W) = SPIEN is held high during the active state. 1h (R/W) = SPIEN is held low during the active state.

Table 26-27. MCSPI_CH1CONF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-2	CLKD	R/W	0h	Frequency divider for SPICLK. (only when the module is a Master SPI device). A programmable clock divider divides the SPI reference clock (CLKSPIREF) with a 4-bit value, and results in a new clock SPICLK available to shift-in and shift-out data. By default the clock divider ratio has a power of two granularity when MCSPI_CHCONF[CLKG] is cleared, Otherwise this register is the 4 LSB bit of a 12-bit register concatenated with clock divider extension MCSPI_CHCTRL[EXTCLK] register.The value description below defines the clock ratio when MCSPI_CHCONF[CLKG] is set to 0. 0h (R/W) = 1 1h (R/W) = 2 2h (R/W) = 4 3h (R/W) = 8 4h (R/W) = 16 5h (R/W) = 32 6h (R/W) = 64 7h (R/W) = 128 8h (R/W) = 256 9h (R/W) = 512 Ah (R/W) = 1024 Bh (R/W) = 2048 Ch (R/W) = 4096 Dh (R/W) = 8192 Eh (R/W) = 16384 Fh (R/W) = 32768
1	POL	R/W	0h	SPICLK polarity 0h (R/W) = SPICLK is held high during the active state 1h (R/W) = SPICLK is held low during the active state
0	PHA	R/W	0h	SPICLK phase 0h (R/W) = Data are latched on odd numbered edges of SPICLK. 1h (R/W) = Data are latched on even numbered edges of SPICLK.

26.4.1.18 MCSPI_CH1STAT Register (Offset = 144h) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_CH1STAT is shown in Figure 26-43 and described in Table 26-28.

Return to Summary Table.

This register provides status information about transmitter and receiver registers of channel 1

Figure 26-43. MCSPI_CH1STAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	RXFFF	RXFFE	TXFFF	TXFFE	EOT	TXS	RXS
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 26-28. MCSPI_CH1STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Read returns 0
6	RXFFF	R	0h	Channel "i" FIFO Receive Buffer Full Status 0h (R) = FIFO Receive Buffer is not full 1h (R) = FIFO Receive Buffer is full
5	RXFFE	R	0h	Channel "i" FIFO Receive Buffer Empty Status 0h (R) = FIFO Receive Buffer is not empty 1h (R) = FIFO Receive Buffer is empty
4	TXFFF	R	0h	Channel "i" FIFO Transmit Buffer Full Status 0h (R) = FIFO Transmit Buffer is not full 1h (R) = FIFO Transmit Buffer is full
3	TXFFE	R	0h	Channel "i" FIFO Transmit Buffer Empty Status 0h (R) = FIFO Transmit Buffer is not empty 1h (R) = FIFO Transmit Buffer is empty
2	EOT	R	0h	Channel "i" End of transfer Status. The definitions of beginning and end of transfer vary with master versus slave and the transfer format (Transmit/Receive modes, Turbo mode). See dedicated chapters for details. 0h (R) = This flag is automatically cleared when the shift register is loaded with the data from the transmitter register (beginning of transfer). 1h (R) = This flag is automatically set to one at the end of an SPI transfer.
1	TXS	R	0h	Channel "i" Transmitter Register Status 0h (R) = Register is full 1h (R) = Register is empty
0	RXS	R	0h	Channel "i" Receiver Register Status 0h (R) = Register is empty 1h (R) = Register is full

26.4.1.19 MCSPI_CH1CTRL Register (Offset = 148h) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_CH1CTRL is shown in Figure 26-44 and described in Table 26-29.

Return to Summary Table.

This register is dedicated to enable the channel 1

Figure 26-44. MCSPI_CH1CTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EXTCLK								RESERVED							EN
R/W-0h								R/W-0h							R/W-0h

Table 26-29. MCSPI_CH1CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	Read returns 0
15-8	EXTCLK	R/W	0h	Clock ratio extension: This register is used to concatenate with MCSPI_CHCONF[CLKD] register for clock ratio only when granularity is one clock cycle (MCSPI_CHCONF[CLKG] set to 1). Then the max value reached is 4096 clock divider ratio. 0h (R/W) = Clock ratio is CLKD + 1 1h (R/W) = Clock ratio is CLKD + 1 + 16 FFh (R/W) = Clock ratio is CLKD + 1 + 4080
7-1	RESERVED	R/W	0h	Read returns 0
0	EN	R/W	0h	Channel Enable 0h (R/W) = Channel "i" is not active 1h (R/W) = Channel "i" is active

26.4.1.20 MCSPI_TX1 Register (Offset = 14Ch) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_TX1 is shown in Figure 26-45 and described in Table 26-30.

Return to Summary Table.

The McSPI channel FIFO transmit buffer register (MCSPI_TXx) contains a single McSPI word to transmit through the serial link, whatever the SPI word length is. Little endian host access SPI 8 bit word on 0; big endian host accesses on 3h. SPI words are transferred with MSB first. Refer to section 26.3.10 "Access to Data Registers" for more information.

Figure 26-45. MCSPI_TX1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDATA																															
R/W-0h																															

Table 26-30. MCSPI_TX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TDATA	R/W	0h	Channel 1 Data to transmit

26.4.1.21 MCSPI_RX1 Register (Offset = 150h) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_RX1 is shown in Figure 26-46 and described in Table 26-31.

Return to Summary Table.

The McSPI channel FIFO receive buffer register (MCSPI_RXx) contains a single McSPI word received through the serial link, whatever the SPI word length is. Little endian host access SPI 8 bit word on 0h; big endian host accesses on 3h. Refer to section 26.3.10 "Access to Data Registers" for more information.

Figure 26-46. MCSPI_RX1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">RDATA</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 26-31. MCSPI_RX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RDATA	R	0h	Channel 1 Received Data

26.4.1.22 MCSPI_CH2CONF Register (Offset = 154h) [reset = 00060000h]

Register mask: FFFFFFFFh

MCSPI_CH2CONF is shown in Figure 26-47 and described in Table 26-32.

Return to Summary Table.

This register is dedicated to the configuration of the channel 2

Figure 26-47. MCSPI_CH2CONF Register

31	30	29	28	27	26	25	24
RESERVED		CLKG	FFER	FFEW	TCS2		SBPOL
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SBE	RESERVED		FORCE	TURBO	IS	DPE1	DPE0
R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAR	DMAW	TRM		WL			
R/W-0h	R/W-0h	R/W-0h		R/W-0h			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WL	EPOL	CLKD				POL	PHA
R/W-0h	R/W-0h	R/W-0h				R/W-0h	R/W-0h

Table 26-32. MCSPI_CH2CONF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	read returns 0
29	CLKG	R/W	0h	Clock divider granularity This register defines the granularity of channel clock divider: power of two or one clock cycle granularity. When this bit is set the register MCSPI_CHCTRL[EXTCLK] must be configured to reach a maximum of 4096 clock divider ratio. Then The clock divider ratio is a concatenation of MCSPI_CHCONF[CLKD] and MCSPI_CHCTRL[EXTCLK] values 0h (R/W) = Clock granularity of power of two 1h (R/W) = One clock cycle ganularity
28	FFER	R/W	0h	FIFO enabled for receive:Only one channel can have this bit field set. 0h (R/W) = The buffer is not used to receive data. 1h (R/W) = The buffer is used to receive data.
27	FFEW	R/W	0h	FIFO enabled for Transmit:Only one channel can have this bit field set. 0h (R/W) = The buffer is not used to transmit data. 1h (R/W) = The buffer is used to transmit data.
26-25	TCS2	R/W	0h	Chip Select Time Control This 2-bits field defines the number of interface clock cycles between CS toggling and first or last edge of SPI clock. 0h (R/W) = 0.5 clock cycle 1h (R/W) = 1.5 clock cycle 2h (R/W) = 2.5 clock cycle 3h (R/W) = 3.5 clock cycle
24	SBPOL	R/W	0h	Start bit polarity 0h (R/W) = Start bit polarity is held to 0 during SPI transfer. 1h (R/W) = Start bit polarity is held to 1 during SPI transfer.
23	SBE	R/W	0h	Start bit enable for SPI transfer 0h (R/W) = Default SPI transfer length as specified by WL bit field 1h (R/W) = Start bit D/CX added before SPI transfer polarity is defined by MCSPI_CH2CONF[SBPOL]

Table 26-32. MCSPI_CH2CONF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22-21	RESERVED	R/W	0h	read returns 0
20	FORCE	R/W	0h	Manual SPIEN assertion to keep SPIEN active between SPI words. (single channel master mode only) 0h (R/W) = Writing 0 into this bit drives low the SPIEN line when MCSPI_CHCONF(i)[EPOL]=0, and drives it high when MCSPI_CHCONF(i)[EPOL]=1. 1h (R/W) = Writing 1 into this bit drives high the SPIEN line when MCSPI_CHCONF(i)[EPOL]=0, and drives it low when MCSPI_CHCONF(i)[EPOL]=1
19	TURBO	R/W	0h	Turbo mode 0h (R/W) = Turbo is deactivated (recommended for single SPI word transfer) 1h (R/W) = Turbo is activated to maximize the throughput for multi SPI words transfer.
18	IS	R/W	1h	Input Select 0h (R/W) = Data Line0 (SPIDAT[0]) selected for reception. 1h (R/W) = Data Line1 (SPIDAT[1]) selected for reception
17	DPE1	R/W	1h	Transmission Enable for data line 1 (SPIDATAGZEN[1]) 0h (R/W) = Data Line1 (SPIDAT[1]) selected for transmission 1h (R/W) = No transmission on Data Line1 (SPIDAT[1])
16	DPE0	R/W	0h	Transmission Enable for data line 0 (SPIDATAGZEN[0]) 0h (R/W) = Data Line0 (SPIDAT[0]) selected for transmission 1h (R/W) = No transmission on Data Line0 (SPIDAT[0])
15	DMAR	R/W	0h	DMA Read request The DMA Read request line is asserted when the channel is enabled and a new data is available in the receive register of the channel. The DMA Read request line is deasserted on read completion of the receive register of the channel. 0h (R/W) = DMA Read Request disabled 1h (R/W) = DMA Read Request enabled
14	DMAW	R/W	0h	DMA Write request. The DMA Write request line is asserted when The channel is enabled and the transmitter register of the channel is empty. The DMA Write request line is deasserted on load completion of the transmitter register of the channel. 0h (R/W) = DMA Write Request disabled 1h (R/W) = DMA Write Request enabled
13-12	TRM	R/W	0h	Transmit/Receive modes 0h (R/W) = Transmit and Receive mode 1h (R/W) = Receive only mode 2h (R/W) = Transmit only mode 3h (R/W) = Reserved

Table 26-32. MCSPI_CH2CONF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-7	WL	R/W	0h	SPI word length 0h (R/W) = Reserved 1h (R/W) = Reserved 2h (R/W) = Reserved 3h (R/W) = 4bits : The SPI word is 4-bits long 4h (R/W) = 5bits : The SPI word is 5-bits long 5h (R/W) = 6bits : The SPI word is 6-bits long 6h (R/W) = 7bits : The SPI word is 7-bits long 7h (R/W) = 8bits : The SPI word is 8-bits long 8h (R/W) = 9bits : The SPI word is 9-bits long 9h (R/W) = 10bits : The SPI word is 10-bits long Ah (R/W) = 11bits : The SPI word is 11-bits long Bh (R/W) = 12bits : The SPI word is 12-bits long Ch (R/W) = 13bits : The SPI word is 13-bits long Dh (R/W) = 14bits : The SPI word is 14-bits long Eh (R/W) = 15bits : The SPI word is 15-bits long Fh (R/W) = 16bits : The SPI word is 16-bits long 10h (R/W) = 17bits : The SPI word is 17-bits long 11h (R/W) = 18bits : The SPI word is 18-bits long 12h (R/W) = 19bits : The SPI word is 19-bits long 13h (R/W) = 20bits : The SPI word is 20-bits long 14h (R/W) = 21bits : The SPI word is 21-bits long 15h (R/W) = 22bits : The SPI word is 22-bits long 16h (R/W) = 23bits : The SPI word is 23-bits long 17h (R/W) = 24bits : The SPI word is 24-bits long 18h (R/W) = 25bits : The SPI word is 25-bits long 19h (R/W) = 26bits : The SPI word is 26-bits long 1Ah (R/W) = 27bits : The SPI word is 27-bits long 1Bh (R/W) = 28bits : The SPI word is 28-bits long 1Ch (R/W) = 29bits : The SPI word is 29-bits long 1Dh (R/W) = 30bits : The SPI word is 30-bits long 1Eh (R/W) = 31bits : The SPI word is 31-bits long 1Fh (R/W) = 32bits : The SPI word is 32-bits long
6	EPOL	R/W	0h	SPIEN polarity 0h (R/W) = SPIEN is held high during the active state. 1h (R/W) = SPIEN is held low during the active state.

Table 26-32. MCSPI_CH2CONF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-2	CLKD	R/W	0h	Frequency divider for SPICLK. (only when the module is a Master SPI device). A programmable clock divider divides the SPI reference clock (CLKSPIREF) with a 4-bit value, and results in a new clock SPICLK available to shift-in and shift-out data. By default the clock divider ratio has a power of two granularity when MCSPI_CHCONF[CLKG] is cleared, Otherwise this register is the 4 LSB bit of a 12-bit register concatenated with clock divider extension MCSPI_CHCTRL[EXTCLK] register.The value description below defines the clock ratio when MCSPI_CHCONF[CLKG] is set to 0. 0h (R/W) = 1 1h (R/W) = 2 2h (R/W) = 4 3h (R/W) = 8 4h (R/W) = 16 5h (R/W) = 32 6h (R/W) = 64 7h (R/W) = 128 8h (R/W) = 256 9h (R/W) = 512 Ah (R/W) = 1024 Bh (R/W) = 2048 Ch (R/W) = 4096 Dh (R/W) = 8192 Eh (R/W) = 16384 Fh (R/W) = 32768
1	POL	R/W	0h	SPICLK polarity 0h (R/W) = SPICLK is held high during the active state 1h (R/W) = SPICLK is held low during the active state
0	PHA	R/W	0h	SPICLK phase 0h (R/W) = Data are latched on odd numbered edges of SPICLK. 1h (R/W) = Data are latched on even numbered edges of SPICLK.

26.4.1.23 MCSPI_CH2STAT Register (Offset = 158h) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_CH2STAT is shown in Figure 26-48 and described in Table 26-33.

Return to Summary Table.

This register provides status information about transmitter and receiver registers of channel 2

Figure 26-48. MCSPI_CH2STAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	RXFFF	RXFFE	TXFFF	TXFFE	EOT	TXS	RXS
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 26-33. MCSPI_CH2STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Read returns 0
6	RXFFF	R	0h	Channel "i" FIFO Receive Buffer Full Status 0h (R) = FIFO Receive Buffer is not full 1h (R) = FIFO Receive Buffer is full
5	RXFFE	R	0h	Channel "i" FIFO Receive Buffer Empty Status 0h (R) = FIFO Receive Buffer is not empty 1h (R) = FIFO Receive Buffer is empty
4	TXFFF	R	0h	Channel "i" FIFO Transmit Buffer Full Status 0h (R) = FIFO Transmit Buffer is not full 1h (R) = FIFO Transmit Buffer is full
3	TXFFE	R	0h	Channel "i" FIFO Transmit Buffer Empty Status 0h (R) = FIFO Transmit Buffer is not empty 1h (R) = FIFO Transmit Buffer is empty
2	EOT	R	0h	Channel "i" End of transfer Status. The definitions of beginning and end of transfer vary with master versus slave and the transfer format (Transmit/Receive modes, Turbo mode). See dedicated chapters for details. 0h (R) = This flag is automatically cleared when the shift register is loaded with the data from the transmitter register (beginning of transfer). 1h (R) = This flag is automatically set to one at the end of an SPI transfer.
1	TXS	R	0h	Channel "i" Transmitter Register Status 0h (R) = Register is full 1h (R) = Register is empty
0	RXS	R	0h	Channel "i" Receiver Register Status 0h (R) = Register is empty 1h (R) = Register is full

26.4.1.24 MCSPI_CH2CTRL Register (Offset = 15Ch) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_CH2CTRL is shown in Figure 26-49 and described in Table 26-34.

Return to Summary Table.

This register is dedicated to enable the channel 2

Figure 26-49. MCSPI_CH2CTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTCLK								RESERVED							EN
R/W-0h								R/W-0h							R/W-0h

Table 26-34. MCSPI_CH2CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	Read returns 0
15-8	EXTCLK	R/W	0h	Clock ratio extension: This register is used to concatenate with MCSPI_CHCONF[CLKD] register for clock ratio only when granularity is one clock cycle (MCSPI_CHCONF[CLKG] set to 1). Then the max value reached is 4096 clock divider ratio. 0h (R/W) = Clock ratio is CLKD + 1 1h (R/W) = Clock ratio is CLKD + 1 + 16 FFh (R/W) = Clock ratio is CLKD + 1 + 4080
7-1	RESERVED	R/W	0h	Read returns 0
0	EN	R/W	0h	Channel Enable 0h (R/W) = Channel "i" is not active 1h (R/W) = Channel "i" is active

26.4.1.25 MCSPI_TX2 Register (Offset = 160h) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_TX2 is shown in Figure 26-50 and described in Table 26-35.

Return to Summary Table.

The McSPI channel FIFO transmit buffer register (MCSPI_TXx) contains a single McSPI word to transmit through the serial link, whatever the SPI word length is. Little endian host access SPI 8 bit word on 0; big endian host accesses on 3h. SPI words are transferred with MSB first. Refer to section 26.3.10 "Access to Data Registers" for more information.

Figure 26-50. MCSPI_TX2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDATA																															
R/W-0h																															

Table 26-35. MCSPI_TX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TDATA	R/W	0h	Channel 2 Data to transmit

26.4.1.26 MCSPI_RX2 Register (Offset = 164h) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_RX2 is shown in Figure 26-51 and described in Table 26-36.

Return to Summary Table.

The McSPI channel FIFO receive buffer register (MCSPI_RXx) contains a single McSPI word received through the serial link, whatever the SPI word length is. Little endian host access SPI 8 bit word on 0h; big endian host accesses on 3h. Refer to section 26.3.10 "Access to Data Registers" for more information.

Figure 26-51. MCSPI_RX2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDATA																															
R-0h																															

Table 26-36. MCSPI_RX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RDATA	R	0h	Channel 2 Received Data

26.4.1.27 MCSPI_CH3CONF Register (Offset = 168h) [reset = 00060000h]

Register mask: FFFFFFFFh

MCSPI_CH3CONF is shown in Figure 26-52 and described in Table 26-37.

Return to Summary Table.

This register is dedicated to the configuration of the channel 3

Figure 26-52. MCSPI_CH3CONF Register

31	30	29	28	27	26	25	24
RESERVED		CLKG	FFER	FFEW	TCS3		SBPOL
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h
23	22	21	20	19	18	17	16
SBE	RESERVED		FORCE	TURBO	IS	DPE1	DPE0
R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-1h	R/W-1h	R/W-0h
15	14	13	12	11	10	9	8
DMAR	DMAW	TRM		WL			
R/W-0h	R/W-0h	R/W-0h		R/W-0h			
7	6	5	4	3	2	1	0
WL	EPOL	CLKD				POL	PHA
R/W-0h	R/W-0h	R/W-0h				R/W-0h	R/W-0h

Table 26-37. MCSPI_CH3CONF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	read returns 0
29	CLKG	R/W	0h	Clock divider granularity This register defines the granularity of channel clock divider: power of two or one clock cycle granularity. When this bit is set the register MCSPI_CHCTRL[EXTCLK] must be configured to reach a maximum of 4096 clock divider ratio. Then The clock divider ratio is a concatenation of MCSPI_CHCONF[CLKD] and MCSPI_CHCTRL[EXTCLK] values 0h (R/W) = Clock granularity of power of two 1h (R/W) = One clock cycle ganularity
28	FFER	R/W	0h	FIFO enabled for receive:Only one channel can have this bit field set. 0h (R/W) = The buffer is not used to receive data. 1h (R/W) = The buffer is used to receive data.
27	FFEW	R/W	0h	FIFO enabled for Transmit:Only one channel can have this bit field set. 0h (R/W) = The buffer is not used to transmit data. 1h (R/W) = The buffer is used to transmit data.
26-25	TCS3	R/W	0h	Chip Select Time Control This 2-bits field defines the number of interface clock cycles between CS toggling and first or last edge of SPI clock. 0h (R/W) = 0.5 clock cycle 1h (R/W) = 1.5 clock cycle 2h (R/W) = 2.5 clock cycle 3h (R/W) = 3.5 clock cycle
24	SBPOL	R/W	0h	Start bit polarity 0h (R/W) = Start bit polarity is held to 0 during SPI transfer. 1h (R/W) = Start bit polarity is held to 1 during SPI transfer.
23	SBE	R/W	0h	Start bit enable for SPI transfer 0h (R/W) = Default SPI transfer length as specified by WL bit field 1h (R/W) = Start bit D/CX added before SPI transfer polarity is defined by MCSPI_CH3CONF[SBPOL]

Table 26-37. MCSPI_CH3CONF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22-21	RESERVED	R/W	0h	read returns 0
20	FORCE	R/W	0h	Manual SPIEN assertion to keep SPIEN active between SPI words. (single channel master mode only) 0h (R/W) = Writing 0 into this bit drives low the SPIEN line when MCSPI_CHCONF(i)[EPOL]=0, and drives it high when MCSPI_CHCONF(i)[EPOL]=1. 1h (R/W) = Writing 1 into this bit drives high the SPIEN line when MCSPI_CHCONF(i)[EPOL]=0, and drives it low when MCSPI_CHCONF(i)[EPOL]=1
19	TURBO	R/W	0h	Turbo mode 0h (R/W) = Turbo is deactivated (recommended for single SPI word transfer) 1h (R/W) = Turbo is activated to maximize the throughput for multi SPI words transfer.
18	IS	R/W	1h	Input Select 0h (R/W) = Data Line0 (SPIDAT[0]) selected for reception. 1h (R/W) = Data Line1 (SPIDAT[1]) selected for reception
17	DPE1	R/W	1h	Transmission Enable for data line 1 (SPIDATAGZEN[1]) 0h (R/W) = Data Line1 (SPIDAT[1]) selected for transmission 1h (R/W) = No transmission on Data Line1 (SPIDAT[1])
16	DPE0	R/W	0h	Transmission Enable for data line 0 (SPIDATAGZEN[0]) 0h (R/W) = Data Line0 (SPIDAT[0]) selected for transmission 1h (R/W) = No transmission on Data Line0 (SPIDAT[0])
15	DMAR	R/W	0h	DMA Read request The DMA Read request line is asserted when the channel is enabled and a new data is available in the receive register of the channel. The DMA Read request line is deasserted on read completion of the receive register of the channel. 0h (R/W) = DMA Read Request disabled 1h (R/W) = DMA Read Request enabled
14	DMAW	R/W	0h	DMA Write request. The DMA Write request line is asserted when The channel is enabled and the transmitter register of the channel is empty. The DMA Write request line is deasserted on load completion of the transmitter register of the channel. 0h (R/W) = DMA Write Request disabled 1h (R/W) = DMA Write Request enabled
13-12	TRM	R/W	0h	Transmit/Receive modes 0h (R/W) = Transmit and Receive mode 1h (R/W) = Receive only mode 2h (R/W) = Transmit only mode 3h (R/W) = Reserved

Table 26-37. MCSPI_CH3CONF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-7	WL	R/W	0h	SPI word length 0h (R/W) = Reserved 1h (R/W) = Reserved 2h (R/W) = Reserved 3h (R/W) = 4bits : The SPI word is 4-bits long 4h (R/W) = 5bits : The SPI word is 5-bits long 5h (R/W) = 6bits : The SPI word is 6-bits long 6h (R/W) = 7bits : The SPI word is 7-bits long 7h (R/W) = 8bits : The SPI word is 8-bits long 8h (R/W) = 9bits : The SPI word is 9-bits long 9h (R/W) = 10bits : The SPI word is 10-bits long Ah (R/W) = 11bits : The SPI word is 11-bits long Bh (R/W) = 12bits : The SPI word is 12-bits long Ch (R/W) = 13bits : The SPI word is 13-bits long Dh (R/W) = 14bits : The SPI word is 14-bits long Eh (R/W) = 15bits : The SPI word is 15-bits long Fh (R/W) = 16bits : The SPI word is 16-bits long 10h (R/W) = 17bits : The SPI word is 17-bits long 11h (R/W) = 18bits : The SPI word is 18-bits long 12h (R/W) = 19bits : The SPI word is 19-bits long 13h (R/W) = 20bits : The SPI word is 20-bits long 14h (R/W) = 21bits : The SPI word is 21-bits long 15h (R/W) = 22bits : The SPI word is 22-bits long 16h (R/W) = 23bits : The SPI word is 23-bits long 17h (R/W) = 24bits : The SPI word is 24-bits long 18h (R/W) = 25bits : The SPI word is 25-bits long 19h (R/W) = 26bits : The SPI word is 26-bits long 1Ah (R/W) = 27bits : The SPI word is 27-bits long 1Bh (R/W) = 28bits : The SPI word is 28-bits long 1Ch (R/W) = 29bits : The SPI word is 29-bits long 1Dh (R/W) = 30bits : The SPI word is 30-bits long 1Eh (R/W) = 31bits : The SPI word is 31-bits long 1Fh (R/W) = 32bits : The SPI word is 32-bits long
6	EPOL	R/W	0h	SPIEN polarity 0h (R/W) = SPIEN is held high during the active state. 1h (R/W) = SPIEN is held low during the active state.

Table 26-37. MCSPI_CH3CONF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-2	CLKD	R/W	0h	Frequency divider for SPICLK. (only when the module is a Master SPI device). A programmable clock divider divides the SPI reference clock (CLKSPIREF) with a 4-bit value, and results in a new clock SPICLK available to shift-in and shift-out data. By default the clock divider ratio has a power of two granularity when MCSPI_CHCONF[CLKG] is cleared, Otherwise this register is the 4 LSB bit of a 12-bit register concatenated with clock divider extension MCSPI_CHCTRL[EXTCLK] register.The value description below defines the clock ratio when MCSPI_CHCONF[CLKG] is set to 0. 0h (R/W) = 1 1h (R/W) = 2 2h (R/W) = 4 3h (R/W) = 8 4h (R/W) = 16 5h (R/W) = 32 6h (R/W) = 64 7h (R/W) = 128 8h (R/W) = 256 9h (R/W) = 512 Ah (R/W) = 1024 Bh (R/W) = 2048 Ch (R/W) = 4096 Dh (R/W) = 8192 Eh (R/W) = 16384 Fh (R/W) = 32768
1	POL	R/W	0h	SPICLK polarity 0h (R/W) = SPICLK is held high during the active state 1h (R/W) = SPICLK is held low during the active state
0	PHA	R/W	0h	SPICLK phase 0h (R/W) = Data are latched on odd numbered edges of SPICLK. 1h (R/W) = Data are latched on even numbered edges of SPICLK.

26.4.1.28 MCSPI_CH3STAT Register (Offset = 16Ch) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_CH3STAT is shown in Figure 26-53 and described in Table 26-38.

Return to Summary Table.

This register provides status information about transmitter and receiver registers of channel 3

Figure 26-53. MCSPI_CH3STAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	RXFFF	RXFFE	TXFFF	TXFFE	EOT	TXS	RXS
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 26-38. MCSPI_CH3STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Read returns 0
6	RXFFF	R	0h	Channel "i" FIFO Receive Buffer Full Status 0h (R) = FIFO Receive Buffer is not full 1h (R) = FIFO Receive Buffer is full
5	RXFFE	R	0h	Channel "i" FIFO Receive Buffer Empty Status 0h (R) = FIFO Receive Buffer is not empty 1h (R) = FIFO Receive Buffer is empty
4	TXFFF	R	0h	Channel "i" FIFO Transmit Buffer Full Status 0h (R) = FIFO Transmit Buffer is not full 1h (R) = FIFO Transmit Buffer is full
3	TXFFE	R	0h	Channel "i" FIFO Transmit Buffer Empty Status 0h (R) = FIFO Transmit Buffer is not empty 1h (R) = FIFO Transmit Buffer is empty
2	EOT	R	0h	Channel "i" End of transfer Status. The definitions of beginning and end of transfer vary with master versus slave and the transfer format (Transmit/Receive modes, Turbo mode). See dedicated chapters for details. 0h (R) = This flag is automatically cleared when the shift register is loaded with the data from the transmitter register (beginning of transfer). 1h (R) = This flag is automatically set to one at the end of an SPI transfer.
1	TXS	R	0h	Channel "i" Transmitter Register Status 0h (R) = Register is full 1h (R) = Register is empty
0	RXS	R	0h	Channel "i" Receiver Register Status 0h (R) = Register is empty 1h (R) = Register is full

26.4.1.29 MCSPI_CH3CTRL Register (Offset = 170h) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_CH3CTRL is shown in Figure 26-54 and described in Table 26-39.

Return to Summary Table.

This register is dedicated to enable the channel 3

Figure 26-54. MCSPI_CH3CTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTCLK								RESERVED							EN
R/W-0h								R/W-0h							R/W-0h

Table 26-39. MCSPI_CH3CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	Read returns 0
15-8	EXTCLK	R/W	0h	Clock ratio extension: This register is used to concatenate with MCSPI_CHCONF[CLKD] register for clock ratio only when granularity is one clock cycle (MCSPI_CHCONF[CLKG] set to 1). Then the max value reached is 4096 clock divider ratio. 0h (R/W) = Clock ratio is CLKD + 1 1h (R/W) = Clock ratio is CLKD + 1 + 16 FFh (R/W) = Clock ratio is CLKD + 1 + 4080
7-1	RESERVED	R/W	0h	Read returns 0
0	EN	R/W	0h	Channel Enable 0h (R/W) = Channel "i" is not active 1h (R/W) = Channel "i" is active

26.4.1.30 MCSPI_TX3 Register (Offset = 174h) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_TX3 is shown in Figure 26-55 and described in Table 26-40.

Return to Summary Table.

The McSPI channel FIFO transmit buffer register (MCSPI_TXx) contains a single McSPI word to transmit through the serial link, whatever the SPI word length is. Little endian host access SPI 8 bit word on 0; big endian host accesses on 3h. SPI words are transferred with MSB first. Refer to section 24.3.10 "Access to Data Registers" for more information.

Figure 26-55. MCSPI_TX3 Register

31–0
TDATA
R/W-0h

Table 26-40. MCSPI_TX3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TDATA	R/W	0h	Channel 3 Data to transmit

26.4.1.31 MCSPI_RX3 Register (Offset = 178h) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_RX3 is shown in Figure 26-56 and described in Table 26-41.

Return to Summary Table.

The McSPI channel FIFO receive buffer register (MCSPI_RXx) contains a single McSPI word received through the serial link, whatever the SPI word length is. Little endian host access SPI 8 bit word on 0h; big endian host accesses on 3h. Refer to section 26.3.10 "Access to Data Registers" for more information.

Figure 26-56. MCSPI_RX3 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">RDATA</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 26-41. MCSPI_RX3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RDATA	R	0h	Channel 3 Received Data

26.4.1.32 MCSPI_XFERLEVEL Register (Offset = 17Ch) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_XFERLEVEL is shown in Figure 26-57 and described in Table 26-42.

Return to Summary Table.

This register provides transfer levels needed while using FIFO buffer during transfer.

Figure 26-57. MCSPI_XFERLEVEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WCNT																AFL								AEL							
R/W-0h																R/W-0h								R/W-0h							

Table 26-42. MCSPI_XFERLEVEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	WCNT	R/W	0h	Spi word counterThis register holds the programmable value of number of SPI word to be transferred on channel which is using the FIFO buffer.When transfer had started, a read back in this register returns the current SPI word transfer index. 0h (R/W) = Counter not used 1h (R/W) = 1word : one word FFFEh (R/W) = 65534word : 65534 spi word FFFFh (R/W) = 65535word : 65535 spi word
15-8	AFL	R/W	0h	Buffer Almost Full This register holds the programmable almost full level value used to determine almost full buffer condition. If the user wants an interrupt or a DMA read request to be issued during a receive operation when the data buffer holds at least n bytes, then the buffer MCSPI_MODULCTRL[AFL] must be set with n-1.The size of this register is defined by the generic parameter FFNBYTE. 0h (R/W) = 1byte : one byte 1h (R/W) = 2bytes : 2 bytes FEh (R/W) = 255bytes : 255bytes FFh (R/W) = 256bytes : 256bytes
7-0	AEL	R/W	0h	Buffer Almost EmptyThis register holds the programmable almost empty level value used to determine almost empty buffer condition. If the user wants an interrupt or a DMA write request to be issued during a transmit operation when the data buffer is able to receive n bytes, then the buffer MCSPI_MODULCTRL[AEL] must be set with n-1. 0h (R/W) = 1byte : one byte 1h (R/W) = 2bytes : 2 bytes FEh (R/W) = 255bytes : 255 bytes FFh (R/W) = 256bytes : 256bytes

26.4.1.33 MCSPI_DAFTX Register (Offset = 180h) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_DAFTX is shown in Figure 26-58 and described in Table 26-43.

Return to Summary Table.

This register contains the SPI words to transmit on the serial link when FIFO used and DMA address is aligned on 256 bit.This register is an image of one of MCSPI_TX(i) register corresponding to the channel which have its FIFO enabled.

Figure 26-58. MCSPI_DAFTX Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">DAFTDATA</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 26-43. MCSPI_DAFTX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DAFTDATA	R/W	0h	FIFO Data to transmit with DMA 256 bit aligned address. This Register is only is used when MCSPI_MODULCTRL[FDAA] is set to "1" and only one of the MCSPI_CH(i)CONF[FFEW] of enabled channels is set. If these conditions are not respected any access to this register return a null value.

26.4.1.34 MCSPI_DAFRX Register (Offset = 1A0h) [reset = 0h]

Register mask: FFFFFFFFh

MCSPI_DAFRX is shown in Figure 26-59 and described in Table 26-44.

Return to Summary Table.

This register contains the SPI words to received on the serial link when FIFO used and DMA address is aligned on 256 bit.This register is an image of one of MCSPI_RX(i) register corresponding to the channel which have its FIFO enabled.

Figure 26-59. MCSPI_DAFRX Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">DAFRDATA</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 26-44. MCSPI_DAFRX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DAFRDATA	R	0h	FIFO Data to transmit with DMA 256 bit aligned address. This Register is only is used when MCSPI_MODULCTRL[FDAA] is set to "1" and only one of the MCSPI_CH(i)CONF[FFEW] of enabled channels is set. If these conditions are not respected any access to this register return a null value.

QSPI

This chapter describes the QSPI of the device.

Topic **Page**

27.1 Introduction

27.1.1 QSPI Features

The main features of the QSPI include:

- Programmable divider for serial data clock generation
- Six pin interface (DCLK, CS_N, DOUT, DIN, QDIN1, QDIN2)
- Programmable data length (No. of bits from 1-32)
- 4 external chip select signals
- Support for 3-, 4- or 6-pin SPI interface
  - 3-pin mode uses spi_dout as inout/spi_din not used
  - 4-pin mode for dual read uses spi_dout as in/spi_din as in
  - 6-pin mode uses spi_dout as in/spi_din as in/spi_qdin0 as in/spi_qdin1 as in
- Programmable transfer or frame size (No. of words from 1 to 4096)
- Optional interrupt generation on word or frame completion
- Programmable CS_N to DOUT delay from 0 to 3 DCLKs
- Programmable signal polarities
- Programmable active clock edge
- Software controllable interface allowing for any type of SPI transfer
- Control through OCP configuration port access

27.1.2 Unsupported Features

The QSPI module does not support the following features.

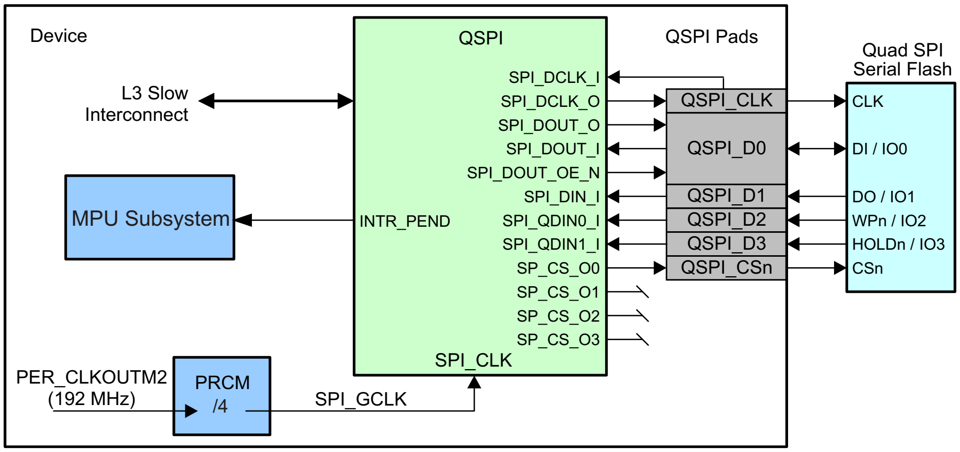
Table 27-1. Unsupported QSPI Features

Feature	Reason
External CS1-3	Not pinned out
Slave wakeup	Not connected

27.2 Integration

The QSPI module allows for single, dual or quad read access to external devices. The module supports a memory mapped interface, which provides a direct memory interface for accessing data from the external SPI device, simplifying software requirements. It is an SPI master only.

Figure 27-1. QSPI Integration



27.2.1 QSPI Connectivity Attributes

The general connectivity attributes for the QSPI module are shown in Table 27-2.

Table 27-2. QSPI Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L3S_GCLK
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart idle
Interrupt Requests	1 interrupt to MPU Subsystem (QSPIINT), PRU-ICSS
DMA Requests	None
Physical Address	L3 slow slave port Memory and control register regions qualified with MAddressSpace[2:0] bits

27.2.2 QSPI Clock and Reset Management

The QSPI has an OCP clock and a separate functional clock.

Table 27-3. QSPI Clock Signals

Clock Signal	Maximum Frequency	Reference Source	Comments
clk Interface / functional clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l3s_gclk from PRCM
spi_clk Functional clock	48 MHz	PER_CLKOUTM2 / 4	pd_per_spi_gclk from PRCM

27.2.3 QSPI Pin List

The external pins for the QSPI module are shown in Table 27-4.

Table 27-4. QSPI Pin List

Pin	Type	Description
QSPI_CLK[1]	O	QSPI output clock
QSPI_CS	O	QSPI chip select
QSPI_DATA[3:0]	I/O	QSPI data input (data[3:0]) and output (data0 only)

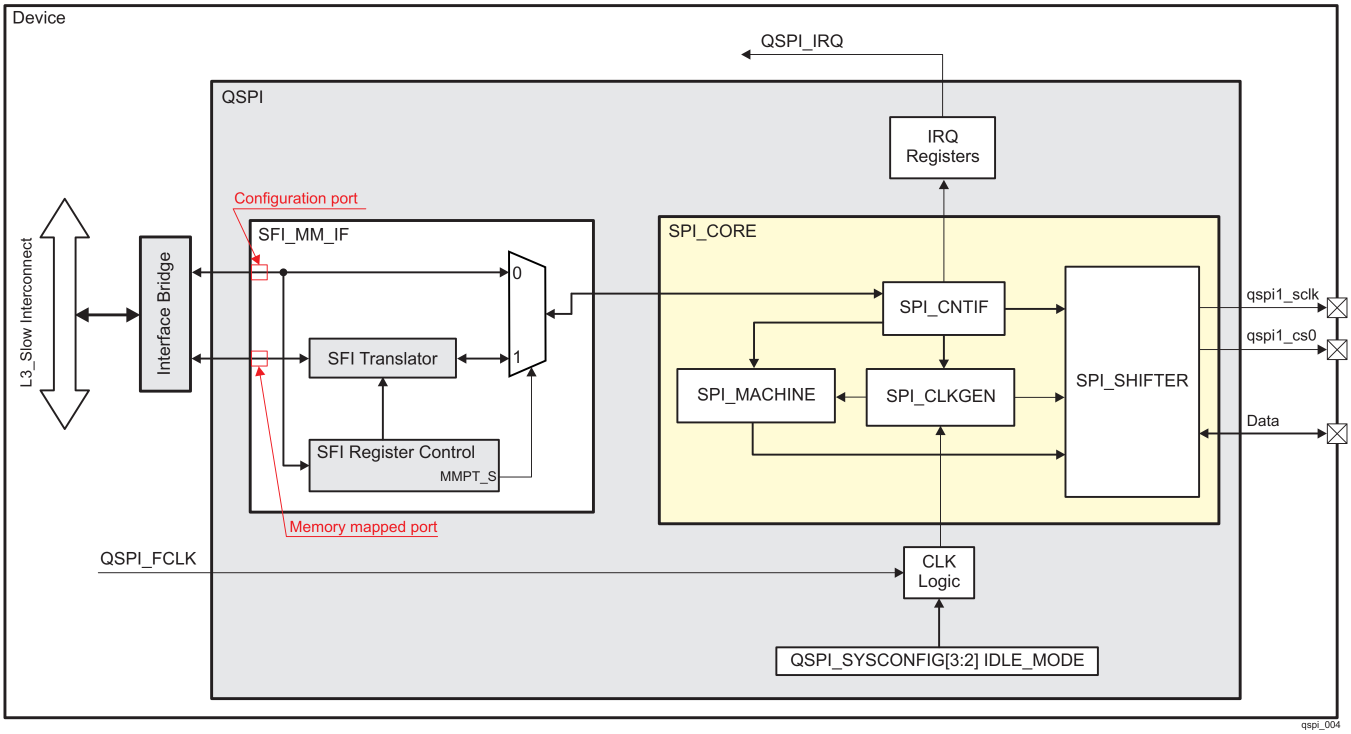
[1] These signals are also used as inputs to re-time or sync data. The associated CONF_<module>_pin_RXACTIVE bit for these signals must be set to 1 to enable the inputs back to the module. It is also recommended to place a 33-ohm resistor in series (close to the processor) on each of these signals to avoid signal reflections.

27.3 QSPI Functional Description

27.3.1 QSPI Block Diagram

Iinitial device boot from external SPI flash memory can be accomplished through the QSPI module. The interface is a simple 4-wire SPI used for control or data transfers. The QSPI also supports a 3-wire SPI protocol where the qspi1_d[0] signal is used as a bidirectional for reads and writes. In addition, a 6-wire mode can be used to support quad read devices. Figure 27-2 shows the QSPI block diagram.

Figure 27-2. QSPI Block Diagram



The QSPI is composed of two blocks. The first one is the SFI memory-mapped interface (SFI_MM_IF) and the second one is the SPI core (SPI_CORE). The SFI_MM_IF block is associated only with SPI flash memories and is used for specifying typical for the SPI flash memories settings (read or write command, number of address and dummy bytes, and so on) unlike the SPI_CORE block, which is associated with the SPI interface itself and is used to configure typical SPI settings (chip-select polarity, serial clock inactive state, SPI clock mode, length of the words transferred, and so on).

The SFI_MM_IF comprises the following two subblocks:

- SFI register control
- SFI translator

The SPI_CORE comprises the following four subblocks:

- SPI control interface (SPI_CNTIF)
- SPI clock generator (SPI_CLKGEN)
- SPI control state machine (SPI_MACHINE)
- SPI data shifter (SPI_SHIFTER)

In addition, an interface bridge connects the two ports (configuration port and memory-mapped port) of the SFI_MM_IF block to the L3_Slow interconnect. There are no software controls associated with this interface bridge.

The QSPI supports long transfers through a frame-style sequence. In its generic SPI use mode, a word can be defined up to 128 bits and multiple words can be transferred during a single access. For each word, a device initiator must read or write the new data and then tell the QSPI to continue the current operation. Using this sequence, a maximum of 4096 128-bit words can be transferred in a single SPI read or write operation. This allows great flexibility when connecting the QSPI to various types of devices.

As opposed to the generic SPI use mode, the communication with serial flash-type devices requires sending a byte command, followed by sending bytes of data. Commands can be sent through the SPI_CORE block to communicate with a serial flash device; however, it is easier to do this using the SFI_MM_IF block because it is intended to ease the communication with serial flash devices. If the SPI_CORE is used to communicate with a serial flash device, software must load the command into the SPI data transfer register with additional configuration fields, perform the byte transfer, then place the data to be sent (or configure for receive) along with additional configuration fields, and perform that transfer. Reads and writes to serial flash devices are more specific. First, the read or write command byte is sent, followed by 1 to 4 bytes of address (corresponding to the address to read/write), then followed by the data write/receive phase. Data is always sent byte oriented. When the address is loaded, data can be continuously read or written, and the address will automatically increment to each byte address internally to the serial flash device.

NOTE: The SFI_MM_IF block only allows reading and writing to an externaly connected SPI flash device. The SFI_MM_IF block does not allow reads or writes to internal configuration and status registers of the SPI flash device. These registers must be accessed through the features of the SPI_CORE block.

27.3.1.1 SFI Register Control

The SFI register control block consists of the following two configuration registers:

- QSPI_SETUP0_REG
- QSPI_SWITCH_REG

The first four registers let the user define the following:

- Byte command for a serial flash read specified by the QSPI_SETUP0_REG[7:0] RCMD bit field
- Byte command for a serial flash write specified by the QSPI_SETUP0_REG[23:16] WCMD bit field
- Number of address bytes required for the particular type of serial flash specified by the QSPI_SETUP0_REG[9:8] NUM_A_BYTES bit field
- Number of “dummy bytes” that may be needed to support the fast read mode function of some serial flash devices. The QSPI_SETUP0_REG[11:10] NUM_D_BYTES bit field specifies the number of "dummy bits." In addition, the QSPI_SETUP0_REG[28:24] NUM_D_BITS bit field can also specify the number of "dummy bits."
- Whether the read command is single (normal), dual, or quad read mode command. This is specified by the QSPI_SETUP0_REG[13:12] READ_TYPE bit field.

The QSPI_SWITCH_REG register acts as a static switch which allows the configuration port (shown in Figure 27-2) to connect directly to the SPI_CORE block, or allows the memory-mapped port (also shown in Figure 27-2) to connect to the SPI_CORE block. This is done using the QSPI_SWITCH_REG[0] MMPT_S bit.

In addition, the QSPI_SWITCH_REG[1] MM_INT_EN bit is used to enable or disable the word complete interrupt during operations using the memory-mapped port.

27.3.1.2 SFI Translator

The SFI translator block represents an FSM which, based on the configuration information loaded into the SFI register control block, converts each input read/write sequence into an SPI_CORE configuration sequence for access to the external serial flash memory.

A read sequence is converted into the following actions:

1. SPI chip-select goes active.
2. Read command byte is issued.
3. 1 to 4 address bytes, which correspond to the first address supplied, are issued.
4. 0 to 3 dummy bytes are issued, if “fast read” is supported.
5. Data bytes are read from the external SPI flash memory.
6. SPI chip-select goes inactive.

For linear addressing mode, action 5 is repeated until the byte count to be transferred reaches zero.

A write sequence is identical to a read sequence, except that a write sequence does not use dummy bytes.

Another important aspect with regard to writes is that a serial flash memory location can only be written to if the bits are erased in advance. Erased means the bits are set to 1. This means that writing only changes 1 contents to 0. It is not possible with this write to change the contents of a bit from 0 to 1. An erase command must be performed to do this operation. Erase commands cannot be executed on single byte locations. Depending on device types, there are page, block, and chip erase commands. To perform an erase command, the particular command must be sent over the SPI bus, and an internal register of the serial flash device must then be polled to determine when the erase completes. The erases must be done through the configuration port by software before performing any writes through the memory-mapped port. This means that writes are passed through to the serial flash device, but if the memory locations being modified are not properly erased before the write, the contents may not result in what was sent.

27.3.1.3 SPI Control Interface

The SPI control interface contains configuration registers used to configure the SPI core functionality of the QSPI. This block maintains all configuration settings for the SPI core (that is, settings specific for the SPI interface itself but not for the SPI flash memories).

The registers defined for this block are:

- The QSPI_PID register, which is read only and contains QSPI revision associated information
- The QSPI_CLOCK_CNTRL_REG register, which is used to control external SPI clock (qspi1_sclk)
- The QSPI_DC_REG register used to define the SPI clock mode and chip-select polarity for the four external SPI devices
- The QSPI_CMD_REG register used to control the operation of the SPI command. This register is also used to configure and transfer data.
- Four data registers used for reading the data received and for writing the data to be transferred. These registers are:
  - QSPI_DATA_REG
  - QSPI_DATA_REG_1
  - QSPI_DATA_REG_2
  - QSPI_DATA_REG_3

  These four registers compose a 128-bit shift register.
- The QSPI_STS_REG register, which contains status information

All of these registers can only be written if the QSPI is not busy. This means that they can be written if the QSPI_STS_REG[0] BUSY bit is 0x0. The QSPI becomes busy when a write to the QSPI_CMD_REG[18:16] CMD bit field is performed. Writing to this bit field starts an SPI transaction and sets the QSPI_STS_REG[0] BUSY bit to 0x1. The CMD bit field can be written again when the BUSY bit is 0x0. In addition, the start of the SPI transaction is synchronized to the qspi1_sclk clock and clearing of the BUSY bit is synchronized to the QSPI_FCLK clock.

The register group QSPI_DATA_REG_3, QSPI_DATA_REG_2, QSPI_DATA_REG_1 and QSPI_DATA_REG is treated as a single 128-bit word for shifting data in and out. The QSPI_DATA_REG_3 register is used for the most significant bits and the QSPI_DATA_REG is used for the least significant bits.This applies for both reads and writes. For example, after reading a 128-bit word (WLEN = 0x7F) the most significant bit of the data read, that is bit 127, will be located at QSPI_DATA_REG_3[31] position and the least significant bit, that is bit 0 of the data read, will be located at the QSPI_DATA_REG[0] position.

The data written to this register group should be right justified so that a data pre-shifting is not required. The QSPI_CMD_REG[25:19] WLEN bit field determines the location of the most significant bit and the bit position that will be shifted out first during a write. In order to shift out byte data the WLEN bit field should be set to 0x7 and the data byte should be written to the lower byte of the QSPI_DATA_REG register. By setting the word length to 0x7 the QSPI_DATA_REG register will look like a pseudo 8-bit shift register. When the user wants to write 40-bit long word the WLEN bit field should be set to 0x27, the 32 least significant bits of data should be written to the QSPI_DATA_REG and the rest 8 most significant bits of data should be written to the lower byte of the QSPI_DATA_REG_1 register. By setting WLEN to 0x27 these two registers will look like a pseudo 40-bit shift register. When the word length is greater than 64 bits the QSPI_DATA_REG_2 register is also used and the previously described logic applies. The QSPI_DATA_REG_3 register is used together with the other three data registers when the word length is greater than 96 bits.

When dual or quad read mode is used the number of the words transferred must be even. This number is configured through the QSPI_CMD_REG[11:0] FLEN bit field.

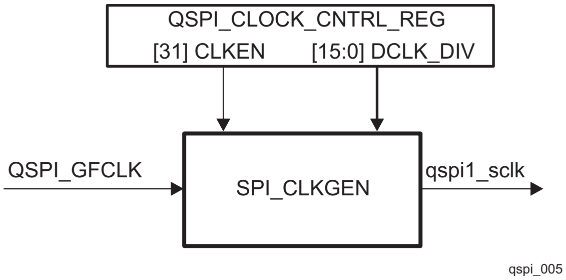
NOTE: The QSPI module does not support a "pass through" mode where the data present on qspi1_d[1] is sent to qspi1_d[0], when 4-pin non-dual read mode is used. This means that setting the QSPI_CMD_REG[18:16] CMD bit field to 0x1 causes the QSPI only to read from an external device using the qspi1_d[1] pad as an input and if a write to the same external device is desired, the CMD bit field should be set to 0x2, which causes the qspi1_d[0] pad to be used as an output.

27.3.1.4 SPI Clock Generator

The SPI clock generator uses the QSPI_FCLK clock as an input, and generates the qspi1_sclk, which is a divided version of the QSPI_FCLK clock. The divide ratio is a 16-bit value configured through the QSPI_CLOCK_CNTRL_REG[15:0] DCLK_DIV bit field and thus provides a division factor in a range from 1 to 65536. The QSPI_FCLK clock is divided by the DCLK_DIV value + 1 to provide the qspi1_sclk clock. When DCLK_DIV = 0x0 the QSPI_FCLK clock equals the DCLK clock. The value in the DCLK_DIV bit field applies only when the QSPI_CLOCK_CNTRL_REG[31] CLKEN bit is set to 0x1. Figure 27-3 shows the SPI_CLKGEN block.

If the CLKEN bit is 0x0 the command specified in the QSPI_CMD_REG[18:16] CMD bit field is not executed and the QSPI_STS_REG[0] BUSY bit is not set. The command is executed only if the CLKEN bit is 0x1 before write to the CMD bit field.

Figure 27-3. SPI_CLKGEN Block



27.3.1.5 SPI Control State-Machine

The SPI control state-machine (SPI_MACHINE) manages the operation of the SPI_CORE block. SPI_MACHINE takes control and configuration information from the registers in the SPI_CNTIF block as input and provides control information to the SPI data shifter. This information is used to control the SPI data port. The SPI_MACHINE also generates status information, which is sent back to the SPI_CNTIF block.

Writing a valid value to the QSPI_CMD_REG[18:16] CMD bit field sets immediately the QSPI_STS_REG[0] BUSY bit to 0x1, activates the corresponding qspi1_cs[n] (n = 0 to 3) and starts the SPI data transaction. The BUSY bit is cleared automatically when QSPI_CMD_REG[25:19] WLEN number of bits are shifted in or out. If the value of the QSPI_STS_REG[27:16] WDCNT bit field is different than 0x0 and WLEN number of bits are shifted already, the SPI_MACHINE waits until another write to the CMD bit field is performed. If the command written to the CMD bit field is valid, then this decrements the value of the WDCNT bit field and starts shfting data in or out again. This is repeated until the WDCNT bit field reaches 0x0, that is, all words of the frame are shifted or till earlier frame termination occurs. While the SPI_MACHINE is waiting for write to the CMD bit field the corresponding qspi1_cs[n] (n = 0 to 3) remains active and the BUSY flag is set to 0x0. In addition, the bit length for each word can be changed during a frame from 1 to 128 bits using the QSPI_CMD_REG[25:19] WLEN bit field.

The SPI_MACHINE also provides a mechanism to terminate the frame earlier. This is done by writing an invalid command to the CMD bit field. An invalid command corresponds to the 0x0 and 0x4 (reserved) values of the CMD bit field. Writing one of these values when the the WDCNT bit field is not equal to 0x0 and when the BUSY flag is 0x0 terminates the frame earlier.

The corresponding qspi1_cs[n] (n = 0 to 3) becomes inactive when all words are shifted or when the frame terminates earlier.

27.3.1.6 SPI Data Shifter

The SPI data shifter handles the capture and generation of the SPI interface signals. Based on control signals from the SPI_MACHINE and SPI_CNTIF blocks, data is shifted in or out on falling or rising edge of qspi1_sclk clock depending on the SPI clock mode selected. Table 27-5 lists the four defined clock modes of operation for the QSPI.

Table 27-5. SPI Clock Modes Definition

Mode	Settings in the QSPI_DC_REG Register		Description
	Value of the CKP bits	Value of the CKPH bits	
0	0	0	Data input captured on rising edge of qspi1_sclk clock. Data output generated on falling edge of qspi1_sclk clock
1	0	1	Data input captured on falling edge of qspi1_sclk clock. Data output generated on rising edge of qspi1_sclk clock
2	1	0	Data input captured on falling edge of qspi1_sclk clock. Data output generated on rising edge of qspi1_sclk clock
3	1	1	Data input captured on rising edge of qspi1_sclk clock. Data output generated on falling edge of qspi1_sclk clock

27.3.2 QSPI Clock Configuration

The QSPI complies with the PRCM slave-idle protocol. The QSPI_FCLK clock is gated based on the values loaded in the QSPI_SYSCONFIG[3:2] IDLE_MODE bit field. Three modes are supported:

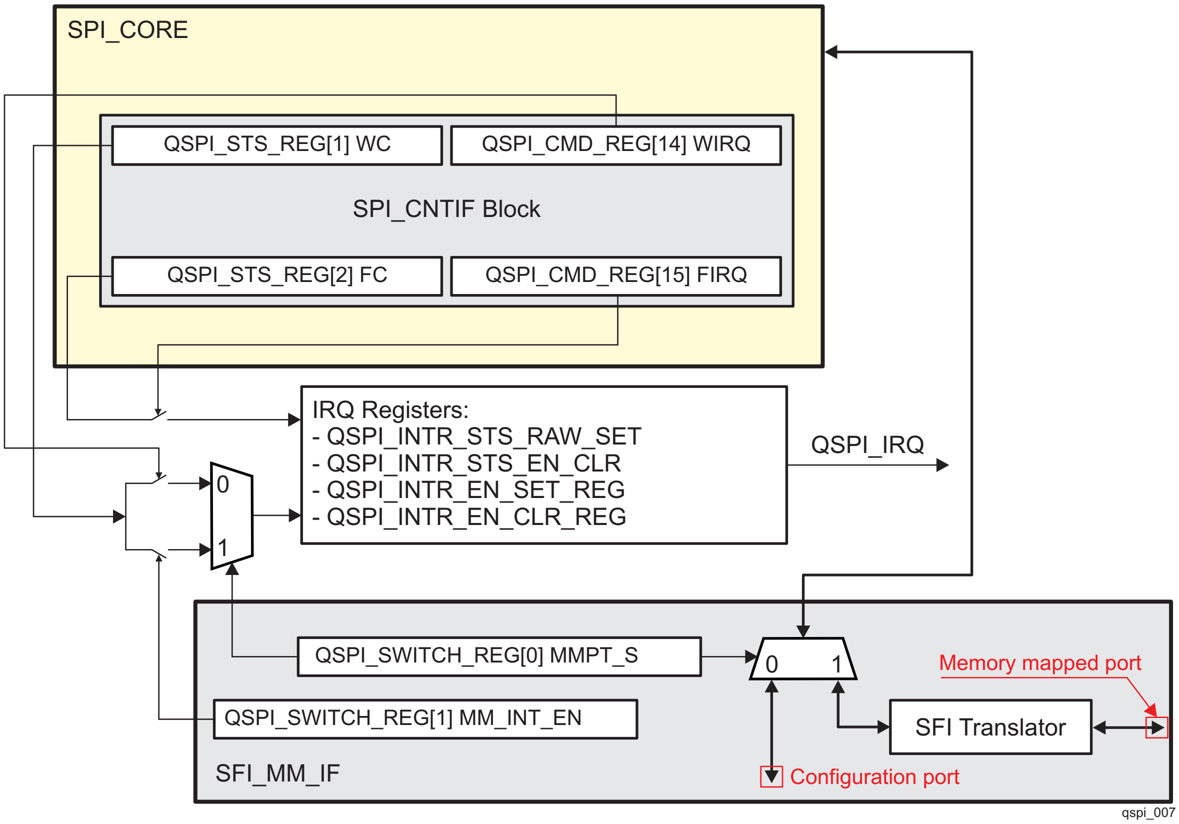
- Force-idle: The QSPI_FCLK clock is gated unconditionally by the QSPI.
- No-idle: The QSPI_FCLK clock is never gated by the QSPI.
- Smart-idle: The QSPI_FCLK clock is gated by the QSPI, depending on its internal requirements.

27.3.3 QSPI Interrupt Requests

The QSPI generates one interrupt request which is connected to the IRQ_CROSSBAR module. This interrupt request, QSPI_IRQ, is connected to the IRQ_CROSSBAR_343 input. The QSPI_IRQ interrupt line can be activated by one of the interrupt events listed in Table 27-6.

Figure 27-4 shows a logical representation of the QSPI interrupt generation scheme.

Figure 27-4. Logical Representation of the QSPI Interrupt Generation Scheme



QSPI_STS_REG[1] WC and QSPI_STS_REG[2] FC are status bits indicating whether word or frame transfer is complete. Setting the corresponding interrupt enable bit (WIRQ or FIRQ) in the QSPI_CMD_REG register allows these events (WC and FC) to generate an interrupt. The WC and FC bits are reset every time the user writes to the QSPI_CMD_REG register or reads the QSPI_STS_REG register. This is done to keep control parameters from changing the interface protocol signals while a transfer is in progress. Additionally, the QSPI_SWITCH_REG[1] MM_INT_EN bit is used to enable or disable the word complete interrupt during operations using the memory-mapped port.

When the QSPI_CMD_REG[14] WIRQ and QSPI_CMD_REG[15] FIRQ bits are set to 0x1 the following applies:

- The QSPI activates its interrupt line only if the interrupts are enabled by setting to 0x1 the corresponding bits in the QSPI_INTR_EN_SET_REG register. These interrupts can be disabled by setting the corresponding bits in the QSPI_INTR_EN_CLR_REG register to 0x1.
- After an interrupt has been serviced, software must clear the corresponding status flag. This is done by

setting the corresponding bit in the QSPI_INTR_STS_EN_CLR register to 0x1, which also clears the corresponding bit in the QSPI_INTR_STS_RAW_SET register. The status flags in the QSPI_INTR_STS_RAW_SET register are set even if the corresponding interrupt is disabled unlike those in the QSPI_INTR_STS_EN_CLR register, which are set only if the corresponding interrupt is enabled.

- The QSPI also generates an interrupt if a certain bit in the QSPI_INTR_STS_RAW_SET register is set to 0x1 and the corresponding interrupt is enabled through the QSPI_INTR_EN_SET_REG register. This feature is useful during user software debugging. In addition, even if interrupts are not enabled a corresponding raw flag in the QSPI_INTR_STS_RAW_SET register is set to 0x1 when an IRQ condition occurs.
- Even if interrupts are not enabled, a certain status bit in the QSPI_INTR_STS_RAW_SET register can also be cleared by setting to 0x1 the corresponding bit in the QSPI_INTR_STS_EN_CLR register.

It must be considered that the previously described scenario applies if the QSPI_CMD_REG[14] WIRQ and QSPI_CMD_REG[15] FIRQ bits are set to 0x1.

NOTE: The QSPI_IRQ interrupt line is activated only if at least one of the following conditions is met:

- The word complete interrupt is enabled:
  - during operations using the memory-mapped port by setting to 0x1 both the QSPI_SWITCH_REG[1] MM_INT_EN and QSPI_INTR_EN_SET_REG[1] WIRQ_ENA_SET bits.
  - during operations using the configuration port by setting to 0x1 both the QSPI_CMD_REG[14] WIRQ and QSPI_INTR_EN_SET_REG[1] WIRQ_ENA_SET bits.
- The frame complete interrupt is enabled setting to 0x1 both the QSPI_CMD_REG[15] FIRQ and QSPI_INTR_EN_SET_REG[0] FIRQ_ENA_SET bits.

The QSPI_IRQ interrupt line is also activated when both the conditions are met.

Table 27-6 lists the event flags and the corresponding mask bits of the sources which can cause interrupts.

Table 27-6. QSPI Events

Event Flag	Event Mask	Description
QSPI_INTR_STS_RAW_SET[1] WIRQ_RAW QSPI_INTR_STS_EN_CLR[1] WIRQ_ENA QSPI_STS_REG[1] WC	QSPI_INTR_EN_SET_REG[1] WIRQ_ENA_SET QSPI_INTR_EN_CLR_REG[1] WIRQ_ENA_CLR QSPI_CMD_REG[14] WIRQ	Word complete interrupt event. Asserted each time after a word is transferred or received.
QSPI_INTR_STS_RAW_SET[0] FIRQ_RAW QSPI_INTR_STS_EN_CLR[0] FIRQ_ENA QSPI_STS_REG[2] FC	QSPI_INTR_EN_SET_REG[0] FIRQ_ENA_SET QSPI_INTR_EN_CLR_REG[0] FIRQ_ENA_CLR QSPI_CMD_REG[15] FIRQ	Frame complete interrupt event. Asserted each time after a frame is transferred or received.

27.3.4 QSPI Memory Regions

Two memory regions are associated with the QSPI. The first memory region is dedicated to the configuration port. Using this memory region, all internal registers can be programmed and serial transfers made from the external SPI device. The L3_Slow start address at which the configuration port is available is 0x4790 0000. The second memory region is associated mainly with the memory-mapped port and is used for communication directly with the external SPI device. This memory region starts at 0x3000 0000 and ends at 0x33FF FFFF L3_Slow address.

It is important to keep in mind that the configuration port provides an access to all the QSPI registers listed in Section 27.4.1. These are configuration registers and also four data registers. The configuration registers are used to configure typical SPI and serial flash memory settings and the four data registers are used for read and write operations. When communicating with an external SPI device (but not an SPI flash memory), the SPI_CORE module should be used and the data exchanged is available through these four data registers, which can be accessed only through the configuration port. When a communication with an external SPI flash memory is desired, the memory-mapped port should be used.

In other words, to read from an external SPI flash memory, first configure the QSPI through the configuration port and then perform a read through the memory-mapped port.

27.4 QSPI Registers

27.4.1 QSPI Registers

Table 27-7 lists the memory-mapped registers for the QSPI. All register offset addresses not listed in Table 27-7 should be considered as reserved locations and the register contents should not be modified.

Table 27-7. QSPI Registers

Offset	Acronym	Register Name	Section
0h	QSPI_PID		Section 27.4.1.1
10h	QSPI_SYSCONFIG		Section 27.4.1.2
20h	QSPI_INTR_STS_RAW_SET		Section 27.4.1.3
24h	QSPI_INTR_STS_EN_CLR		Section 27.4.1.4
28h	QSPI_INTR_EN_SET_REG		Section 27.4.1.5
2Ch	QSPI_INTR_EN_CLR_REG		Section 27.4.1.6
30h	QSPI_INTC_EOI_REG		Section 27.4.1.7
40h	QSPI_CLOCK_CNTRL_REG		Section 27.4.1.8
44h	QSPI_DC_REG		Section 27.4.1.9
48h	QSPI_CMD_REG		Section 27.4.1.10
4Ch	QSPI_STS_REG		Section 27.4.1.11
50h	QSPI_DATA_REG		Section 27.4.1.12
54h to 60h	QSPI_SETUP_REG_0 to QSPI_SETUP_REG_3		Section 27.4.1.13
64h	QSPI_SWITCH_REG		Section 27.4.1.14
68h	QSPI_DATA_REG_1		Section 27.4.1.15
6Ch	QSPI_DATA_REG_2		Section 27.4.1.16
70h	QSPI_DATA_REG_3		Section 27.4.1.17

27.4.1.1 QSPI_PID Register (offset = 0h) [reset = 4F400000h]

QSPI_PID is shown in Figure 27-5 and described in Table 27-8.

Figure 27-5. QSPI_PID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCHEME		RSVD		FUNC											
R-1h		R-0h		R-F40h											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RTL_VERSION					MAJOR			CUSTOM		MINOR					
R-0h					R-0h			R-0h		R-0h					

Table 27-8. QSPI_PID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	The scheme of the register used. This indicates the PDR3.5 Method
29-28	RSVD	R	0h	
27-16	FUNC	R	F40h	The function of the module being used
15-11	RTL_VERSION	R	0h	RTL version number
10-8	MAJOR	R	0h	Major revision number
7-6	CUSTOM	R	0h	
5-0	MINOR	R	0h	Minor Revision Number

27.4.1.2 QSPI_SYSCONFIG Register (offset = 10h) [reset = 28h]

QSPI_SYSCONFIG is shown in Figure 27-6 and described in Table 27-9.

Figure 27-6. QSPI_SYSCONFIG Register

31	30	29	28	27	26	25	24
RSVD_2							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RSVD_2							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RSVD_2							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RSVD_2		RESERVED		IDLE_MODE		RSVD_1	
R-0h		W-2h		R/W-2h		R-0h	

Table 27-9. QSPI_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RSVD_2	R	0h	
5-4	RESERVED	W	2h	
3-2	IDLE_MODE	R/W	2h	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state 0h (R/W) = Local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements. 1h (R/W) = No-idle mode: local target never enters idle state. 2h (R/W) = Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wakeup events 3h (R/W) = Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state. Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented
1-0	RSVD_1	R	0h	

27.4.1.3 QSPI_INTR_STS_RAW_SET Register (offset = 20h) [reset = 0h]

QSPI_INTR_STS_RAW_SET is shown in Figure 27-7 and described in Table 27-10.

This register contains the raw interrupt status.

Figure 27-7. QSPI_INTR_STS_RAW_SET Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RSVD</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RSVD</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RSVD</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RSVD</td><td>WIRQ_RAW</td><td>FIRQ_RAW</td></tr>
<tr><td colspan="6">R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 27-10. QSPI_INTR_STS_RAW_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RSVD	R/W	0h	
1	WIRQ_RAW	R/W	0h	Word Interrupt Status. Read indicates raw status. Writing 1 will set status. Writing 0 has no effect. 0h (R/W) = Inactive. 1h (R/W) = Active.
0	FIRQ_RAW	R/W	0h	Frame Interrupt Status. Read indicates raw status. Writing 1 will set status. Writing 0 has no effect. 0h (R/W) = Inactive. 1h (R/W) = Active.

27.4.1.4 QSPI_INTR_STS_EN_CLR Register (offset = 24h) [reset = 0h]

QSPI_INTR_STS_EN_CLR is shown in Figure 27-8 and described in Table 27-11.

Interrupt Status Enabled/Clear Register.

Figure 27-8. QSPI_INTR_STS_EN_CLR Register

31	30	29	28	27	26	25	24
RSVD							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RSVD							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RSVD							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RSVD						WIRQ_ENA	FIRQ_ENA
R-0h						R/W-0h	R/W-0h

Table 27-11. QSPI_INTR_STS_EN_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RSVD	R	0h	
1	WIRQ_ENA	R/W	0h	Word Interrupt Enabled Status. Read indicates enabled status. Writing 1 will clear interrupt. Writing 0 has no effect. 0h (R/W) = Inactive. 1h (R/W) = Active.
0	FIRQ_ENA	R/W	0h	Frame Interrupt Enabled Status. Read indicates enabled status. Writing 1 will clear interrupt. Writing 0 has no effect. 0h (R/W) = Inactive. 1h (R/W) = Active.

27.4.1.5 QSPI_INTR_EN_SET_REG Register (offset = 28h) [reset = 0h]

QSPI_INTR_EN_SET_REG is shown in Figure 27-9 and described in Table 27-12.

Interrupt Enable/Set Register.

Figure 27-9. QSPI_INTR_EN_SET_REG Register

31	30	29	28	27	26	25	24
RSVD							
R-0h							

23	22	21	20	19	18	17	16
RSVD							
R-0h							

15	14	13	12	11	10	9	8
RSVD							
R-0h							

7	6	5	4	3	2	1	0
RSVD						WIRQ_ENA_SET	FIRQ_ENA_SET
R-0h						R/W-0h	R/W-0h

Table 27-12. QSPI_INTR_EN_SET_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RSVD	R	0h	
1	WIRQ_ENA_SET	R/W	0h	Word Interrupt Enable/Set. Read indicates interrupt enable. Writing 1 will set interrupt enabled. Writing 0 has no effect. 0h (R/W) = Inactive. 1h (R/W) = Active.
0	FIRQ_ENA_SET	R/W	0h	Frame Interrupt Enable/Set. Read indicates interrupt enable. Writing 1 will set interrupt enabled. Writing 0 has no effect. 0h (R/W) = Inactive. 1h (R/W) = Active.

27.4.1.6 QSPI_INTR_EN_CLR_REG Register (offset = 2Ch) [reset = 0h]

QSPI_INTR_EN_CLR_REG is shown in Figure 27-10 and described in Table 27-13.

Interrupt Enable/Clear Register.

Figure 27-10. QSPI_INTR_EN_CLR_REG Register

31	30	29	28	27	26	25	24
RSVD							
R-0h							
23	22	21	20	19	18	17	16
RSVD							
R-0h							
15	14	13	12	11	10	9	8
RSVD							
R-0h							
7	6	5	4	3	2	1	0
RSVD						WIRQ_ENA_CLR	FIRQ_ENA_CLR
R-0h						R/W-0h	R/W-0h

Table 27-13. QSPI_INTR_EN_CLR_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RSVD	R	0h	
1	WIRQ_ENA_CLR	R/W	0h	Word Interrupt Enable/Clear. Read indicates interrupt enable. Writing 1 will clear interrupt enabled. Writing 0 has no effect. 0h (R/W) = Inactive. 1h (R/W) = Active.
0	FIRQ_ENA_CLR	R/W	0h	Frame Interrupt Enable/Clear. Read indicates interrupt enable. Writing 1 will clear interrupt enabled. Writing 0 has no effect. 0h (R/W) = Inactive. 1h (R/W) = Active.

27.4.1.7 QSPI_INTC_EOI_REG Register (offset = 30h) [reset = 0h]

QSPI_INTC_EOI_REG is shown in Figure 27-11 and described in Table 27-14.

INTC EOI Register.

Figure 27-11. QSPI_INTC_EOI_REG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOI_VECTOR																															
R/W-0h																															

Table 27-14. QSPI_INTC_EOI_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	EOI_VECTOR	R/W	0h	Number associated with the ipgenericirq for intr output. There are 1 interrupt outputs. Write 0x0 : Write to intr IP Generic. Any other write value is ignored.

27.4.1.8 QSPI_CLOCK_CNTRL_REG Register (offset = 40h) [reset = 0h]

QSPI_CLOCK_CNTRL_REG is shown in Figure 27-12 and described in Table 27-15.

SPI Clock Control (SPICC) Register. SPICC controls the SPI clock generation. The SPICC controls the SPI clock generation. The input for clock division is the input SPI_CLK signal. The output clock will be divided by DCLK_DIV+1 to provide the output SPI interface clock as well as controlling the main portion of the design. Note that loading a value or 0 input DCLK_DIV will force the input SPI_CLK to be used directly for the SPI interface clock. The value in DCLK_DIV is only loaded when CLKEN transitions from a 0 to 1 state. This register can only be written to when the SPI is not busy, as defined by SPISR[0].

Figure 27-12. QSPI_CLOCK_CNTRL_REG Register

31	30	29	28	27	26	25	24
CLKEN	RESERVED						
R/W-0h	R-0h						
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
DCLK_DIV							
R/W-0h							
7	6	5	4	3	2	1	0
DCLK_DIV							
R/W-0h							

Table 27-15. QSPI_CLOCK_CNTRL_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CLKEN	R/W	0h	Clock Enable. 0h (R/W) = Data clock is turned off. 1h (R/W) = Data clock is enabled.
30-16	RESERVED	R	0h	
15-0	DCLK_DIV	R/W	0h	Serial data clock divide by ratio

27.4.1.9 QSPI_DC_REG Register (offset = 44h) [reset = 0h]

QSPI_DC_REG is shown in Figure 27-13 and described in Table 27-16.

SPI Device Control (SPIDC) Register. The SPIDC controls the different modes for each output chip select. The SPIDC controls the different modes for each output chip select. NOTE: The combination of [CKPn, CKPHn] creates the SPI mode . Most serial Flash devices only support SPI modes 0 and 3. SPI devices transmit and receive data on opposite edge s of the SPI clock. Note that changing the clock polarity also swaps the transmit/receive clock edge relationship. If a slave device states that it receives data on the rising edge and transmits on the falling edge of the clock, then it can only support mode 0 or 3 (CKPn = 0, CKPHn = 0 OR CKPn = 1, CKPHn = 1). This register can only be written to when the SPI is not busy, as defined by SPISR[0].

Figure 27-13. QSPI_DC_REG Register

31	30	29	28	27	26	25	24
RSVD_3			DD3		CKPH3	CSP3	CKP3
R-0h			0h		0h	0h	0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RSVD_2			DD2		CKPH2	CSP2	CKP2
R-0h			0h		0h	0h	0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RSVD_1			DD1		CKPH1	CSP1	CKP1
R-0h			0h		0h	0h	0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RSVD_0			DD0		CKPH0	CSP0	CKP0
R-0h			0h		0h	0h	0h

Table 27-16. QSPI_DC_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RSVD_3	R	0h	
28-27	DD3		0h	Data delay for chip select 3. 0h (R/W) = Data is output on the same cycle as the CS_N goes active. 1h (R/W) = Data is output 1 DCLK cycle after the CS_N goes active. 2h (R/W) = Data is output 2 DCLK cycles after the CS_N goes active. 3h (R/W) = Data is output 3 DCLK cycles after the CS_N goes active.
26	CKPH3		0h	Clock phase for chip select 3. See register description note. 0h (R/W) = If CKP3 = 0, data shifted out on falling edge and input on rising edge. If CKP3 = 1, data shifted out on rising edge and input on falling edge. 1h (R/W) = If CKP3 = 0, data shifted out on rising edge and input on falling edge. If CKP3 = 1, data shifted out on falling edge and input on rising edge.
25	CSP3		0h	Chip select polarity for chip select 3. 0h (R/W) = Active low. 1h (R/W) = Active high.
24	CKP3		0h	Clock polarity for chip select 3. 0h (R/W) = When data is not being transferred, SCK = 0 1h (R/W) = When data is not being transferred, SCK = 1
23-21	RSVD_2	R	0h	

Table 27-16. QSPI_DC_REG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20-19	DD2		0h	Data delay for chip select 2. 0h (R/W) = Data is output on the same cycle as the CS_N goes active. 1h (R/W) = Data is output 1 DCLK cycle after the CS_N goes active. 2h (R/W) = Data is output 2 DCLK cycles after the CS_N goes active. 3h (R/W) = Data is output 3 DCLK cycles after the CS_N goes active.
18	CKPH2		0h	Clock phase for chip select 2. See register description note. 0h (R/W) = If CKP2 = 0, data shifted out on falling edge and input on rising edge. If CKP2 = 1, data shifted out on rising edge and input on falling edge. 1h (R/W) = If CKP2 = 0, data shifted out on rising edge and input on falling edge. If CKP2 = 1, data shifted out on falling edge and input on rising edge.
17	CSP2		0h	Chip select polarity for chip select 2. 0h (R/W) = Active low. 1h (R/W) = Active high.
16	CKP2		0h	Clock polarity for chip select 2. 0h (R/W) = When data is not being transferred, SCK = 0 1h (R/W) = When data is not being transferred, SCK = 1
15-13	RSVD_1	R	0h	
12-11	DD1		0h	Data delay for chip select 1. 0h (R/W) = Data is output on the same cycle as the CS_N goes active. 1h (R/W) = Data is output 1 DCLK cycle after the CS_N goes active. 2h (R/W) = Data is output 2 DCLK cycles after the CS_N goes active. 3h (R/W) = Data is output 3 DCLK cycles after the CS_N goes active.
10	CKPH1		0h	Clock phase for chip select 1. See register description note. 0h (R/W) = If CKP1 = 0, data shifted out on falling edge and input on rising edge. If CKP1 = 1, data shifted out on rising edge and input on falling edge. 1h (R/W) = If CKP1 = 0, data shifted out on rising edge and input on falling edge. If CKP1 = 1, data shifted out on falling edge and input on rising edge.
9	CSP1		0h	Chip select polarity for chip select 1. 0h (R/W) = Active low. 1h (R/W) = Active high.
8	CKP1		0h	Clock polarity for chip select 1. 0h (R/W) = When data is not being transferred, SCK = 0 1h (R/W) = When data is not being transferred, SCK = 1
7-5	RSVD_0	R	0h	
4-3	DD0		0h	Data delay for chip select 0. 0h (R/W) = Data is output on the same cycle as the CS_N goes active. 1h (R/W) = Data is output 1 DCLK cycle after the CS_N goes active. 2h (R/W) = Data is output 2 DCLK cycles after the CS_N goes active. 3h (R/W) = Data is output 3 DCLK cycles after the CS_N goes active.

Table 27-16. QSPI_DC_REG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	CKPH0		0h	Clock phase for chip select 0. See register description note. 0h (R/W) = If CKP0 = 0, data shifted out on falling edge and input on rising edge. If CKP0 = 1, data shifted out on rising edge and input on falling edge. 1h (R/W) = If CKP0 = 0, data shifted out on rising edge and input on falling edge. If CKP0 = 1, data shifted out on falling edge and input on rising edge.
1	CSP0		0h	Chip select polarity for chip select. 0h (R/W) = Active low. 1h (R/W) = Active high.
0	CKP0		0h	Clock polarity for chip select. 0h (R/W) = When data is not being transferred, SCK = 0 1h (R/W) = When data is not being transferred, SCK = 1

27.4.1.10 QSPI_CMD_REG Register (offset = 48h) [reset = 0h]

QSPI_CMD_REG is shown in Figure 27-14 and described in Table 27-17.

SPI Command Register (SPICR). Sets up the SPI command. Since the SPI convention is to always shift in new data (LSB position) while shifting out data from the MSB position the 'read', 'write' and 'read dual' commands actually just initiate a transfer. Write commands will assert the spi_dout_oe_n signal to enable the tristate buffer for spi_dout_o. Read commands will deassert the spi_dout_oe_n signal to disable the tristate buffer for spi_dout_o. Executing any of these commands will initiate the next word transfer (except for the case of "read dual"). The 'read dual' command will also initiate a transfer like read or write. However, this command is used to communicate with serial Flash devices which support dual read output. Dual read output mode uses the spi_dout signal as an input along with the spi_din input (spi_dout is therefore a bidirectional signal). WLEN transfers must be even. The spi_din input will contain the odd number bytes and the spi_dout bidirectional/input will contain the even number bytes. This read mode effectively doubles the read bandwidth of the design. This command can only be used on those devices which support a dual read command.

The 'read quad' command is similar to the read dual command, except it uses a 6 pin interface. The

In particular, transfers are started by writing to byte 2 of this register (only byte 2 will start a transfer). Writing a "reserved" value to the CMD register will terminate the frame transfer. This can be used to abort a frame if desired. Reserved commands are CMD = 00 and CMD_MODE=1 AND CMD=11.

This register can only be written to when the SPI is not busy, as defined by SPISR[0]. If the SPI clock has not been enabled (SPICC[31] = 0), then writing to this register will NOT start the SPI transfer. The transfer will not be queued to start upon the activation of the clock. The transfer will ONLY start if SPICC[31] = 1 when this register is written.

Figure 27-14. QSPI_CMD_REG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="2">RSVD_3</td><td colspan="2">CSNUM</td><td colspan="4">RSVD_2</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="2">R/W-0h</td><td colspan="4">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="5">WLEN</td><td colspan="3">CMD</td></tr>
<tr><td colspan="5">R/W-0h</td><td colspan="3">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>FIRQ</td><td>WIRQ</td><td colspan="2">RSVD</td><td colspan="4">FLEN</td></tr>
<tr><td>R/W-0h</td><td>R/W-0h</td><td colspan="2">R-0h</td><td colspan="4">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">FLEN</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 27-17. QSPI_CMD_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RSVD_3	R	0h	Reserved.
29-28	CSNUM	R/W	0h	Device select. Sets the active chip select for the transfer. 0h (R/W) = Chip select 0 active. 1h (R/W) = Reserved. 2h (R/W) = Reserved. 3h (R/W) = Reserved.
27-24	RSVD_2	R	0h	Reserved.
23-19	WLEN	R/W	0h	Word length. Sets the size of the individual transfers from 1 to 32 bits. 0h (R/W) = 1 bit. 1h (R/W) = 2 bits. 1Fh (R/W) = 32 bits.

Table 27-17. QSPI_CMD_REG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18-16	CMD	R/W	0h	Transfer command. 1h (R/W) = 4 pin Read Single 2h (R/W) = 4 pin Write Single 3h (R/W) = 4 pin Read Dual 5h (R/W) = 3 pin Read Single 6h (R/W) = 3 pin Write Single 7h (R/W) = 6 pin Read Quad
15	FIRQ	R/W	0h	Frame count interrupt enable. 0h (R/W) = Disable. 1h (R/W) = Enable.
14	WIRQ	R/W	0h	Word count interrupt enable. 0h (R/W) = Disable. 1h (R/W) = Enable.
13-12	RSVD	R	0h	Reserved
11-0	FLEN	R/W	0h	Frame Length. 0h (R/W) = 1 word. 1h (R/W) = 2 words. FFFh (R/W) = 4096 words.

27.4.1.11 QSPI_STS_REG Register (offset = 4Ch) [reset = 0h]

QSPI_STS_REG is shown in Figure 27-15 and described in Table 27-18.

SPI Status Register (SPISR). The SPI Status Register contains indicators to allow the user to monitor the progression of a frame transfer. The SPI Status Register contains indicators to allow the user to monitor the progression of a frame transfer. The word complete (WC) and frame complete (FC) bits are used as stimulus for generating interrupts. Setting the corresponding interrupt enable bit in the SPI Command Register will allow these events to generate an interrupt. The WDCNT field will reset itself when transitioning from the LOAD state to the SHIFT state. THE WC and FC fields will be reset every time the user writes to the SPI Command Register or the SPI Status Register is read.
The BUSY bit of this register (SPISR[0]) is used to block write access to the SPICC, SPIDC, SPICR and SPIDR registers. This is done to keep control parameters from changing the interface protocol signals while a transfer is in progress. If a write is made while BUSY is active, the write will simply not occur. BUSY is set immediately when SPICR[18:16] are written and BUSY will be cleared when the current word has completed transfer (on SPISR[1]). However, if the SPI clock enable is not set (SPICC[31] = 0) and a command is executed (SPICR[18:16] is written to start a command), BUSY will NOT be set and the command will not be executed. BUSY will ONLY go high if SPICC[31] = 1 and SPICR[18:16] is written.

Figure 27-15. QSPI_STS_REG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="4">RSVD_2</td><td colspan="4">WDCNT</td></tr>
<tr><td colspan="4">R-0h</td><td colspan="4">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">WDCNT</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RSVD</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">RSVD</td><td>FC</td><td>WC</td><td>BUSY</td></tr>
<tr><td colspan="5">R-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 27-18. QSPI_STS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RSVD_2	R	0h	
27-16	WDCNT	R/W	0h	Word count. This field will reflect the 1 to 4096 words transferred
15-3	RSVD	R	0h	
2	FC	R/W	0h	Frame complete. This bit is set after all of the requested words have been transmitted. 0h (R/W) = Transfer is not complete. 1h (R/W) = Transfer is complete. This bit is reset when the SPI Status Register is read.
1	WC	R/W	0h	Word complete. This bit is set after each word transfer is completed. 0h (R/W) = Word transfer is not complete. 1h (R/W) = Word transfer is complete. This bit is reset when the SPI Status Register is read.
0	BUSY	R/W	0h	Busy bit. Active transfer in progress. This bit is only set during an active word transfer. Between words, the bit will clear to signal that it is ok to read/write the data registers. 0h (R/W) = Idle. 1h (R/W) = Busy.

27.4.1.12 QSPI_DATA_REG Register (offset = 50h) [reset = 0h]

QSPI_DATA_REG is shown in Figure 27-16 and described in Table 27-19.

SPI Data Register (SPIDR). Data received in the data register is shifted into the LSB position and the contents of the register are shifted to the left. The register is cleared between reads or writes. Data received in the data register is shifted into the LSB position and the contents of the register are shifted to the left. The register is cleared between reads or writes.
When writing data to the register it should be right justified so pre-shifting is not required. The word length field will determine the location of the most significant bit and the bit position that will be shifted out first during a write. In order to shift out byte data the word length should be set to 7 and the data byte should be written to the lower byte of the data register. By setting the word length to 7 the data register will look like a pseudo 8-bit shift register. The word length setting does not affect the VBUSP read or write operations.
This register can only be written to when the SPI is not busy, as defined by SPISR[0].

Figure 27-16. QSPI_DATA_REG Register

31 – 0
DATA
0h

Table 27-19. QSPI_DATA_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA		0h	Data register for read and write operations

27.4.1.13 QSPI_SETUP_REG_0 to QSPI_SETUP_REG_3 Register (offset = 54h to 60h) [reset = 20203h]

QSPI_SETUP_REG_0 to QSPI_SETUP_REG_3 is shown in Figure 27-17 and described in Table 27-20.

Memory Mapped SPI Setup Register n. The Memory Mapped SPI Setup Register contains the read/write command setup for the Memory Mapped Protocol Translator (effecting Chip Select n output).
The Memory Mapped SPI Setup Register contains the read/write command setup for the Memory Mapped Protocol Translator, request n input (effecting Chip Select n output). This corresponds to MAddrSpace = 001 (n = 0), MAddrSpace = 010 (n = 1), MAddrSpace = 011 (n = 2), and MAddrSpace = 100, 101, 110, and 111 (n = 3) .
Note that by default (reset), the device uses a write command of 2, read command of 3 and number of address bytes of 3. This default covers most serial Flash devices, but can be changed.

Figure 27-17. QSPI_SETUP_REG_0 to QSPI_SETUP_REG_3 Register

31	30	29	28	27	26	25	24
RSVD_2			NUM_D_BITS				
R-0h			0h				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WCMD							
2h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RSVD_1		READ_TYPE		NUM_D_BYTES		NUM_A_BYTES	
R-0h		0h		0h		2h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RCMD							
3h							

Table 27-20. QSPI_SETUP_REG_0 to QSPI_SETUP_REG_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RSVD_2	R	0h	
28-24	NUM_D_BITS		0h	Number of dummy bits to use if NUM_D_BYTES = 0.
23-16	WCMD		2h	Write command.
15-14	RSVD_1	R	0h	
13-12	READ_TYPE		0h	Determines if the read command is a single, dual or quad read mode command. 0h (R/W) = Normal read (all data input on spi_din) 1h (R/W) = Dual read (odd bytes input on spi_din; even on spi_dout) 2h (R/W) = Normal read (all data input on spi_din) 3h (R/W) = Quad read (uses spi_qdin0/1)
11-10	NUM_D_BYTES		0h	Number of dummy bytes to be used for fast read. 0h (R/W) = No dummy bytes required. 1h (R/W) = 1 2h (R/W) = 2 3h (R/W) = 3
9-8	NUM_A_BYTES		2h	Number of address bytes to be sent. 0h (R/W) = 1 byte. 1h (R/W) = 2 bytes. 2h (R/W) = 3 bytes. 3h (R/W) = 4 bytes.
7-0	RCMD		3h	Read Command.

27.4.1.14 QSPI_SWITCH_REG Register (offset = 64h) [reset = 0h]

QSPI_SWITCH_REG is shown in Figure 27-18 and described in Table 27-21.

Memory Mapped SPI Switch Register. The Memory Mapped SPI Switch Register allows the CPU to switch control of the core SPI module's configuration port between the configuration VBUSP port and the Memory Mapped Protocol Translator.
The Memory Mapped SPI Switch Register allows the CPU to switch control of the core SPI module s configuration port between the configuration VBUSP port and the Memory Mapped Protocol Translator.
In addition, an interrupt enable field is defined which is used to enable or disable word count interrupt generation in memory mapped mode.

Figure 27-18. QSPI_SWITCH_REG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RSVD</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RSVD</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RSVD</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RSVD</td><td>MM_INT_EN</td><td>MMPT_S</td></tr>
<tr><td colspan="6">R-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 27-21. QSPI_SWITCH_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RSVD	R	0h	
1	MM_INT_EN	R/W	0h	Memory Mapped mode interrupt enable. 0h (R/W) = Interrupts are disabled during memory mapped operations. 1h (R/W) = Word count interrupts are enabled for memory mapped operations.
0	MMPT_S	R/W	0h	MMPT select. 0h (R/W) = Configuration port is selected to control configuration of core SPI module. 1h (R/W) = Memory mapped protocol translator is selected to control configuration port of core SPI module.

27.4.1.15 QSPI_DATA_REG_1 Register (offset = 68h) [reset = 0h]

QSPI_DATA_REG_1 is shown in Figure 27-19 and described in Table 27-22.

SPI Data1 Register (SPIDR1). Data received in the data register is shifted into the LSB position and the contents of the register are shifted to the left. The register is cleared between reads or writes. This acts as the 2nd 32 bit register of the 128 bit shift register in/out
Data received in the data register is shifted into the LSB position and the contents of the register are shifted to the left. The register is cleared between reads or writes.
When writing data to the register it should be right justified so pre-shifting is not required. The word length field will determine the location of the most significant bit and the bit position that will be shifted out first during a write. In order to shift out byte data the word length should be set to '7' and the data byte should be written to the lower byte of the data register. By setting the word length to '7' the data register will look like a pseudo 8-bit shift register. The word length setting does not affect the VBUSP read or write operations.

Figure 27-19. QSPI_DATA_REG_1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">DATA</td></tr>
<tr><td colspan="32">0h</td></tr>
</table>

Table 27-22. QSPI_DATA_REG_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA		0h	Data register for read and write operations

27.4.1.16 QSPI_DATA_REG_2 Register (offset = 6Ch) [reset = 0h]

QSPI_DATA_REG_2 is shown in Figure 27-20 and described in Table 27-23.

SPI Data2 Register (SPIDR2).
Data received in the data register is shifted into the LSB position and the contents of the register are shifted to the left. The register is cleared between reads or writes. This acts as the 2nd 32 bit register of the 128 bit shift register in/out
Data received in the data register is shifted into the LSB position and the contents of the register are shifted to the left. The register is cleared between reads or writes.
When writing data to the register it should be right justified so pre-shifting is not required. The word length field will determine the location of the most significant bit and the bit position that will be shifted out first during a write. In order to shift out byte data the word length should be set to '7' and the data byte should be written to the lower byte of the data register. By setting the word length to '7' the data register will look like a pseudo 8-bit shift register. The word length setting does not affect the VBUSP read or write operations.

Figure 27-20. QSPI_DATA_REG_2 Register

31–0
DATA
0h

Table 27-23. QSPI_DATA_REG_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA		0h	Data register for read and write operations

27.4.1.17 QSPI_DATA_REG_3 Register (offset = 70h) [reset = 0h]

QSPI_DATA_REG_3 is shown in Figure 27-21 and described in Table 27-24.

SPI Data3 Register (SPIDR3).
Data received in the data register is shifted into the LSB position and the contents of the register are shifted to the left. The register is cleared between reads or writes. This acts as the 2nd 32 bit register of the 128 bit shift register in/out
Data received in the data register is shifted into the LSB position and the contents of the register are shifted to the left. The register is cleared between reads or writes.
When writing data to the register it should be right justified so pre-shifting is not required. The word length field will determine the location of the most significant bit and the bit position that will be shifted out first during a write. In order to shift out byte data the word length should be set to '7' and the data byte should be written to the lower byte of the data register. By setting the word length to '7' the data register will look like a pseudo 8-bit shift register. The word length setting does not affect the VBUSP read or write operations.

Figure 27-21. QSPI_DATA_REG_3 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">DATA</td></tr>
<tr><td colspan="32">0h</td></tr>
</table>

Table 27-24. QSPI_DATA_REG_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA		0h	Data register for read and write operations

Chapter 28

General-Purpose Input/Output

This chapter describes the GPIO of the device.

Topic **Page**

28.1 Introduction

28.1.1 Purpose of the Peripheral

The general-purpose interface combines six general-purpose input/output (GPIO) modules. Each GPIO module provides 32 dedicated general-purpose pins with input and output capabilities; thus, the general-purpose interface supports up to 192 (6 × 32) pins. These pins can be configured for the following applications:

- Data input (capture)/output (drive)
- Keyboard interface with a debounce cell
- Interrupt generation in active mode upon the detection of external events. Detected events are processed by two parallel independent interrupt-generation submodules to support biprocessor operations.
- Wake-up request generation in idle mode upon the detection of external events.

28.1.2 GPIO Features

Each GPIO module is made up of 32 identical channels. Each channel can be configured to be used in the following applications:

- Data input/output
- Keyboard interface with a de-bouncing cell
- Synchronous interrupt generation (in active mode) upon the detection of external events (signal transition(s) and/or signal level(s))
- Wake-up request generation (in Idle mode) upon the detection of signal transition(s)

Global features of the GPIO interface are:

- Synchronous interrupt requests from each channel are processed by two identical interrupt generation sub-modules to be used independently by the ARM Subsystem
- Wake-up requests from input channels are merged together to issue one wake-up signal to the system
- Shared registers can be accessed through “Set & Clear” protocol

28.1.3 Unsupported GPIO Features

The wake-up feature of the GPIO modules is only supported on GPIO0.

28.2 Integration

The device instantiates six GPIO_V2 modules. Each GPIO module provides the support for 32 dedicated pins with input and output configuration capabilities. Input signals can be used to generate interruptions and wake-up signal. Two Interrupt lines are available for bi-processor operation. Pins can be dedicated to be used as a keyboard controller.

With six GPIO modules, the device allows for a maximum of 192 GPIO pins. (The exact number available varies as a function of the device configuration and pin muxing.) GPIO0 is in the Wakeup domain and may be used to wakeup the device via external sources. GPIO[1:5] are located in the peripheral domain.

Figure 28-1 and Figure 28-2 show the GPIO integration.

Figure 28-1. GPIO0 Module Integration

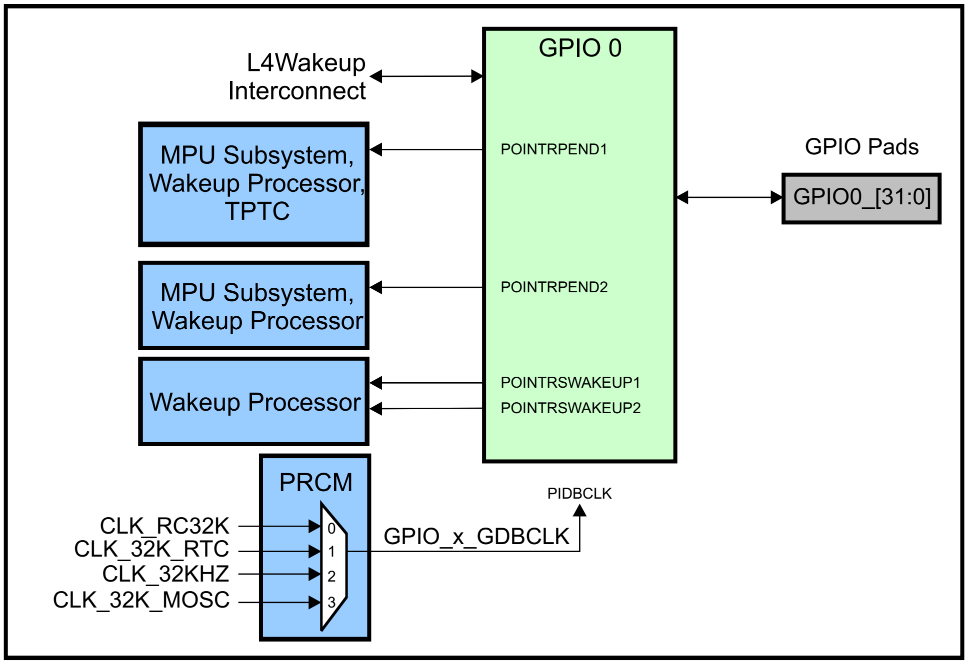
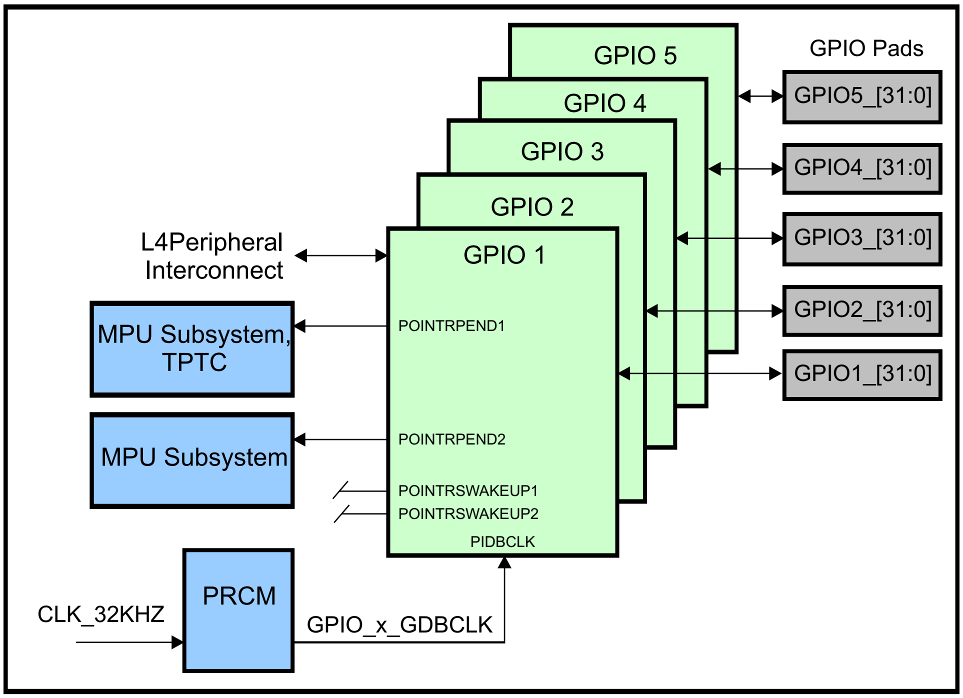


Figure 28-2. GPIO[1–5] Module Integration



28.2.1 GPIO Connectivity Attributes

The general connectivity attributes for the GPIO modules in the device are shown in Table 28-1 and Table 28-2.

Table 28-1. GPIO0 Connectivity Attributes

Attributes	Type
Power Domain	Wakeup Domain
Clock Domain	PD_WKUP_L4_WKUP_GCLK (OCP) GPIO_0_GDBCLK (Debounce)
Reset Signals	WKUP_DOM_RST_N
Idle/Wakeup Signals	Smart Idle / Slave Wakeup
Interrupt Requests	Two Interrupts: INTRPEND1 (GPIOINT0A) to MPU subsystem and Wakeup Processor INTRPEND2 (GPIOINT0B) to MPU subsystem and Wakeup Processor Two wakeups to Wakeup Processor: WAKEUP1 and WAKEUP2
DMA Requests	Interrupt Requests are redirected as DMA requests: 1 DMA request (GPIOEVT0)
Physical Address	L4 Wakeup slave port

Table 28-2. GPIO[1:5] Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L4LS_GCLK (OCP) GPIO_1_GDBCLK (GPIO1 Debounce) GPIO_2_GDBCLK (GPIO2 Debounce) GPIO_3_GDBCLK (GPIO3 Debounce) GPIO_4_GDBCLK (GPIO4 Debounce) GPIO_5_GDBCLK (GPIO5 Debounce)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	Two Interrupts: INTRPEND1 (GPIOINTxA) to MPU subsystem INTRPEND2 (GPIOINTxB) to MPU subsystem
DMA Requests	Interrupt Requests are redirected as DMA requests: 1 DMA request per instance (GPIOEVTx)
Physical Address	L4 Peripheral slave port

28.2.2 GPIO Clock and Reset Management

The GPIO modules require two clocks: The de-bounce clock is used for the de-bouncing cells. The interface clock provided by the peripheral bus (L4 interface) is also the functional clock and is used through the entire GPIO module (except within the de-bouncing sub-module). It clocks the OCP interface and the internal logic. For GPIO0 the debounce clock is selected from one of the following sources using the CLKSEL_GPIO0_DBCLK register in the PRCM:

- The on-chip ~32.768 kHz oscillator (CLK_RC32K)
- The PER PLL generated 32.768 kHz clock (CLK_32KHZ)
- The external 32.768 kHz oscillator/clock for the RTC module (CLK_32K_RTC)
- The divided down 32.768 kHz oscillator/clock from the master oscillator (CLK_32K_MOSC)

Table 28-3. GPIO Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
GPIO0			
Functional / Interface clock	26 MHz	CLK_M_OSC	pd_wkup_l4_wkup_gclk From PRCM
Debounce Functional clock	32.768 KHz	CLK_RC32K CLK_32KHZ (PER_CLKOUTM2 / 5859.375) CLK_32K_RTC CLK_32K_MOSC (CLK_M_OSC divide down)	pd_wkup_gpio0_gdbclk From PRCM
GPIO[1:5]			
Functional / Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk From PRCM
Debounce Functional clock (GPIO1)	32.768 KHz	CLK_32KHZ (PER_CLKOUTM2 / 5859.375)	pd_per_gpio_1_gdbclk From PRCM
Debounce Functional clock (GPIO2)	32.768 KHz	CLK_32KHZ (PER_CLKOUTM2 / 5859.375)	pd_per_gpio_2_gdbclk From PRCM
Debounce Functional clock (GPIO3)	32.768 KHz	CLK_32KHZ (PER_CLKOUTM2 / 5859.375)	pd_per_gpio_3_gdbclk From PRCM
Debounce Functional clock (GPIO4)	32.768 KHz	CLK_32KHZ (PER_CLKOUTM2 / 5859.375)	pd_per_gpio_4_gdbclk From PRCM
Debounce Functional clock (GPIO5)	32.768 KHz	CLK_32KHZ (PER_CLKOUTM2 / 5859.375)	pd_per_gpio_5_gdbclk From PRCM

28.2.3 GPIO Pin List

Each GPIO module includes 32 interface I/Os. These signals are designated as shown in Table 28-4. Note that for this device, most of these signals will be multiplexed with functional signals from other interfaces.

Table 28-4. GPIO Pin List

Pin	Type	Description
GPIO0_[31:0]	I/O	General-purpose input and output pins
GPIO1_[31:0]		
GPIO2_[31:0]		
GPIO3_[31:0]		
GPIO4_[31:0]		
GPIO5_[31:0]		

28.3 Functional Description

This section discusses the operational details and basic functions of the GPIO peripheral.

28.3.1 Operating Modes

Four operating modes are defined for the module:

- **Active mode:** the module is running synchronously on the interface clock, interrupt can be generated according to the configuration and the external signals.
- **Idle mode:** the module is in a waiting state, interface clock can be stopped , no interrupt can be generated, a wake-up signal can be generated according to the configuration and external signals. If the debouncing clock is active, the debouncing cell can be used to sample and to filter the input to generate a wakeup event.
- **Inactive mode:** the module has no activity, interface clock can be stopped, no interrupt can be generated, and the wake-up feature is inhibited.
- **Disabled mode:** the module is not used, internal clock paths are gated, no interrupt or wake-up request can be generated.

The Idle and Inactive modes are configured within the module and activated on request by the host processor through system interface sideband signals. The Disabled mode is set by software through a dedicated configuration bit. It unconditionally gates the internal clock paths not use for the system interface. All module registers are 8, 16 or 32-bit accessible through the OCP compatible interface (little endian encoding). In active mode, the event detection (level or transition) is performed in the GPIO module using the interface clock. The detection's precision is set by the frequency of this clock and the selected internal gating scheme.

28.3.2 Clocking and Reset Strategy

28.3.2.1 Clocks

GPIO module runs using two clocks:

- The debouncing clock is used for the debouncing sub-module logic (without the corresponding configuration registers). This module can sample the input line and filters the input level using a programmed delay.
- The interface clock provided by the peripheral bus (OCP compatible system interface). It is used through the entire GPIO module (except within the debouncing sub-module logic). It clocks the OCP interface and the internal logic. Clock gating features allow adapting the module power consumption to the activity.

28.3.2.2 Clocks, Gating and Active Edge Definitions

The interface clock provided by the peripheral bus (OCP compatible system interface) is used through the entire GPIO module. Two clock domains are defined: the OCP interface and the internal logic. Each clock domain can be controlled independently. Sampling operations for the data capture and for the events detection are done using the rising edge. The data loaded in the data output register (GPIO_DATAOUT) is set at the output GPIO pins synchronously with the rising edge of the interface clock.

Five clock gating features are available:

- Clock for the system interface logic can be gated when the module is not accessed, if the AUTOIDLE configuration bit in the system configuration register (GPIO_SYSCONFIG) is set. Otherwise, this logic is free running on the interface clock.
- Clock for the input data sample logic can be gated when the data in register is not accessed.
- Four clock groups are used for the logic in the synchronous events detection. Each 8 input GPIO_V2 pins group will have a separate enable signal depending on the edge/level detection register setting. If a group requires no detection, then the corresponding clock will be gated. All channels are also gated using a 'one out of N' scheme. N can take the values 1, 2, 4 or 8. The interface clock is enabled for this logic one cycle every N cycles. When N is equal to 1, there is no gating and this logic is free running on the interface clock. When N is between 2 to 8, this logic is running at the equivalent

frequency of interface clock frequency divided by N.
- In Inactive mode, all internal clock paths are gated.
- In Disabled mode, all internal clock paths not used for the system interface are gated. All GPIO registers are accessible synchronously with the interface clock.

28.3.2.3 Sleep Mode Request and Acknowledge

Upon a Sleep mode request issued by the host processor, the GPIO module goes to the Idle mode according to the IDLEMODE field in the system configuration register (GPIO_SYSCONFIG).

- IDLEMODE = 0 (Force-Idle mode): the GPIO goes in Inactive mode independently of the internal module state and the Idle acknowledge is unconditionally sent. In Force-Idle mode, the module is in Inactive mode and its wake-up feature is totally inhibited.
- IDLEMODE = 1h (No-Idle mode): the GPIO does not go to the Idle mode and the Idle acknowledge is never sent.
- IDLEMODE = 2h (Smart-Idle mode) or IDLEMODE = 3h (Smart-Idle mode): the GPIO module evaluates its internal capability to have the interface clock switched off. Once there is no more internal activity (the data input register completed to capture the input GPIO pins, there is no pending interrupt, all interrupt status bits are cleared, and there is no write access to GPIO_DEBOUNCINGTIME register pending to be synchronized), the Idle acknowledge is asserted and the GPIO enters Idle mode, ready to issue a wake-up request when the expected transition occurs on an enabled GPIO input pin. This wake-up request is effectively sent only if the ENAWAKEUP bit in GPIO_SYSCONFIG is set to enable the GPIO wakeup capability. When the system is awake, the Idle Request goes inactive, the Idle acknowledge and wake-up request (if it is the GPIO that triggered the system’s wakeup) signals are immediately de-asserted, and the asynchronous wake-up request (if existing) is reflected into the synchronous interrupt status registers.

NOTE: Idle mode request and Idle acknowledge are system interface sideband signals. Once the GPIO acknowledges the Sleep mode request (Idle acknowledge has been sent), the interface clock can be stopped anytime.

Upon a Sleep mode request issued by the host processor, the GPIO module goes to the Idle mode only if there is no active bit in GPIO_IRQSTS_RAW_n registers.

28.3.2.4 Reset

The OCP hardware Reset signal has a global reset action on the GPIO. All configuration registers, all DFFs clocked with the Interface clock or Debouncing clock and all internal state machines are reset when the OCP hardware Reset is active (low level). The RESETDONE bit in the system status register (GPIO_SYSSTS) monitors the internal reset status: it is set when the Reset is completed on both OCP and Debouncing clock domains. The software Reset (SOFTRESET bit in the system configuration register) has the same effect as the OCP hardware Reset signal, and the RESETDONE bit in GPIO_SYSSTS is updated in the same condition.

28.3.3 Interrupt and Wake-up Features

28.3.3.1 Functional Description

In order to generate an interrupt request to a host processor upon a defined event (level or logic transition) occurring on a GPIO pin, the GPIO configuration registers have to be programmed as follows:

- Interrupts for the GPIO channel must be enabled in the GPIO_IRQSTS_SET_0 and/or GPIO_IRQSTS_SET_1 registers.
- The expected event(s) on input GPIO to trigger the interrupt request has to be selected in the GPIO_LEVELDETECT0, GPIO_LEVELDETECT1, GPIO_RISINGDETECT, and GPIO_FALLINGDETECT registers.

In order to generate a wake-up request to a host processor upon a defined event (logic transition) occurring on a GPIO pin, the GPIO configuration registers have to be programmed as follows:

- The GPIO channel must be enabled in the GPIO_IRQWAKEN register.
- The expected event(s) on input GPIO to trigger the interrupt (or the wake-up) request has to be selected in the GPIO_RISINGDETECT and GPIO_FALLINGDETECT registers. Wake-up request can only be generated on rising and/or on falling transitions.

For instance, interrupt generation on both edges on input k is configured by setting to 1 the *k*th bit in registers GPIO_RISINGDETECT and GPIO_FALLINGDETECT along with the interrupt enabling for one or both interrupt lines (GPIO_IRQSTS_SET_n).

NOTE: All interrupt (or wake-up) sources (the 32 input GPIO channels) are merged together to issue two synchronous interrupt requests 1 and 2, and two asynchronous wake-up requests.

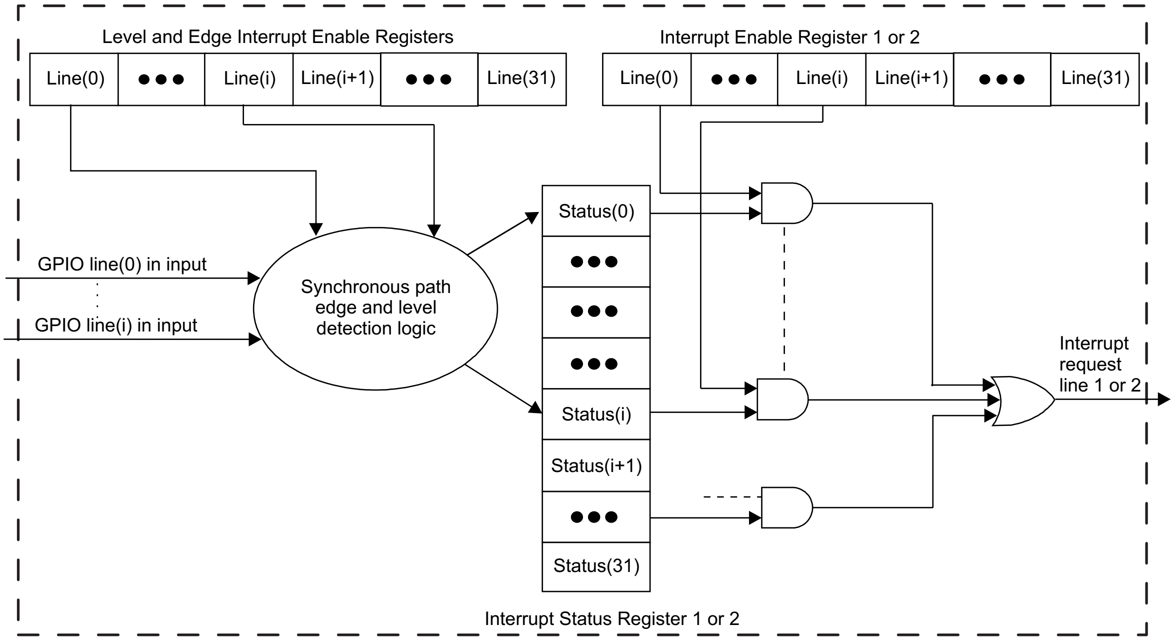
28.3.3.2 Synchronous Path: Interrupt Request Generation

In Active mode, once the GPIO configuration registers have been set to enable the interrupt generation, a synchronous path (Figure 28-3) samples the transitions and levels on the input GPIO with the internally gated interface clock. When an event matches the programmed settings, the corresponding bit in the GPIO_IRQSTS_RAW_n registers is set to 1 and, on the following interface clock cycle, the interrupt lines 1 and/or 2 are activated (depending on the GPIO_IRQSTS_SET_n registers).

Due to the sampling operation, the minimum pulse width on the input GPIO to trigger a synchronous interrupt request is two times the internally gated interface clock period (the internally gated interface clock period is equal to N times the interface clock period). This minimum pulse width has to be met before and after any expected level transition detection. Level detection requires the selected level to be stable for at least two times the internally gated interface clock period to trigger a synchronous interrupt.

As the module is synchronous, latency is minimal between the expected event occurrence and the activation of the interrupt line(s). This should not exceed 3 internally gated interface clock cycles + 2 interface clock cycles when the debouncing feature is not used. When the debouncing feature is active, the latency depends on the GPIO_DEBOUNCINGTIME register value and should be less than 3 internally gated interface clock cycles + 2 interface clock cycles + GPIO_DEBOUNCINGTIME value debouncing clock cycles + 3 debouncing clock cycles.

Figure 28-3. Interrupt Request Generation



GPIO0–5

28.3.3.3 Asynchronous Path: Wake-up Request Generation

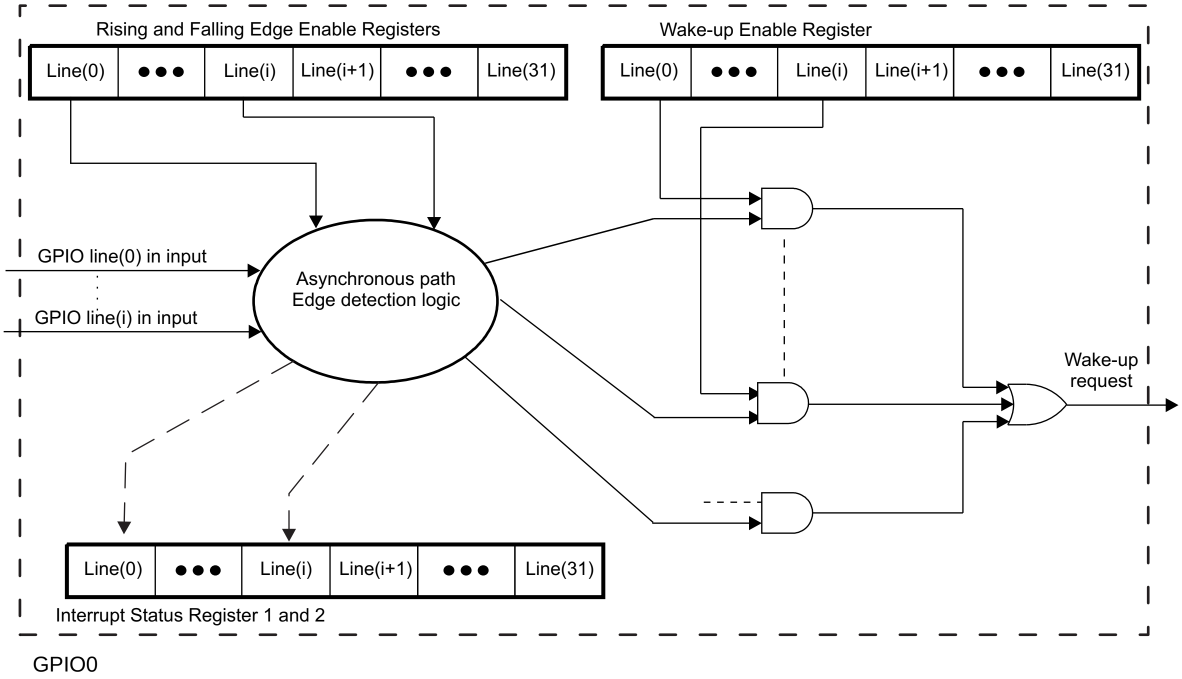
In Idle mode (interface clock is shut down, the GPIO configuration registers have been previously programmed), an asynchronous path (Figure 28-4) detects the expected transition(s) on input GPIO (according to the registers programming) and activates an asynchronous wake-up request if the wakeup enable register is set. There is only one external wake-up line, since all wake-up sources are merged together. Once the system is in wake up, the interface clock is re-started and, according to the input GPIO pin that triggered the wake-up request, the corresponding bits in the GPIO_IRQSTS_RAW_n registers are synchronously set to 1; on the following internal clock cycle, the interrupt lines 1 and/or 2 are active (active high) when the corresponding bits in GPIO_IRQSTS_SET_n registers are set.

NOTE: If debouncing is not enabled, there is no minimum input pulse width to trigger the wake-up request since there is no sampling operation.

If debouncing is used, the minimum pulse width is set by the debouncing specified time.

The ENAWAKEUP bit in the system configuration register (GPIO_SYSCONFIG) enables or disables the GPIO wake-up feature globally: if this bit is 0, the GPIO_IRQWAKEN has no effect.

Figure 28-4. Wake-Up Request Generation



28.3.3.4 Interrupt (or Wake-up) Line Release

When the host processor receives an interrupt request issued by the GPIO module, it can read the corresponding GPIO_IRQSTS_n register to find out which input GPIO has triggered the interrupt (or the wake-up request). After servicing the interrupt (or acknowledging the wake-up request), the processor resets the status bit and releases the interrupt line by writing a 1 in the corresponding bit of the GPIO_IRQSTS_n register. If there is still a pending interrupt request to serve (all bits in the GPIO_IRQSTS_RAW_n register not masked by the GPIO_IRQSTS_SET_n, which are not cleared by setting the GPIO_IRQSTS_CLR_n), the interrupt line will be re-asserted.

28.3.4 General-Purpose Interface Basic Programming Model

28.3.4.1 Power Saving by Grouping the Edge/Level Detection

Each GPIO module implements four gated clocks used by the edge/level detection logic to save power. Each group of eight input GPIO pins generates a separate enable signal depending on the edge/level detection register setting (because the input is 32 bits, four groups of eight inputs are defined for each GPIO module). If a group requires no edge/level detection, then the corresponding clock is gated (cut off). Grouping the edge/level enable can save the power consumption of the module as described in the following example.

If any of the registers:

- GPIO_LEVELDETECT0
- GPIO_LEVELDETECT1
- GPIO_RISINGDETECT
- GPIO_FALLINGDETECT

are set to 0101 0101h, then all clocks are active (power consumption is high);

are set to 0000 00FFh, then a single clock is active.

NOTE: When the clocks are enabled by writing to the GPIO_LEVELDETECT0, GPIO_LEVELDETECT1, GPIO_RISINGDETECT, and GPIO_FALLINGDETECT registers, the detection starts after 5 clock cycles. This period is required to clean the synchronization edge/level detection pipeline.

The mechanism is independent of each clock group. If the clock has been started before a new setting is performed, the following is recommended: first, set the new detection required; second, disable the previous setting (if necessary). In this way, the corresponding clock is not gated and the detection starts immediately.

28.3.4.2 Set and Clear Instructions

The GPIO module implements the set-and-clear protocol register update for the data output and interrupt enable, and wake-up enable registers. This protocol is an alternative to the atomic test and set operations and consists of writing operations at dedicated addresses (one address for setting bit[s] and one address for clearing bit[s]). The data to write is 1 at bit position(s) to clear (or to set) and 0 at unaffected bit(s).

Registers can be accessed in two ways:

- Standard: Full register read and write operations at the primary register address
- Set and clear (recommended): Separate addresses are provided to set (and clear) bits in registers. Writing 1 at these addresses sets (or clears) the corresponding bit into the equivalent register; writing a 0 has no effect.

Therefore, for these registers, three addresses are defined for one unique physical register. Reading these addresses has the same effect and returns the register value.

28.3.4.2.1 Clear Instruction

28.3.4.2.1.1 Clear Interrupt Enable Registers (GPIO_IRQSTS_CLR_0 and GPIO_IRQSTS_CLR_1):

- A write operation in the clear interrupt enable1 (or enable2) register clears the corresponding bit in the interrupt enable1 (or enable2) register when the written bit is 1; a written bit at 0 has no effect.
- A read of the clear interrupt enable1 (or enable2) register returns the value of the interrupt enable1 (or enable2) register.

28.3.4.2.1.2 Clear Wake-up Enable Register (GPIO_CLRWKUENA):

- A write operation in the clear wake-up enable register clears the corresponding bit in the wake-up enable register when the written bit is 1; a written bit at 0 has no effect.
- A read of the clear wake-up enable register returns the value of the wake-up enable register.

28.3.4.2.1.3 Clear Data Output Register (GPIO_CLRDATAOUT):

- A write operation in the clear data output register clears the corresponding bit in the data output register when the written bit is 1; a written bit at 0 has no effect.
- A read of the clear data output register returns the value of the data output register.

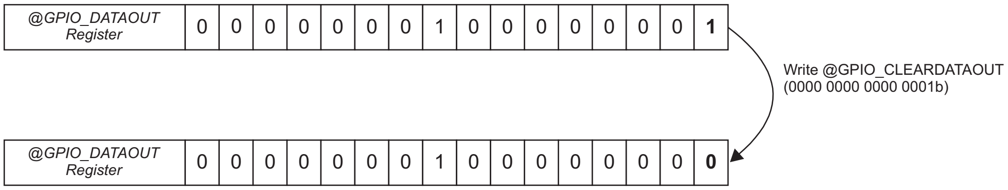
28.3.4.2.1.4 Clear Instruction Example

Assume the data output register (or one of the interrupt/wake-up enable registers) contains the binary value, 0000 0001 0000 0001h, and you want to clear bit 0.

With the clear instruction feature, write 0000 0000 0000 0001h at the address of the clear data output register (or at the address of the clear interrupt/wake-up enable register). After this write operation, a reading of the data output register (or the interrupt/wake-up enable register) returns 0000 0001 0000 0000h; bit 0 is cleared.

NOTE: Although the general-purpose interface registers are 32-bits wide, only the 16 least-significant bits are represented in this example.

Figure 28-5. Write @ GPIO_CLRDATAOUT Register Example



28.3.4.2.2 Set Instruction

28.3.4.2.2.1 Set Interrupt Enable Registers (GPIO_IRQSTS_SET_0 and GPIO_IRQSTS_SET_1):

- A write operation in the set interrupt enable1 (or enable2) register sets the corresponding bit in the interrupt enable1 (or enable2) register when the written bit is 1; a written bit at 0 has no effect.
- A read of the set interrupt enable1 (or enable2) register returns the value of the interrupt enable1 (or enable2) register.

28.3.4.2.2.2 Set Wake-up Enable Register (GPIO_SETWKUENA):

- A write operation in the set wake-up enable register sets the corresponding bit in the wake-up enable register when the written bit is 1; a written bit at 0 has no effect.
- A read of the set wake-up enable register returns the value of the wake-up enable register.

28.3.4.2.2.3 Set Data Output Register (GPIO_SETDATAOUT):

- A write operation in the set data output register sets the corresponding bit in the data output register when the written bit is 1; a written bit at 0 has no effect.
- A read of the set data output register returns the value of the data output register.

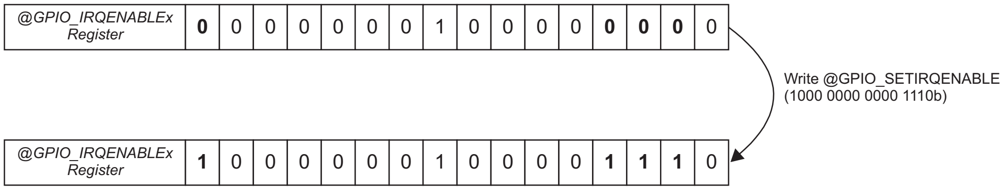
28.3.4.2.2.4 *Set Instruction Example*

Assume the interrupt enable1 (or enable2) register (or the data output register) contains the binary value, 0000 0001 0000 0000h, and you want to set bits 15, 3, 2, and 1.

With the set instruction feature, write 1000 0000 0000 1110h at the address of the set interrupt enable1 (or enable2) register (or at the address of the set data output register). After this write operation, a reading of the interrupt enable1 (or enable2) register (or the data output register) returns 1000 0001 0000 1110h; bits 15, 3, 2, and 1 are set.

NOTE: Although the general-purpose interface registers are 32-bits wide, only the 16 least-significant bits are represented in this example.

Figure 28-6. Write @ GPIO_SETIRQENx Register Example



28.3.4.3 Data Input (Capture)/Output (Drive)

The output enable register (GPIO_OE) controls the output/input capability for each pin. At reset, all the GPIO-related pins are configured as input and output capabilities are disabled. This register is not used within the module; its only function is to carry the pads configuration.

When configured as an output (the desired bit reset in GPIO_OE), the value of the corresponding bit in the GPIO_DATAOUT register is driven on the corresponding GPIO pin. Data is written to the data output register synchronously with the interface clock. This register can be accessed with read/write operations or by using the alternate set and clear protocol register update feature. This feature lets you set or clear specific bits of this register with a single write access to the set data output register (GPIO_SETDATAOUT) or to the clear data output register (GPIO_CLRDATAOUT) address. If the application uses a pin as an output and does not want interrupt/wake-up generation from this pin, the application must properly configure the wake-up enable and the interrupt enable registers.

When configured as an input (the desired bit set to 1 in GPIO_OE), the state of the input can be read from the corresponding bit in the GPIO_DATAIN register. The input data is sampled synchronously with the interface clock and then captured in the data input register synchronously with the interface clock. When the GPIO pin levels change, they are captured into this register after two interface clock cycles (the required cycles to synchronize and to write data). If the application uses a pin as an input, the application must properly configure the wake-up enable and the interrupt enable registers to the interrupt and wake-up feature as needed.

28.3.4.4 Debouncing Time

To enable the debounce feature for a pin, the GPIO configuration registers must be programmed as follows:

- The GPIO pin must be configured as input in the output enable register (write 1 to the corresponding bit of the GPIO_OE register).
- The debouncing time must be set in the debouncing value register (GPIO_DEBOUNCINGTIME). The GPIO_DEBOUNCINGTIME register is used to set the debouncing time for all input lines in the GPIO module. The value is global for all the ports of one GPIO module, so up to six different debouncing values are possible. The debounce cell is running with the debounce clock (32 kHz). This register represents the number of the clock cycle(s) (one cycle is 31 microseconds long) to be used.

  The following formula describes the required input stable time to be propagated to the debounced output:

Debouncing time = (DEBOUNCETIME + 1) × 31 µs

Where the DEBOUNCETIME field value in the GPIO_DEBOUNCINGTIME register is from 0 to 255.

- The debouncing feature must be enabled in the debouncing enable register (write 1 to the corresponding DEBOUNCEEN bit in the GPIO_DEBOUNCEN register).

28.3.4.5 GPIO as a Keyboard Interface

The general-purpose interface can be used as a keyboard interface (Figure 28-7). You can dedicate channels based on the keyboard matrix = * c). Figure 28-7 shows row channels configured as inputs with the input debounce feature enabled. The row channels are driven high with an external pull-up. Column channels are configured as outputs and drive a low level.

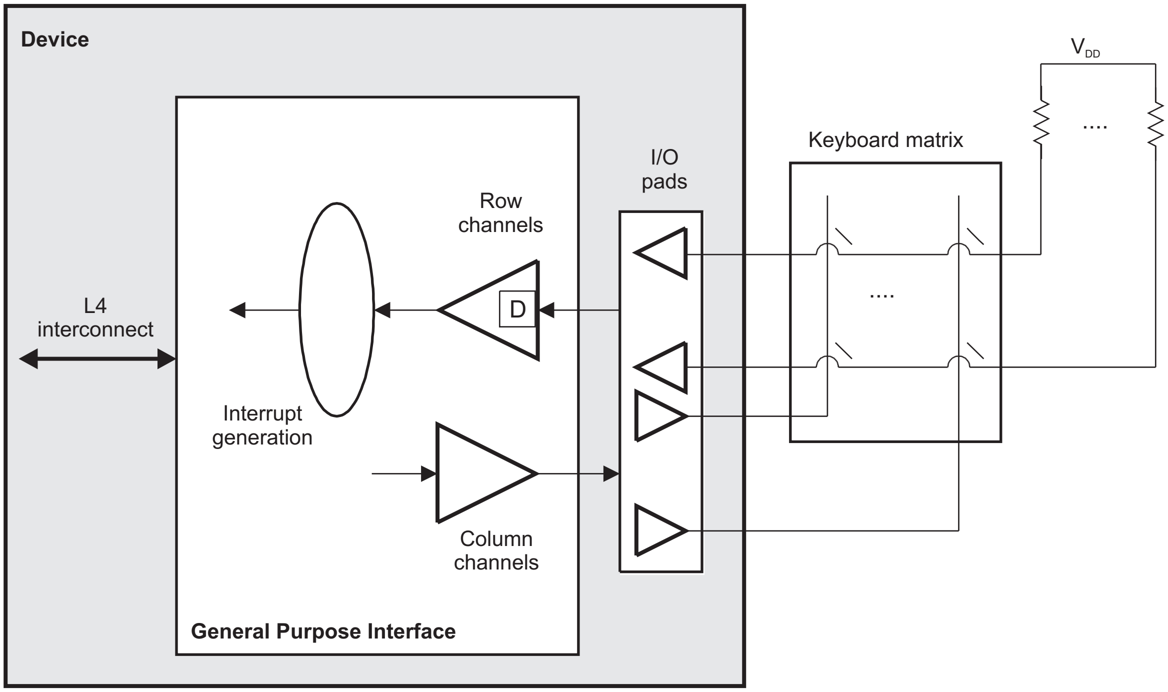
When a keyboard matrix key is pressed, the corresponding row and column lines are shorted together and a low level is driven on the corresponding row channel. This generates an interrupt based on the proper configuration (see Section 28.3.3).

When the keyboard interrupt is received, the processor can disable the keyboard interrupt and scan the column channels for the key coordinates.

- The scanning sequence has as many states as column channels: For each step in the sequence, the processor drives one column channel low and the others high.
- The processor reads the values of the row channels and thus detects which keys in the column are pressed.

At the end of the scanning sequence, the processor establishes which keys are pressed. The keyboard interface can then be reconfigured in the interrupt waiting state.

Figure 28-7. General-Purpose Interface Used as a Keyboard Interface



28.4 GPIO Registers

28.4.1 GPIO Registers

Table 28-5 lists the memory-mapped registers for the GPIO. All register offset addresses not listed in Table 28-5 should be considered as reserved locations and the register contents should not be modified.

Table 28-5. GPIO REGISTERS

Offset	Acronym	Register Name	Section
0h	GPIO_REVISION		Section 28.4.1.1
10h	GPIO_SYSCONFIG		Section 28.4.1.2
20h	GPIO_EOI		Section 28.4.1.3
24h	GPIO_IRQSTS_RAW_0		Section 28.4.1.4
28h	GPIO_IRQSTS_RAW_1		Section 28.4.1.5
2Ch	GPIO_IRQSTS_0		Section 28.4.1.6
30h	GPIO_IRQSTS_1		Section 28.4.1.7
34h	GPIO_IRQSTS_SET_0		Section 28.4.1.8
38h	GPIO_IRQSTS_SET_1		Section 28.4.1.9
3Ch	GPIO_IRQSTS_CLR_0		Section 28.4.1.10
40h	GPIO_IRQSTS_CLR_1		Section 28.4.1.11
44h	GPIO_IRQWAKEN_0		Section 28.4.1.12
48h	GPIO_IRQWAKEN_1		Section 28.4.1.13
114h	GPIO_SYSSTS		Section 28.4.1.14
130h	GPIO_CTRL		Section 28.4.1.15
134h	GPIO_OE		Section 28.4.1.16
138h	GPIO_DATAIN		Section 28.4.1.17
13Ch	GPIO_DATAOUT		Section 28.4.1.18
140h	GPIO_LEVELDETECT0		Section 28.4.1.19
144h	GPIO_LEVELDETECT1		Section 28.4.1.20
148h	GPIO_RISINGDETECT		Section 28.4.1.21
14Ch	GPIO_FALLINGDETECT		Section 28.4.1.22
150h	GPIO_DEBOUNCEN		Section 28.4.1.23
154h	GPIO_DEBOUNCINGTIME		Section 28.4.1.24
190h	GPIO_CLRDATAOUT		Section 28.4.1.25
194h	GPIO_SETDATAOUT		Section 28.4.1.26

28.4.1.1 GPIO_REVISION Register (offset = 0h) [reset = 50600801h]

Register mask: FFFFFFFFh

GPIO_REVISION is shown in Figure 28-8 and described in Table 28-6.

The GPIO revision register is a read only register containing the revision number of the GPIO module. A write to this register has no effect, the same as the reset.

Figure 28-8. GPIO_REVISION Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="2">SCHEME</td><td colspan="2">RESERVED</td><td colspan="4">FUNC</td></tr>
<tr><td colspan="2">R-1h</td><td colspan="2">R-1h</td><td colspan="4">R-60h</td></tr>
</table>

<table>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">FUNC</td></tr>
<tr><td colspan="8">R-60h</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="5">RTL</td><td colspan="3">MAJOR</td></tr>
<tr><td colspan="5">R-1h</td><td colspan="3">R-0h</td></tr>
</table>

<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">CUSTOM</td><td colspan="6">MINOR</td></tr>
<tr><td colspan="2">R-0h</td><td colspan="6">R-1h</td></tr>
</table>

Table 28-6. GPIO_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	Used to distinguish between old Scheme and current.
29-28	RESERVED	R	1h	Reads return 0x1
27-16	FUNC	R	60h	Indicates a software compatible module family
15-11	RTL	R	1h	RTL version
10-8	MAJOR	R	0h	Major Revision
7-6	CUSTOM	R	0h	Indicates a special version for a particular device.
5-0	MINOR	R	1h	Minor Revision

28.4.1.2 GPIO_SYSCONFIG Register (offset = 10h) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_SYSCONFIG is shown in Figure 28-9 and described in Table 28-7.

The GPIO_SYSCONFIG register controls the various parameters of the L4 interconnect. When the AUTOIDLE bit is set, the GPIO_DATAIN read command has a 3 OCP cycle latency due to the data in sample gating mechanism. When the AUTOIDLE bit is not set, the GPIO_DATAIN read command has a 2 OCP cycle latency.

Figure 28-9. GPIO_SYSCONFIG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">RESERVED</td><td colspan="2">IDLEMODE</td><td>ENAWAKEUP</td><td>SOFTRESET</td><td>AUTOIDLE</td></tr>
<tr><td colspan="3">R-0h</td><td colspan="2">R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 28-7. GPIO_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4-3	IDLEMODE	R/W	0h	Power Management, Req/Ack control. 0h (R/W) = Force-idle. An idle request is acknowledged unconditionally. 1h (R/W) = No-idle. An idle request is never acknowledged 2h (R/W) = Smart-idle. Acknowledgment to an idle request is given based on the internal activity of the module 3h (R/W) = Smart Idle Wakeup (GPIO0 only)
2	ENAWAKEUP	R/W	0h	Wakeup control. 0h (R/W) = Wakeup generation is disabled. 1h (R/W) = Wakeup capability is enabled upon expected transition on input GPIO pin
1	SOFTRESET	R/W	0h	Software Reset. Set this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads, it always returns 0. 0h (R/W) = Normal Mode 1h (R/W) = The module is reset.
0	AUTOIDLE	R/W	0h	Internal interface clock gating strategy 0h (R/W) = Internal Interface OCP clock is free-running 1h (R/W) = Automatic internal OCP clock gating, based on the OCP interface activity

28.4.1.3 GPIO_EOI Register (offset = 20h) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_EOI is shown in Figure 28-10 and described in Table 28-8.

This module supports DMA events with its interrupt signal. This register must be written to after the DMA completes in order for subsequent DMA events to be triggered from this module.

Figure 28-10. GPIO_EOI Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							DMAEVT_ACK
R-0h							W-0h

Table 28-8. GPIO_EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	DMAEVT_ACK	W	0h	Write 0 to acknowledge DMA event has been completed. Module will be able to generate another DMA event only when the previous one has been acknowledged using this register. Reads always returns 0. 0h (W) = EOI for interrupt line number 0. Read returns 0. 1h (W) = EOI for interrupt line number 1. Read returns 0.

28.4.1.4 GPIO_IRQSTS_RAW_0 Register (offset = 24h) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_IRQSTS_RAW_0 is shown in Figure 28-11 and described in Table 28-9.

The GPIO_IRQSTS_RAW_0 register provides core status information for the interrupt handling, showing all active events (enabled and not enabled). The fields are read-write. Writing a 1 to a bit sets it to 1, that is, triggers the IRQ (mostly for debug). Writing a 0 has no effect, that is, the register value is not be modified. Only enabled, active events trigger an actual interrupt request on the IRQ output line.

Figure 28-11. GPIO_IRQSTS_RAW_0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">INTLINE_N</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 28-9. GPIO_IRQSTS_RAW_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE_N	R/W	0h	Interrupt n status. 0h (W) = No effect. 1h (W) = IRQ is triggered.

28.4.1.5 GPIO_IRQSTS_RAW_1 Register (offset = 28h) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_IRQSTS_RAW_1 is shown in Figure 28-12 and described in Table 28-10.

The GPIO_IRQSTS_RAW_1 register provides core status information for the interrupt handling, showing all active events (enabled and not enabled). The fields are read-write. Writing a 1 to a bit sets it to 1, that is, triggers the IRQ (mostly for debug). Writing a 0 has no effect, that is, the register value is not be modified. Only enabled, active events trigger an actual interrupt request on the IRQ output line.

Figure 28-12. GPIO_IRQSTS_RAW_1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">INTLINE_N</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 28-10. GPIO_IRQSTS_RAW_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE_N	R/W	0h	Interrupt n status. 0h (W) = No effect. 1h (W) = IRQ is triggered.

28.4.1.6 GPIO_IRQSTS_0 Register (offset = 2Ch) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_IRQSTS_0 is shown in Figure 28-13 and described in Table 28-11.

The GPIO_IRQSTS_0 register provides core status information for the interrupt handling, showing all active events which have been enabled. The fields are read-write. Writing a 1 to a bit clears the bit to 0, that is, clears the IRQ. Writing a 0 has no effect, that is, the register value is not modified. Only enabled, active events trigger an actual interrupt request on the IRQ output line.

Figure 28-13. GPIO_IRQSTS_0 Register

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="32">INTLINE_N</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 28-11. GPIO_IRQSTS_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE_N	R/W	0h	Interrupt n status. 0h (W) = No effect. 0h (R) = IRQ is not triggered. 1h (R) = IRQ is triggered. 1h (W) = Clears the IRQ.

28.4.1.7 GPIO_IRQSTS_1 Register (offset = 30h) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_IRQSTS_1 is shown in Figure 28-14 and described in Table 28-12.

The GPIO_IRQSTS_1 register provides core status information for the interrupt handling, showing all active events which have been enabled. The fields are read-write. Writing a 1 to a bit clears the bit to 0, that is, clears the IRQ. Writing a 0 has no effect, that is, the register value is not modified. Only enabled, active events trigger an actual interrupt request on the IRQ output line.

Figure 28-14. GPIO_IRQSTS_1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">INTLINE_N</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 28-12. GPIO_IRQSTS_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE_N	R/W	0h	Interrupt n status. 0h (W) = No effect. 0h (R) = IRQ is not triggered. 1h (W) = Clears the IRQ. 1h (R) = IRQ is triggered.

28.4.1.8 GPIO_IRQSTS_SET_0 Register (offset = 34h) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_IRQSTS_SET_0 is shown in Figure 28-15 and described in Table 28-13.

All 1-bit fields in the GPIO_IRQSTS_SET_0 register enable a specific interrupt event to trigger an interrupt request. Writing a 1 to a bit enables the interrupt field. Writing a 0 has no effect, that is, the register value is not modified.

Figure 28-15. GPIO_IRQSTS_SET_0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">INTLINE_N</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 28-13. GPIO_IRQSTS_SET_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE_N	R/W	0h	Interrupt n status. 0h = No effect. 1h = Enable IRQ generation.

28.4.1.9 GPIO_IRQSTS_SET_1 Register (offset = 38h) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_IRQSTS_SET_1 is shown in Figure 28-16 and described in Table 28-14.

All 1-bit fields in the GPIO_IRQSTS_SET_1 register enable a specific interrupt event to trigger an interrupt request. Writing a 1 to a bit enables the interrupt field. Writing a 0 has no effect, that is, the register value is not modified.

Figure 28-16. GPIO_IRQSTS_SET_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE_N																															
R/W-0h																															

Table 28-14. GPIO_IRQSTS_SET_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE_N	R/W	0h	Interrupt n status. 0h = No effect. 1h = Enable IRQ generation.

28.4.1.10 GPIO_IRQSTS_CLR_0 Register (offset = 3Ch) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_IRQSTS_CLR_0 is shown in Figure 28-17 and described in Table 28-15.

All 1-bit fields in the GPIO_IRQSTS_CLR_0 register clear a specific interrupt event. Writing a 1 to a bit disables the interrupt field. Writing a 0 has no effect, that is, the register value is not modified.

Figure 28-17. GPIO_IRQSTS_CLR_0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">INTLINE_N</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 28-15. GPIO_IRQSTS_CLR_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE_N	R/W	0h	Interrupt n status. 0h = No effect. 1h = Disable IRQ generation

28.4.1.11 GPIO_IRQSTS_CLR_1 Register (offset = 40h) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_IRQSTS_CLR_1 is shown in Figure 28-18 and described in Table 28-16.

All 1-bit fields in the GPIO_IRQSTS_CLR_1 register clear a specific interrupt event. Writing a 1 to a bit disables the interrupt field. Writing a 0 has no effect, that is, the register value is not modified.

Figure 28-18. GPIO_IRQSTS_CLR_1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">INTLINE_N</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 28-16. GPIO_IRQSTS_CLR_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE_N	R/W	0h	Interrupt n status. 0h = No effect. 1h = Disable IRQ generation.

28.4.1.12 GPIO_IRQWAKEN_0 Register (offset = 44h) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_IRQWAKEN_0 is shown in Figure 28-19 and described in Table 28-17.

Every 1-bit field in the GPIO_IRQWAKEN_0 register enables a specific (synchronous) IRQ request source to generate an asynchronous wakeup (on the appropriate swakeup line). This register allows the user to mask the expected transition on input GPIO from generating a wakeup request. The GPIO_IRQWAKEN_0 is programmed synchronously with the interface clock before any Idle mode request coming from the host processor. Note: In Force-Idle mode, the module wake-up feature is totally inhibited. The wake-up generation can also be gated at module level using the EnaWakeUp bit from GPIO_SYSCONFIG register.

Figure 28-19. GPIO_IRQWAKEN_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															
R/W-0h																															

Table 28-17. GPIO_IRQWAKEN_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE	R/W	0h	Wakeup Set for Interrupt Line 0h = Disable wakeup generation. 1h = Enable wakeup generation.

28.4.1.13 GPIO_IRQWAKEN_1 Register (offset = 48h) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_IRQWAKEN_1 is shown in Figure 28-20 and described in Table 28-18.

Every 1-bit field in the GPIO_IRQWAKEN_1 register enables a specific (synchronous) IRQ request source to generate an asynchronous wakeup (on the appropriate swakeup line). This register allows the user to mask the expected transition on input GPIO from generating a wakeup request. The GPIO_IRQWAKEN_1 is programmed synchronously with the interface clock before any Idle mode request coming from the host processor. Note: In Force-Idle mode, the module wake-up feature is totally inhibited. The wake-up generation can also be gated at module level using the EnaWakeUp bit from GPIO_SYSCONFIG register.

Figure 28-20. GPIO_IRQWAKEN_1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">INTLINE</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 28-18. GPIO_IRQWAKEN_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE	R/W	0h	Wakeup Set for Interrupt Line 0h = Disable wakeup generation. 1h = Enable wakeup generation.

28.4.1.14 GPIO_SYSSTS Register (offset = 114h) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_SYSSTS is shown in Figure 28-21 and described in Table 28-19.

The GPIO_SYSSTS register provides the reset status information about the GPIO module. It is a read-only register; a write to this register has no effect.

Figure 28-21. GPIO_SYSSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED							RESETDONE
R-0h							R-0h

Table 28-19. GPIO_SYSSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	RESETDONE	R	0h	Reset status information. 0h (R) = Internal Reset is on-going 1h (R) = Reset completed

28.4.1.15 GPIO_CTRL Register (offset = 130h) [reset = 2h]

Register mask: FFFFFFFFh

GPIO_CTRL is shown in Figure 28-22 and described in Table 28-20.

The GPIO_CTRL register controls the clock gating functionality. The DISABLEMODULE bit controls a clock gating feature at the module level. When set, this bit forces the clock gating for all internal clock paths. Module internal activity is suspended. System interface is not affected by this bit. System interface clock gating is controlled with the AUTOIDLE bit in the system configuration register (GPIO_SYSCONFIG). This bit is to be used for power saving when the module is not used because of the multiplexing configuration selected at the chip level. This bit has precedence over all other internal configuration bits.

Figure 28-22. GPIO_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED							
R-0h							

7	6	5	4	3	2	1	0
RESERVED					GATINGRATIO		DISABLEMODULE
R-0h					R/W-1h		R/W-0h

Table 28-20. GPIO_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-1	GATINGRATIO	R/W	1h	Gating Ratio. Controls the clock gating for the event detection logic. 0h = Functional clock is interface clock. 1h = Functional clock is interface clock divided by 2. 2h = Functional clock is interface clock divided by 4. 3h = Functional clock is interface clock divided by 8.
0	DISABLEMODULE	R/W	0h	Module Disable 0h = Module is enabled, clocks are not gated. 1h = Module is disabled, internal clocks are gated

28.4.1.16 GPIO_OE Register (offset = 134h) [reset = FFFFFFFFh]

Register mask: FFFFFFFFh

GPIO_OE is shown in Figure 28-23 and described in Table 28-21.

The GPIO_OE register is used to enable the pins output capabilities. At reset, all the GPIO related pins are configured as input and output capabilities are disabled. This register is not used within the module, its only function is to carry the pads configuration. When the application is using a pin as an output and does not want interrupt generation from this pin, the application can/has to properly configure the Interrupt Enable registers.

Figure 28-23. GPIO_OE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUTPUTEN_N																															
R/W-FFFFFFFFh																															

Table 28-21. GPIO_OE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	OUTPUTEN_N	R/W	FFFFFFFFh	Output Data Enable 0h = The corresponding GPIO port is configured as an output. 1h = The corresponding GPIO port is configured as an input.

28.4.1.17 GPIO_DATAIN Register (offset = 138h) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_DATAIN is shown in Figure 28-24 and described in Table 28-22.

The GPIO_DATAIN register is used to register the data that is read from the GPIO pins. The GPIO_DATAIN register is a read-only register. The input data is sampled synchronously with the interface clock and then captured in the GPIO_DATAIN register synchronously with the interface clock. So, after changing, GPIO pin levels are captured into this register after two interface clock cycles (the required cycles to synchronize and to write the data). When the AUTOIDLE bit in the system configuration register (GPIO_SYSCONFIG) is set, the GPIO_DATAIN read command has a 3 OCP cycle latency due to the data in sample gating mechanism. When the AUTOIDLE bit is not set, the GPIO_DATAIN read command has a 2 OCP cycle latency.

Figure 28-24. GPIO_DATAIN Register

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="32">DATAIN</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 28-22. GPIO_DATAIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATAIN	R	0h	Sampled input data

28.4.1.18 GPIO_DATAOUT Register (offset = 13Ch) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_DATAOUT is shown in Figure 28-25 and described in Table 28-23.

The GPIO_DATAOUT register is used for setting the value of the GPIO output pins. Data is written to the GPIO_DATAOUT register synchronously with the interface clock. This register can be accessed with direct read/write operations or using the alternate Set/Clear feature. This feature enables to set or clear specific bits of this register with a single write access to the set data output register (GPIO_SETDATAOUT) or to the clear data output register (GPIO_CLEARDATAOUT) address.

Figure 28-25. GPIO_DATAOUT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">DATAOUT</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 28-23. GPIO_DATAOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATAOUT	R/W	0h	Data to set on output pins

28.4.1.19 GPIO_LEVELDETECT0 Register (offset = 140h) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_LEVELDETECT0 is shown in Figure 28-26 and described in Table 28-24.

The GPIO_LEVELDETECT0 register is used to enable/disable for each input lines the low-level (0) detection to be used for the interrupt request generation. Enabling at the same time high-level detection and low-level detection for one given pin makes a constant interrupt generator.

Figure 28-26. GPIO_LEVELDETECT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LEVELDETECT0_N																															
R/W-0h																															

Table 28-24. GPIO_LEVELDETECT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LEVELDETECT0_N	R/W	0h	Low Level Interrupt Enable 0h = Disable the IRQ assertion on low-level detect. 1h = Enable the IRQ assertion on low-level detect.

28.4.1.20 GPIO_LEVELDETECT1 Register (offset = 144h) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_LEVELDETECT1 is shown in Figure 28-27 and described in Table 28-25.

The GPIO_LEVELDETECT1 register is used to enable/disable for each input lines the high-level (1) detection to be used for the interrupt request generation. Enabling at the same time high-level detection and low-level detection for one given pin makes a constant interrupt generator.

Figure 28-27. GPIO_LEVELDETECT1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">LEVELDETECT1_N</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 28-25. GPIO_LEVELDETECT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LEVELDETECT1_N	R/W	0h	High Level Interrupt Enable 0h = Disable the IRQ assertion on high-level detect. 1h = Enable the IRQ assertion on high-level detect.

28.4.1.21 GPIO_RISINGDETECT Register (offset = 148h) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_RISINGDETECT is shown in Figure 28-28 and described in Table 28-26.

The GPIO_RISINGDETECT register is used to enable/disable for each input lines the rising-edge (transition 0 to 1) detection to be used for the interrupt request generation.

Figure 28-28. GPIO_RISINGDETECT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RISINGDETECT_N																															
R/W-0h																															

Table 28-26. GPIO_RISINGDETECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RISINGDETECT_N	R/W	0h	Rising Edge Interrupt Enable 0h = Disable the IRQ on rising-edge detect. 1h = Enable the IRQ on rising-edge detect.

28.4.1.22 GPIO_FALLINGDETECT Register (offset = 14Ch) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_FALLINGDETECT is shown in Figure 28-29 and described in Table 28-27.

The GPIO_FALLINGDETECT register is used to enable/disable for each input lines the falling-edge (transition 1 to 0) detection to be used for the interrupt request generation.

Figure 28-29. GPIO_FALLINGDETECT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FALLINGDETECT_N																															
R/W-0h																															

Table 28-27. GPIO_FALLINGDETECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FALLINGDETECT_N	R/W	0h	Falling Edge Interrupt Enable 0h = Disable the IRQ on falling-edge detect. 1h = Enable the IRQ on falling-edge detect.

28.4.1.23 GPIO_DEBOUNCEN Register (offset = 150h) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_DEBOUNCEN is shown in Figure 28-30 and described in Table 28-28.

The GPIO_DEBOUNCEN register is used to enable/disable the debouncing feature for each input line.

Figure 28-30. GPIO_DEBOUNCEN Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

DEBOUNCEEN_N

R/W-0h

Table 28-28. GPIO_DEBOUNCEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBOUNCEEN_N	R/W	0h	Input Debounce Enable 0h = Disable debouncing feature on the corresponding input port. 1h = Enable debouncing feature on the corresponding input port.

28.4.1.24 GPIO_DEBOUNCINGTIME Register (offset = 154h) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_DEBOUNCINGTIME is shown in Figure 28-31 and described in Table 28-29.

The GPIO_DEBOUNCINGTIME register controls debouncing time (the value is global for all ports). The debouncing cell is running with the debouncing clock (32 kHz), this register represents the number of the clock cycle(s) (31 s long) to be used.

Figure 28-31. GPIO_DEBOUNCINGTIME Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DEBOUNCETIME							
R-0h								R/W-0h							

Table 28-29. GPIO_DEBOUNCINGTIME Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved.
7-0	DEBOUNCETIME	R/W	0h	Input Debouncing Value in 31 microsecond steps. Debouncing Value = (DEBOUNCETIME + 1) * 31 microseconds.

28.4.1.25 GPIO_CLRDATAOUT Register (offset = 190h) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_CLRDATAOUT is shown in Figure 28-32 and described in Table 28-30.

Writing a 1 to a bit in the GPIO_CLRDATAOUT register clears to 0 the corresponding bit in the GPIO_DATAOUT register; writing a 0 has no effect. A read of the GPIO_CLRDATAOUT register returns the value of the data output register (GPIO_DATAOUT).

Figure 28-32. GPIO_CLRDATAOUT Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

INTLINE_N

R/W-0h

Table 28-30. GPIO_CLRDATAOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE_N	R/W	0h	Clear Data Output Register 0h = No effect. 1h = Clear the corresponding bit in the GPIO_DATAOUT register.

28.4.1.26 GPIO_SETDATAOUT Register (offset = 194h) [reset = 0h]

Register mask: FFFFFFFFh

GPIO_SETDATAOUT is shown in Figure 28-33 and described in Table 28-31.

Writing a 1 to a bit in the GPIO_SETDATAOUT register sets to 1 the corresponding bit in the GPIO_DATAOUT register; writing a 0 has no effect. A read of the GPIO_SETDATAOUT register returns the value of the data output register (GPIO_DATAOUT).

Figure 28-33. GPIO_SETDATAOUT Register

31–0
INTLINE_N
R/W-0h

Table 28-31. GPIO_SETDATAOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE_N	R/W	0h	Set Data Output Register 0h = No effect. 1h = Set the corresponding bit in the GPIO_DATAOUT register.

Graphics Accelerator (SGX)

This chapter describes the graphics accelerator for the device.

NOTE: The Graphics Accelerator (SGX) is not supported for the AMIC120.

Topic **Page**

29.1 Introduction

This chapter describes the 2D/3D graphics accelerator (SGX) for the device.

NOTE: The SGX subsystem is a Texas Instruments instantiation of the POWERVR® SGX530 core from Imagination Technologies Ltd.

This document contains materials that are ©2003-2007 Imagination Technologies Ltd.

POWERVR® and USSE™ are trademarks or registered trademarks of Imagination Technologies Ltd.

The 2D/3D graphics accelerator (SGX) subsystem accelerates 2-dimensional (2D) and 3-dimensional (3D) graphics applications. The SGX subsystem is based on the POWERVR® SGX core from Imagination Technologies. SGX is a new generation of programmable POWERVR graphic cores. The POWERVR SGX530 v1.2.5 architecture is scalable and can target market segments, from portable devices to HMI.

29.1.1 POWERVR SGX Main Features

- 2D and 3D graphics
- Tile-based architecture
- Universal scalable shader engine (USSE™) – multithreaded engine incorporating pixel and vertex shader functionality
- Advanced shader feature set: in excess of OpenGL2.0
- Industry-standard API support: OpenGL ES 1.1 and 2.0, OpenVG v1.0.1
- Fine-grained task switching, load balancing, and power management
- Advanced geometry direct memory access (DMA) driven operation for minimum CPU interaction
- Programmable high-quality image anti-aliasing
- POWERVR SGX core MMU for address translation from the core virtual address to the external physical address (up to 4GB address range)
- Fully virtualized memory addressing for OS operation in a unified memory architecture
- Advanced and standard 2D operations [e.g., vector graphics, BLTs (block level transfers), ROPs (raster operations)]
- 32K stride support

29.1.2 SGX 3D Features

- Deferred pixel shading
- On-chip tile floating point depth buffer
- 8-bit stencil with on-chip tile stencil buffer
- 8 parallel depth/stencil tests per clock
- Scissor test
- Texture support:
  - Cube map
  - Projected textures
  - 2D textures
  - Nonsquare textures
- Texture formats:
  - RGBA 8888, 565, 1555
  - Monochromatic 8, 16, 16f, 32f, 32int
  - Dual channel, 8:8, 16:16, 16f:16f
  - Compressed textures PVR-TC1, PVR-TC2, ETC1
  - Programmable support for YUV 420 and 422 formats for YUV/RGB color conversion
- Resolution support:
  - Frame buffer maximum size = 2048 x 2048
  - Texture maximum size = 2048 x 2048
- Texture filtering:
  - Bilinear, trilinear, anisotropic
  - Independent minimum and maximum control
- Antialiasing:
  - 4x multisampling
  - Up to 16x full scene anti-aliasing
  - Programmable sample positions
- Indexed primitive list support
  - Bus mastered
- Programmable vertex DMA
- Render to texture:
  - Including twiddled formats
  - Auto MipMap generation
- Multiple on-chip render targets (MRT).
  Note: Performance is limited when the on-chip memory is not available.

29.1.3 Universal Scalable Shader Engine (USSE) – Key Features

The USSE is the engine core of the POWERVR SGX architecture and supports a broad range of instructions.

- Single programming model:
  - Multithreaded with 16 simultaneous execution threads and up to 64 simultaneous data instances
  - Zero-cost swapping in, and out, of threads
  - Cached program execution model
  - Dedicated pixel processing instructions
  - Dedicated video encode/decode instructions
- SIMD execution unit supporting operations in:
  - 32-bit IEEE float
  - 2-way 16-bit fixed point
  - 4-way 8-bit integer
  - 32-bit bit-wise (logical only)
- Static and dynamic flow control:
  - Subroutine calls
  - Loops
  - Conditional branches
  - Zero-cost instruction predication
- Procedural geometry:
  - Allows generation of primitives
  - Effective geometry compression
  - High-order surface support
- External data access:
  - Permits reads from main memory using cache
  - Permits writes to main memory
  - Data fence facility
  - Dependent texture reads

29.1.4 Unsupported Features

There are no unsupported SGX530 features for this device.

29.2 Integration

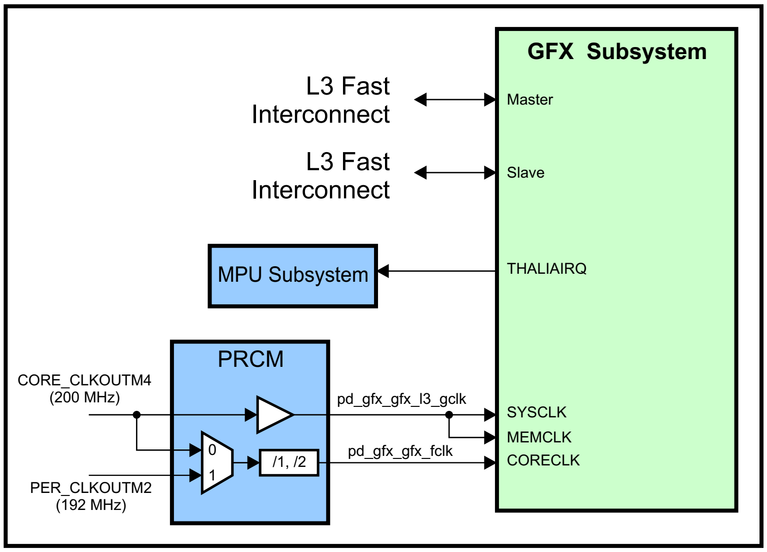


Figure 29-1. SGX530 Integration

29.2.1 SGX530 Connectivity Attributes

The general connectivity attributes of the SGX530 are shown in the following table.

Table 29-1. SGX530 Connectivity Attributes

Attributes	Type
Power domain	GFX Domain
Clock domain	SGX_CLK
Reset signals	SGX_RST
Idle/Wakeup signals	Smart Idle Initiator Standby
Interrupt request	THALIAIRQ (GFXINT) to MPU Subsystem
DMA request	None
Physical address	L3 Fast slave port

29.2.2 SGX530 Clock and Reset Management

The SGX530 uses separate functional and interface clocks. The SYSCLK is the clock for the slave interface and runs at the L3F frequency. The MEMCLK is the clock for the memories and master interface and also runs at the L3F frequency. The CORECLK is the functional clock. It can be sourced from either the L3F clock (CORE_CLKOUTM4) or from the 192 MHz PER_CLKOUTM2 and can optionally be divided by 2.

Table 29-2. SGX530 Clock Signals

Clock signal	Max Freq	Reference / Source	Comments
SYSCLK Interface clock	200 MHz	CORE_CLKOUTM4	pd_gfx_gfx_l3_gclk From PRCM
MEMCLK Memory Clock	200 MHz	CORE_CLKOUTM4	pd_gfx_gfx_l3_gclk From PRCM
CORECLK Functional clock	200 MHz	PER_CLKOUTM2 or CORE_CLKOUTM4	pd_gfx_gfx_fclk From PRCM

29.2.3 SGX530 Pin List

The SGX530 module does not include any external interface pins.

29.3 Functional Description

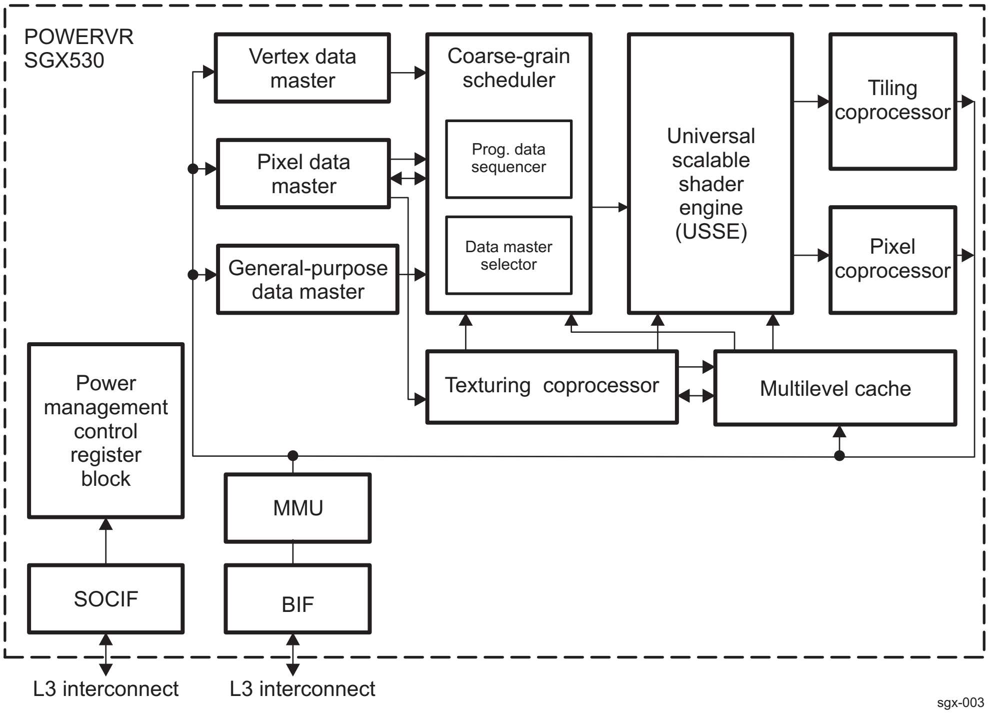
29.3.1 SGX Block Diagram

The SGX subsystem is based on the POWERVR® SGX530 core from Imagination Technologies. The architecture uses programmable and hard coded pipelines to perform various processing tasks required in 2D, 3D, and video processing. The SGX architecture comprises the following elements:

- Coarse grain scheduler
  - Programmable data sequencer (PDS)
  - Data master selector (DMS)
- Vertex data master (VDM)
- Pixel data master (PDM)
- General-purpose data master
- USSE
- Tiling coprocessor
- Pixel coprocessor
- Texturing coprocessor
- Multilevel cache

Figure 29-2 shows a block diagram of the SGX cores.

Figure 29-2. SGX Block Diagram



29.3.2 SGX Elements Description

The coarse grain scheduler (CGS) is the main system controller for the POWERVR SGX architecture. It consists of two stages, the DMS and the PDS. The DMS processes requests from the data masters and determines which tasks can be executed given the resource requirements. The PDS then controls the loading and processing of data on the USSE.

There are three data masters in the SGX core:

- The VDM is the initiator of transform and lighting processing within the system. The VDM reads an input control stream, which contains triangle index data and state data. The state data indicates the PDS program, size of the vertices, and the amount of USSE output buffer resource available to the VDM. The triangle data is parsed to determine unique indices that must be processed by the USSE. These are grouped together according to the configuration provided by the driver and presented to the DMS.

- The PDM is the initiator of rasterization processing within the system. Each pixel pipeline processes pixels for a different half of a given tile, which allows for optimum efficiency within each pipe due to locality of data. It determines the amount of resource required within the USSE for each task. It merges this with the state address and issues a request to the DMS for execution on the USSE.
- The general-purpose data master responds to events within the system (such as end of a pass of triangles from the ISP, end of a tile from the ISP, end of render, or parameter stream breakpoint event). Each event causes either an interrupt to the host or synchronized execution of a program on the PDS. The program may, or may not cause a subsequent task to be executed on the USSE.

The USSE is a user-programmable processing unit. Although general in nature, its instructions and features are optimized for three types of task: processing vertices (vertex shading), processing pixels (pixel shading), and video/imaging processing.

The multilevel cache is a 2-level cache consisting of two modules: the main cache and the mux/arbiter/demux/decompression unit (MADD). The MADD is a wrapper around the main cache module designed to manage and format requests to and from the cache, as well as providing Level 0 caching for texture and USSE requests. The MADD can accept requests from the PDS, USSE, and texture address generator modules. Arbitration, as well as any required texture decompression, are performed between the three data streams.

The texturing coprocessor performs texture address generation and formatting of texture data. It receives requests from either the iterators or USSE modules and translates these into requests in the multilevel cache. Data returned from the cache are then formatted according to the texture format selected, and sent to the USSE for pixel-shading operations.

To process pixels in a tiled manner, the screen is divided into tiles and arranged as groups of tiles by the tiling coprocessor. An inherent advantage of tiling architecture is that a large amount of vertex data can be rejected at this stage, thus reducing the memory storage requirements and the amount of pixel processing to be performed.

The pixel coprocessor is the final stage of the pixel-processing pipeline and controls the format of the final pixel data sent to the memory. It supplies the USSE with an address into the output buffer and then USSE returns the relevant pixel data. The address order is determined by the frame buffer mode. The pixel coprocessor contains a dithering and packing function.

Programmable Real-Time Unit Subsystem and Industrial Communication Subsystem (PRU-ICSS)

This chapter describes the PRU-ICSS for the device.

Topic **Page**

30.1 Introduction

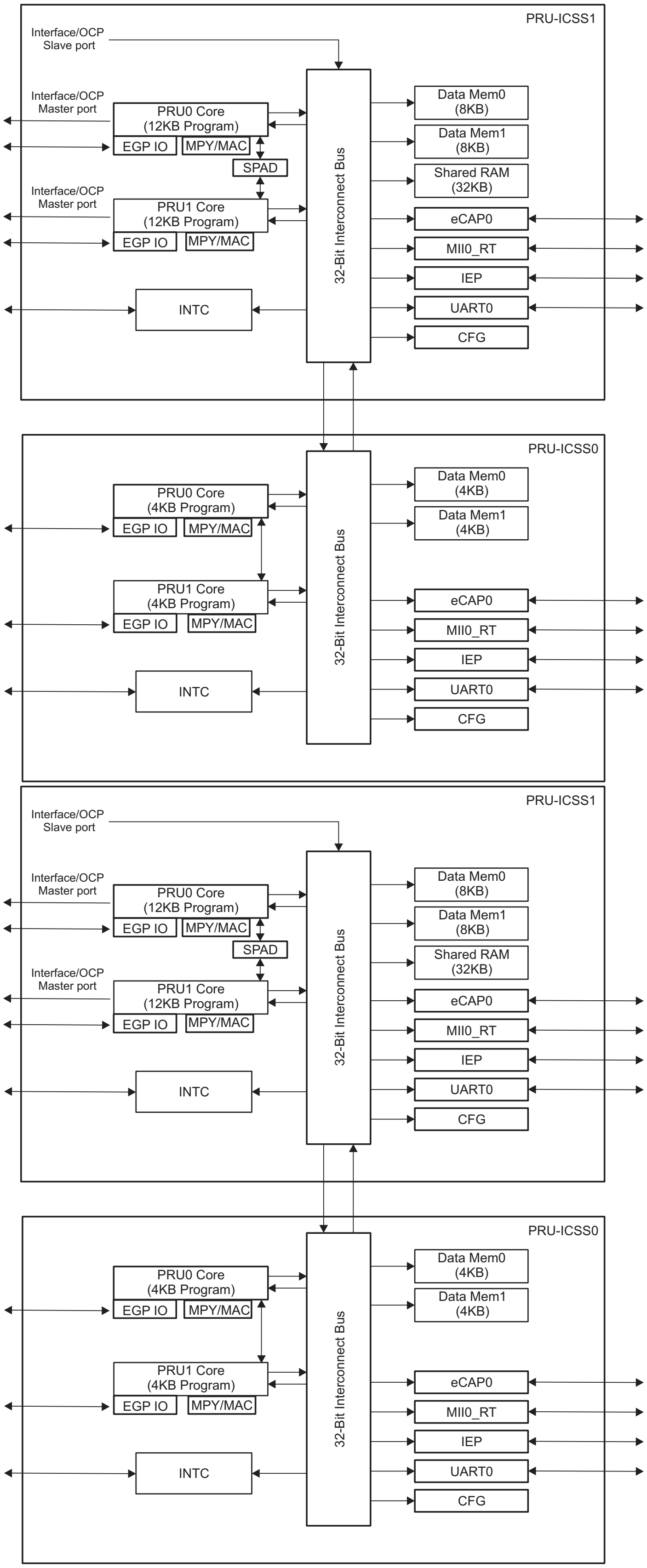
The Programmable Real-Time Unit Subsystem and Industrial Communication Subsystem (PRU-ICSS) consists of dual 32-bit RISC cores (Programmable Real-Time Units, or PRUs), shared, data, and instruction memories, internal peripheral modules, and an interrupt controller (INTC). The programmable nature of the PRU-ICSS, along with their access to pins, events and all SoC resources, provides flexibility in implementing fast real-time responses, specialized data handling operations, custom peripheral interfaces, and in offloading tasks from the other processor cores of the system-on-chip (SoC).

This device contains two subsystems: PRU-ICSS1 and PRU-ICSS0. PRU-ICSS1 is considered a superset of PRU-ICSS0. Figure 30-1 details PRU-ICSS1 and PRU-ICSS0.

The PRU cores within the subsystems have access to all resources on the SoC through the Interface/OCP Master port, and the external host processors can access the PRU-ICSS resources through the Interface/OCP Slave port. The 32-bit interconnect bus connects the various internal and external masters to the resources inside the PRU-ICSS. The INTC handles system input events and posts events back to the device-level host CPU.

The PRU cores are programmed with a small, deterministic instruction set. Each PRU can operate independently or in coordination with each other and can also work in coordination with the device-level host CPU. This interaction between processors is determined by the nature of the firmware loaded into the PRU's instruction memories.

Figure 30-1. PRU-ICSS Block Diagram



30.1.1 Features

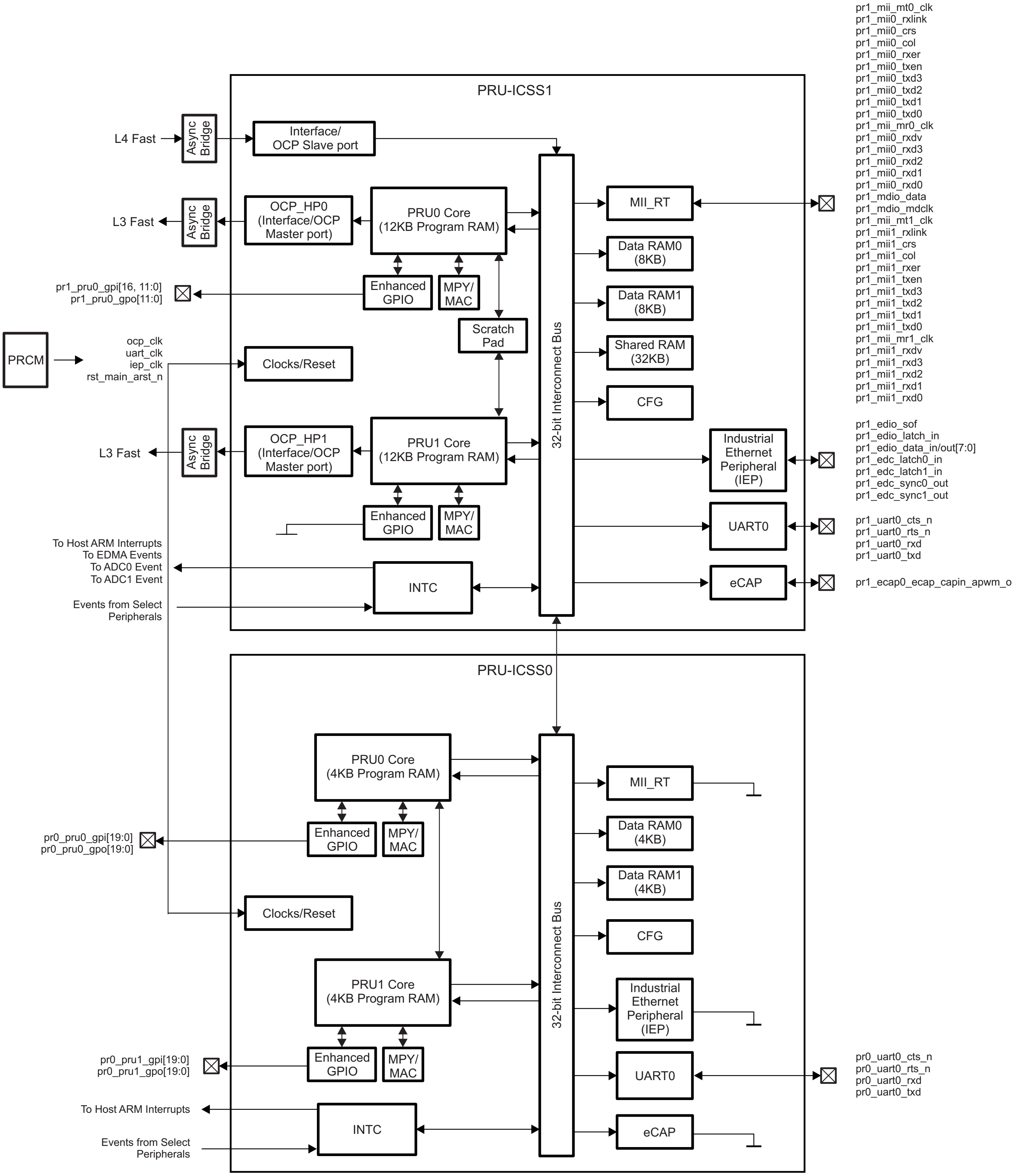
The PRU-ICSS includes the following main features:

- Two PRUs each with:
  - Program memory (PRU-ICSS1 - 12KB, PRU-ICSS0 - 4KB)
  - Data memory (PRU-ICSS1 - 8KB, PRU-ICSS0 - 4KB)
  - High Performance Interface/OCP Master port for accessing external memories[1]
  - Enhanced GPIO (EGPIO) with async capture, serial, 9ch Sigma Delta[2], and 3ch EnDat 2.2 support
  - Multiplier with optional accumulation (MPY/MAC)
- One scratch pad (SPAD) memory (only PRU-ICSS1)
  - 3 Banks of 30 32-bit registers
- Broadside direct connect between PRU cores within subsystem
- 32KB general purpose shared memory (only PRU-ICSS1)
- One Interrupt Controller (INTC)
  - Up to 64 input events supported
  - Interrupt mapping to 10 interrupt channels
  - 10 Host interrupts (2 to PRU0 and PRU1, 1 to PRU-ICSS1 and PRU-ICSS0, 7 output to chip level)
  - Each system event can be enabled and disabled
  - Each host event can be enabled and disabled
  - Hardware prioritization of events
- 16 software events generated by 2 PRUs
- One Ethernet MII_RT module with two MII ports and configurable connections to PRUs[3][4]
- One Industrial Ethernet Peripheral (IEP) to manage/generate Industrial Ethernet functions[3]
  - One Industrial Ethernet timer with 10 capture[4] and 16 compare events
  - Two Industrial Ethernet sync signals and two latch signals[4]
  - Two Industrial Ethernet 16-bit watchdog timers[4]
  - Industrial Ethernet digital IOs
- One 16550-compatible UART with a dedicated 192-MHz clock, supporting up to 12Mbaud for PROFIBUS DP[3]
- One Enhanced Capture Module (ECAP)[3]
- Flexible power management support
- Integrated 32-bit interconnect bus for connecting the various internal and external masters to the resources inside the PRU-ICSS
- Interface/OCP Slave port for external masters to access PRU-ICSS memories[1]
- Optional address translation for PRU transaction to External Host[1]
- All memories within the PRU-ICSS support parity

30.2 Integration

The device includes two Programmable Real-Time Unit Subsystem and Industrial Communication Subsystem (PRU-ICSS), each consisting of two independent Programmable Real-time Units (PRUs). Each PRU is a 32-bit Load/Store RISC processor with dedicated memories. The PRU-ICSS integration is shown in Figure 30-2.

Figure 30-2. PRU-ICSS Integration



For the availability of all features, see the device features in Chapter 1, *Introduction* SPRUHL7.

30.2.1 PRU-ICSS Connectivity Attributes

The general connectivity attributes for the PRU subsystem are shown in Table 30-1.

Table 30-1. PRU-ICSS Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_PRU_ICSS_OCP_GCLK (OCP clock) PD_PER_PRU_ICSS_IEP_GCLK (IEP clock) PD_PER_PRU_ICSS_UART_GCLK (UART clock)
Reset Signals	PRU_ICSS_LRST_N
Idle/Wakeup Signals	Standby Idle
Interrupt Requests	14 Interrupts pr1_host_intr[7:6](1) to MPU Subsystem pr1_host_intr[4:1](1) to MPU Subsystem pr1_host_intr[0](1) to MPU Subsystem, ADC0, and ADC1 pr0_host_intr[7:6](1) to MPU Subsystem pr0_host_intr[4:0](1) to MPU Subsystem
DMA Requests	No dedicated DMA events but pr1_host_intr[7:6](1) interrupt outputs also connected as DMA events
Physical Address	L3 Fast Slave Port

(1) pr<*k*>_host_intr[0:7] corresponds to Host-2 to Host-9 of the PRU-ICSS<*k*> interrupt controller.

30.2.2 PRU-ICSS Clock and Reset Management

The PRU-ICSS module uses the following functional and OCP interface clocks.

Table 30-2. PRU-ICSS Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
ocp_clk Interface Clock	225 MHz	CORE_CLKOUTM4 or Display PLL CLKOUT	pd_per_pru_icss_ocp_gclk from PRCM Clocks both L3 master and L3F slave
uart_clk Functional Clock	192 MHz	PER_CLKOUTM2	pd_per_pru_icss_uart_gclk from PRCM UART Clock
iep_clk Functional Clock	200 MHz	CORE_CLKOUTM4	pd_per_pru_icss_iep_gclk from PRCM Industrial Ethernet Peripheral Clock

NOTE: Default frequency of ocp_clk is 200 MHz using CORE_CLKOUTM4 reference clock. The max frequency 225 MHz is achievable through Display PLL CLKOUT. Refer to Section 13.2 for DSS limitations when configuring this PLL for frequencies >200 MHz.

30.2.3 PRU-ICSS Pin List

The PRU-ICSS external interface signals are shown in Table 30-3. The PRU GPI/GPO pin function depends on the operation mode. Refer to Table 30-3 for a complete list of pin functions for each mode.

Table 30-3. PRU-ICSS Pin List

Pin	Type	Description
PRU-ICSS1		
pr1_mii_mr0_clk	I	MII0 Receive Clock
pr1_mii0_rxdv	I	MII0 Receive Data Valid
pr1_mii0_rxd[3:0]	I	MII0 Receive Data
pr1_mii0_rxlink	I	MII0 Receive Link
pr1_mii0_rxer	I	MII0 Receive Data Error
pr1_mii0_crs	I	MII0 Carrier Sense
pr1_mii0_col	I	MII0 Carrier Sense
pr1_mii_mt0_clk	I	MII0 Transmit Clock
pr1_mii0_txen	O	MII0 Transmit Enable
pr1_mii0_txd[3:0]	O	MII0 Transmit Data
pr1_mii_mr1_clk	I	MII1 Receive Clock
pr1_mii1_rxdv	I	MII1 Receive Data Valid
pr1_mii1_rxd[3:0]	I	MII1 Receive Data
pr1_mii1_rxlink	I	MII1 Receive Link
pr1_mii1_rxer	I	MII1 Receive Data Error
pr1_mii1_crs	I	MII1 Carrier Sense
pr1_mii1_col	I	MII1 Carrier Sense
pr1_mii_mt1_clk	I	MII1 Transmit Clock
pr1_mii1_txen	O	MII1 Transmit Enable
pr1_mii1_txd[3:0]	O	MII1 Transmit Data
pr1_mdio_mdclk	O	MDIO Clk
pr1_mdio_data	I/O	MDIO Data
pr1_edio_sof	O	ECAT Digital I/O Start of Frame
pr1_edio_outvalid	O	ECAT Digital I/O Output Valid
pr1_edio_latch_in	I	ECAT Digital I/O Latch In
pr1_edio_data_in[7:0]	I	ECAT Digital I/O Data In
pr1_edio_data_out[7:0]	O	ECAT Digital I/O Data Out
pr1_edc_sync0_out	O	ECAT Distributed Clock Sync Out
pr1_edc_sync1_out	O	ECAT Distributed Clock Sync Out
pr1_edc_latch0_in	I	ECAT Distributed Clock Latch In
pr1_edc_latch1_in	I	ECAT Distributed Clock Latch In
pr1_uart0_cts_n	I	UART Clear to Send
pr1_uart0_rts_n	O	UART Request to Send
pr1_uart0_rxd	I	UART Receive Data
pr1_uart0_txd	O	UART Transmit Data
pr1_ecap0_ecap_capin_apwm_o	IO	Enhanced capture (ECAP) input or Auxiliary PWM out
pr1_pru0_gpo[11:0]	IO	PRU-ICSS1 PRU0 Register R30 (GPO) Outputs
pr1_pru0_gpi[11:0]	IO	PRU-ICSS1 PRU0 Register R31 (GPI) Inputs
pr1_pru0_gpi[16]	IO	PRU-ICSS1 PRU0 Register R31 (GPI) Input

Table 30-3. PRU-ICSS Pin List (continued)

Pin	Type	Description
	PRU-ICSS0	
pr0_uart0_cts_n	I	UART Clear to Send
pr0_uart0_rts_n	O	UART Request to Send
pr0_uart0_rxd	I	UART Receive Data
pr0_uart0_txd	O	UART Transmit Data
pr0_pru0_gpo[19:0]	IO	PRU-ICSS0 PRU0 Register R30 (GPO) Outputs
pr0_pru0_gpi[19:0]	IO	PRU-ICSS0 PRU0 Register R31 (GPI) Inputs
pr0_pru1_gpo[19:0]	IO	PRU-ICSS0 PRU1 Register R30 (GPO) Outputs
pr0_pru1_gpi[19:0]	IO	PRU-ICSS0 PRU1 Register R31 (GPI) Inputs

Note: R30 may not be initialized after reset. To avoid outside effect, R30 should be initialized before PINMUX configuration.

30.3 PRU-ICSS Memory Map Overview

The PRU-ICSS comprises various distinct addressable regions that are mapped to both a local and global memory map. The local memory maps are maps with respect to the PRU point of view. Each PRU-ICSS can also access the memories within the other subsystem without going through an external port, as described in Section 30.3.1.2.The global memory maps are maps with respect to the Host point of view, but can also be accessed by the PRU-ICSS. All global or external accesses are routed through PRU-ICSS1. PRU-ICSS0 does not have direct access to memories outside the subsystem.

30.3.1 Local Memory Map

The PRU-ICSS memory map is documented in Table 30-4 (Instruction Space) and in Table 30-5 (Data Space). Note that these two memory maps are implemented inside the PRU-ICSS and are local to the components of the PRU-ICSS.

30.3.1.1 Local Instruction Memory Map

Each PRU has a dedicated Instruction Memory (8KB12KB for PRU-ICSS1, 4KB for PRU-ICSS0) which needs to be initialized by a Host processor before the PRU executes instructions. This region is only accessible to masters via the interface/ OCP slave port when the PRU is not running.

Table 30-4. Local Instruction Memory Map

Start Address	PRU-ICSS1		PRU-ICSS0	
	PRU0	PRU1	PRU0	PRU1
0x0000_0000	12KB IRAM	12KB IRAM	4KB IRAM	4KB IRAM

30.3.1.2 Local Data Memory Map

The local data memory map in Table 30-5 allows each PRU core to access the PRU-ICSS addressable regions (both its own subsystem and the other subsystem) and the external host’s memory map.

The PRU accesses the other PRU-ICSS memory map through an expansion port starting at address 0x0004_0000. The address seen by the other PRU-ICSS will be translated by hardware, subtracting 0x0004_0000.

The PRU accesses the external Host memory map through the Interface/OCP Master port (System OCP_HP0/1) starting at address 0x0008_0000. By default, memory addresses between 0x0000_0000 – 0x0007_FFFF will correspond to the PRU-ICSS local address in Table 30-5. To access an address between 0x0000_0000–0x0007_FFFF of the external Host map, the address offset of –0x0008_0000 feature is enabled through the PMAO register of the PRU-ICSS CFG register space. Note that all external accesses to or from PRU-ICSS0 are routed through PRU-ICSS1.

Table 30-5. Local Data Memory Map

Start Address	PRU-ICSS1		PRU-ICSS0	
	PRU0	PRU1	PRU0	PRU1
0x0000_0000	PRU-ICSS1 Data 8KB RAM 0(1)	PRU-ICSS1 Data 8KB RAM 1(1)	PRU-ICSS0 Data 4KB RAM 0(1)	PRU-ICSS0 Data 4KB RAM 1(1)
0x0000_2000	PRU-ICSS1 Data 8KB RAM 1(1)	PRU-ICSS1 Data 8KB RAM 0(1)	PRU-ICSS0 Data 4KB RAM 1(1)	PRU-ICSS0 Data 4KB RAM 0(1)
0x0001_0000	PRU-ICSS1 Shared Data 32KB RAM 2	PRU-ICSS1 Shared Data 32KB RAM 2	Reserved	Reserved
0x0002_0000	PRU-ICSS1 INTC	PRU-ICSS1 INTC	PRU-ICSS0 INTC	PRU-ICSS0 INTC
0x0002_2000	PRU-ICSS1 PRU0 Control	PRU-ICSS1 PRU0 Control	PRU-ICSS0 PRU0 Control	PRU-ICSS0 PRU0 Control
0x0002_2400	Reserved	Reserved	Reserved	Reserved

(1) Data RAM0 is intended to be the primary data memory for PRU0, as is Data RAM1 for PRU1. However, both PRU cores can access Data RAM0 and Data RAM1 to pass information between PRUs. Each PRU core accesses their intended Data RAM at address 0x0000_0000 and the other Data RAM at address 0x0000_2000.

Table 30-5. Local Data Memory Map (continued)

Start Address	PRU-ICSS1		PRU-ICSS0	
	PRU0	PRU1	PRU0	PRU1
0x0002_4000	PRU-ICSS1 PRU1 Control	PRU-ICSS1 PRU1 Control	PRU-ICSS0 PRU1 Control	PRU-ICSS0 PRU1 Control
0x0002_4400	Reserved	Reserved	Reserved	Reserved
0x0002_6000	PRU-ICSS1 CFG	PRU-ICSS1 CFG	PRU-ICSS0 CFG	PRU-ICSS0 CFG
0x0002_8000	PRU-ICSS1 UART 0	PRU-ICSS1 UART 0	PRU-ICSS0 UART 0	PRU-ICSS0 UART 0
0x0002_A000	Reserved	Reserved	Reserved	Reserved
0x0002_C000	Reserved	Reserved	Reserved	Reserved
0x0002_E000	PRU-ICSS1 IEP	PRU-ICSS1 IEP	PRU-ICSS0 IEP	PRU-ICSS0 IEP
0x0003_0000	PRU-ICSS1 eCAP 0	PRU-ICSS1 eCAP 0	PRU-ICSS0 eCAP 0	PRU-ICSS0 eCAP 0
0x0003_2000	PRU-ICSS1 MII_RT CFG	PRU-ICSS1 MII_RT CFG	PRU-ICSS0 MII_RT CFG	PRU-ICSS0 MII_RT CFG
0x0003_2400	PRU-ICSS1 MII MDIO	PRU-ICSS1 MII MDIO	PRU-ICSS0 MII MDIO	PRU-ICSS0 MII MDIO
0x0003_4000	Reserved	Reserved	Reserved	Reserved
0x0003_8000	Reserved	Reserved	Reserved	Reserved
0x0004_0000	Ext port to PRU-ICSS0	Ext port to PRU-ICSS0	Ext port to PRU-ICSS1	Ext port to PRU-ICSS1
0x0008_0000	System OCP_HP0	System OCP_HP1	System OCP_HP0 (2)	System OCP_HP1(2)

(2) All external accesses to or from PRU-ICSS0 are routed through PRU-ICSS1.

30.3.2 Global Memory Map

The global view of the PRU-ICSS internal memories and control ports is shown in Table 30-6. The offset addresses of each region are implemented inside the PRU-ICSS but the global device memory mapping places the PRU-ICSS slave port in the address range shown in the external Host top-level memory map.

The global memory map is with respect to the Host point of view, but it can also be accessed by the PRU-ICSS. Note that PRU0 and PRU1 can use either the local or global addresses to access their internal memories, but using the local addresses will provide access time several cycles faster than using the global addresses. This is because when accessing via the global address the access needs to be routed through the switch fabric outside PRU-ICSS and back in through the PRU-ICSS slave port.

Each of the PRUs can access the rest of the device memory (including memory mapped peripheral and configuration registers) using the global memory space addresses. See Table 30-6, *Memory Map*, for base addresses of each module in the device. Note that all global or external accesses to or from PRU-ICSS0 are routed through PRU-ICSS1.

Table 30-6. Global Memory Map

Offset Address	PRU-ICSS
PRU-ICSS1	
0x0000_0000	PRU_ICSS1 Data 8KB RAM 0
0x0000_2000	PRU_ICSS1 Data 8KB RAM 1
0x0001_0000	PRU_ICSS1 Shared Data 32KB RAM 2
0x0002_0000	PRU_ICSS1 INTC
0x0002_2000	PRU_ICSS1 PRU0 Control
0x0002_2400	PRU_ICSS1 PRU0 Debug
0x0002_4000	PRU_ICSS1 PRU1 Control
0x0002_4400	PRU_ICSS1 PRU1 Debug
0x0002_6000	PRU_ICSS1 CFG
0x0002_8000	PRU_ICSS1 UART 0
0x0002_A000	Reserved
0x0002_C000	Reserved
0x0002_E000	PRU_ICSS1 IEP

Table 30-6. Global Memory Map (continued)

Offset Address	PRU-ICSS
0x0003_0000	PRU_ICSS1 eCAP 0
0x0003_2000	Reserved
0x0003_2400	Reserved
0x0003_2000	PRU_ICSS1 MII_RT CFG
0x0003_2400	PRU_ICSS1 MII MDIO
0x0003_4000	PRU_ICSS1 PRU0 12KB IRAM
0x0003_8000	PRU_ICSS1 PRU1 12KB IRAM
PRU-ICSS0[1]	
0x0004_0000	PRU_ICSS0 Data 4KB RAM 0
0x0004_2000	PRU_ICSS0 Data 4KB RAM 1
0x0005_0000	Reserved
0x0006_0000	PRU_ICSS0 INTC
0x0006_2000	PRU_ICSS0 PRU0 Control
0x0006_2400	PRU_ICSS0 PRU0 Debug
0x0006_4000	PRU_ICSS0 PRU1 Control
0x0006_4400	PRU_ICSS0 PRU1 Debug
0x0006_6000	PRU_ICSS0 CFG
0x0006_8000	PRU_ICSS0 UART 0
0x0006_A000	Reserved
0x0006_C000	Reserved
0x0006_E000	PRU_ICSS0 IEP
0x0007_0000	PRU_ICSS0 eCAP 0
0x0007_2000	Reserved
0x0007_2400	Reserved
0x0007_2000	PRU_ICSS0 MII_RT CFG
0x0007_2400	PRU_ICSS0 MII MDIO
0x0007_4000	PRU_ICSS0 PRU0 4KB IRAM
0x0007_8000	PRU_ICSS0 PRU1 4KB IRAM

[1] The PRU-ICSS0 memory map is accessed through the PRU-ICSS1 expansion port.

30.4 Functional Description

30.4.1 PRU Cores

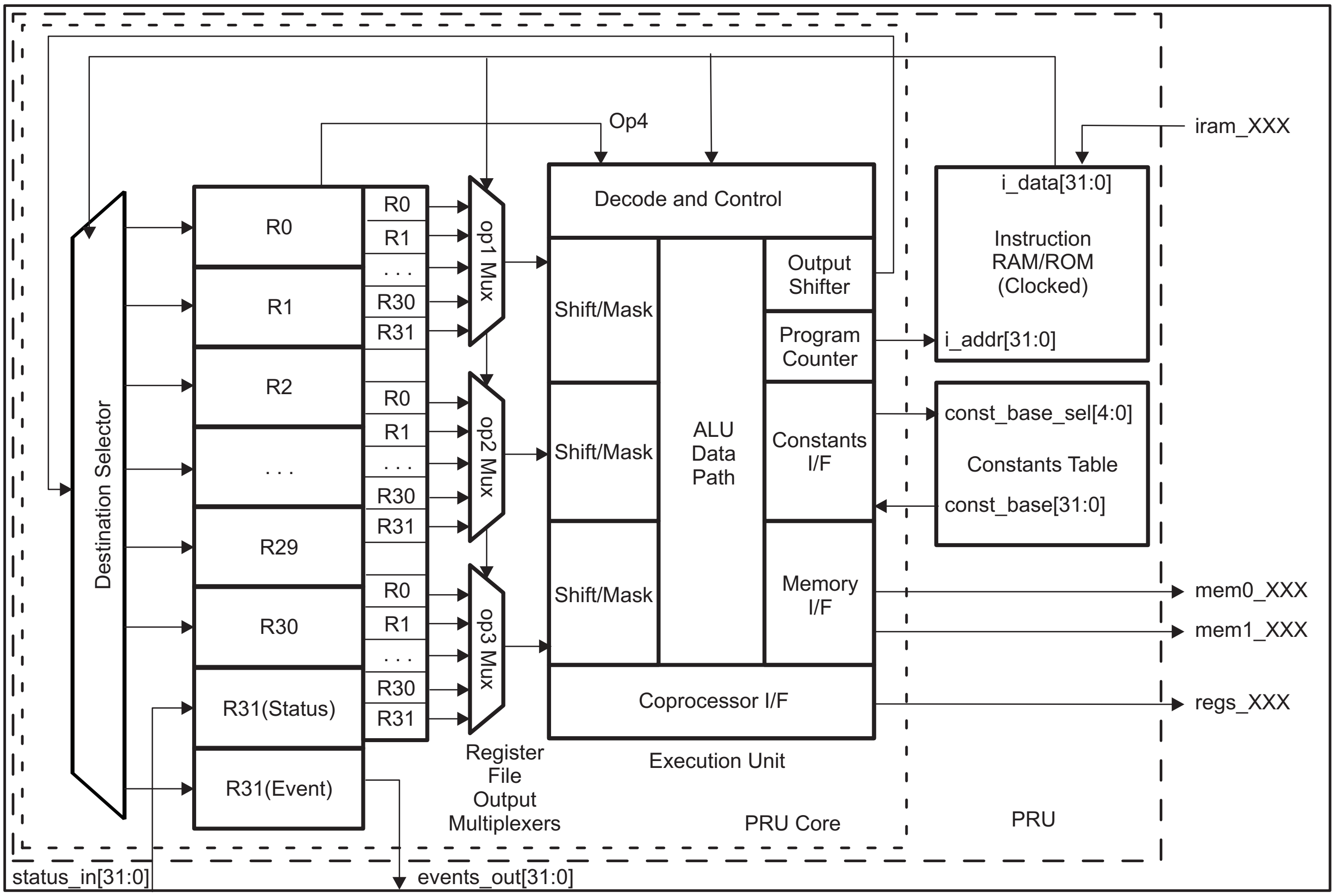
The PRU is a processor optimized for performing embedded tasks that require manipulation of packed memory mapped data structures, handling of system events that have tight real-time constraints and interfacing with systems external to the SoC. The PRU is both very small and very efficient at handling such tasks.

The major attributes of the PRU are as follows.

Attribute	Value
IO Architecture	Load / Store
Data Flow Architecture	Register to Register
Core Level Bus Architecture	
Type	4-Bus Harvard (1 Instruction, 3 Data)
Instruction I/F	32-Bit
Memory I/F 0	32-Bit
Memory I/F 1	32-Bit
Execution Model	
Issue Type	Scalar
Pipelining	None (Purposefully)
Ordering	In Order
ALU Type	Unsigned Integer
Registers	
General Purpose (GP)	30(R1 – R30)
External Status	1 (R31)
GP / Indexing	1 (R0)
Addressability in Instruction	Bit, Byte (8-bit), Halfword (16-bit), Word (32-bit), Pointer
Addressing Modes	
Load Immediate	16-bit Immediate
Load / Store – Memory	Register Base + Register Offset Register Base + 8-bit Immediate Offset Register Base with auto increment / decrement Constant Table Base + Register Offset Constant Table Base + 8-bit Immediate Offset Constant Table Base with auto increment / decrement
Data Path Width	32-Bits
Instruction Width	32-Bits
Accessibility to Internal PRU Structures	Provides 32-bit slave with three regions: • Instruction RAM • Control / Status registers • Debug access to internal registers (R0-R31) and constant table

The processor is based on a four-bus architecture which allows instructions to be fetched and executed concurrently with data transfers. In addition, an input is provided in order to allow external status information to be reflected in the internal processor status register. Figure 30-3 shows a block diagram of the processing element and the associated instruction RAM/ROM that contains the code that is to be executed.

Figure 30-3. PRU Block Diagram



30.4.1.1 Constant Table

The PRU Constants Table is a structure of hard-coded memory addresses for commonly used peripherals and memories.

The constants table exists to more efficiently load/store data to these commonly accessed addresses by:

- Reduce a PRU instruction by not needing to pre-load an address into the internal register file before loading/storing data to memory address.
- Maximizing the usage of the PRU register file for embedded processing applications by moving many of the commonly used constant or deterministically calculated base addresses from the internal register file to an external table.

Table 30-7. PRU0/1 Constant Table

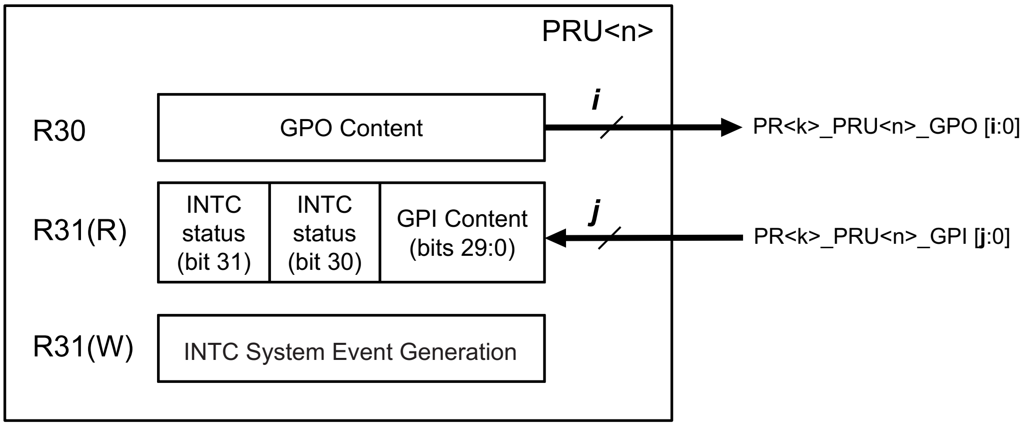
Entry No.	Region Pointed To	Value [31:0]
0	PRU-ICSS INTC (local)	0x0002_0000
1	DMTIMER2	0x4804_0000
2	I2C1	0x4802_A000
3	PRU-ICSS eCAP (local)	0x0003_0000
4	PRU-ICSS CFG (local)	0x0002_6000
5	MMCSD0	0x4806_0000
6	MCSPI0	0x4803_0000
7	PRU-ICSS UART0 (local)	0x0002_8000
8	McASP0 DMA	0x4600_0000
9	CPSW	0x4A10_0000
10	Reserved	0x4831_8000
11	UART1	0x4802_2000
12	UART2	0x4802_4000
13	Reserved	0x4831_0000
14	DCAN0	0x481C_C000
15	DCAN1	0x481D_0000
16	MCSPI1	0x481A_0000
17	I2C2	0x4819_C000
18	eHRPWM1/eCAP1/eQEP1	0x4830_0000
19	eHRPWM2/eCAP2/eQEP2	0x4830_2000
20	eHRPWM3/eCAP3/eQEP3	0x4830_4000
21	PRU-ICSS MDIO (local)	0x0003_2400
22	Mailbox0	0x480C_8000
23	Spinlock	0x480C_A000
24	PRU-ICSS PRU0/1 Data RAM (local)	0x0000_0n00, n = c24_blk_index[3:0]
25	PRU-ICSS PRU1/0 Data RAM (local)	0x0000_2n00, n = c25_blk_index[3:0]
26	PRU-ICSS IEP (local)	0x0002_En00, n = c26_blk_index[3:0]
27	PRU-ICSS MII_RT (local)	0x0003_2n00, n = c27_blk_index[3:0]
28	PRU-ICSS Shared RAM (local)	0x00nn_nn00, nnnn = c28_pointer[15:0]
29	TPCC (EDMA)	0x49nn_nn00, nnnn = c29_pointer[15:0]
30	L3 OCMC0 SRAM	0x40nn_nn00, nnnn = c30_pointer[15:0]
31	EMIF0 DDR Base	0x80nn_nn00, nnnn = c31_pointer[15:0]

NOTE: The addresses in constants entries 24–31 are partially programmable. Their programmable bit field (e.g., c24_blk_index[3:0]) is programmable through the PRU CTRL register space. As a general rule, the PRU should configure this field before using the partially programmable constant entries.

30.4.1.2 PRU Module Interface to PRU I/Os and INTC

The PRU module interface consists of the PRU internal registers 30 and 31 (R30 and R31). Figure 30-4 shows the PRU module interface and the functionality of R30 and R31. The register R31 serves as an interface with the dedicated PRU general purpose input (GPI) pins and INTC. Reading R31 returns status information from the GPI pins and INTC via the PRU Real Time Status Interface. Writing to R31 generates PRU system events via the PRU Event Interface. The register R30 serves as an interface with the dedicated PRU general purpose output (GPO) pins. Note that the R30/R31 GPO or GPI register content changes depending on the selected configuration (i.e. MII_RT, Sigma Delta, Parallel Capture, etc). See the subsequent sections for more details.

Figure 30-4. PRU Module Interface



30.4.1.2.1 Real-Time Status Interface Mapping (R31): Interrupt Events Input

The PRU Real-Time Status Interface directly feeds information into register 31 (R31) of the PRU's internal register file. The firmware on the PRU uses the status information to make decisions during execution. The status interface is comprised of signals from different modules inside of the PRU-ICSS which require some level of interaction with the PRU. More details on the Host interrupts imported into bit 30 and 31 of register R31 of both the PRUs is provided in Section 30.4.2, *Interrupt Controller*.

Table 30-8. Real-Time Status Interface Mapping (R31) Field Descriptions

Bit	Field	Value	Description
31	pru_intr_in[1]		PRU Host Interrupt 1 from local INTC
30	pru_intr_in[0]		PRU Host Interrupt 0 from local INTC
29-0	pru<n>_r31_status[29:0]		Status inputs from primary input via Enhanced GPI port

30.4.1.2.2 Event Interface Mapping (R31): PRU System Events

This PRU Event Interface directly feeds pulsed event information out of the PRU's internal ALU. These events are exported out of the PRU-ICSS and need to be connected to the system interrupt controller at the SoC level. The event interface can be used by the firmware to create interrupts from the PRU to the Host processor.

Figure 30-5. Event Interface Mapping (R31)

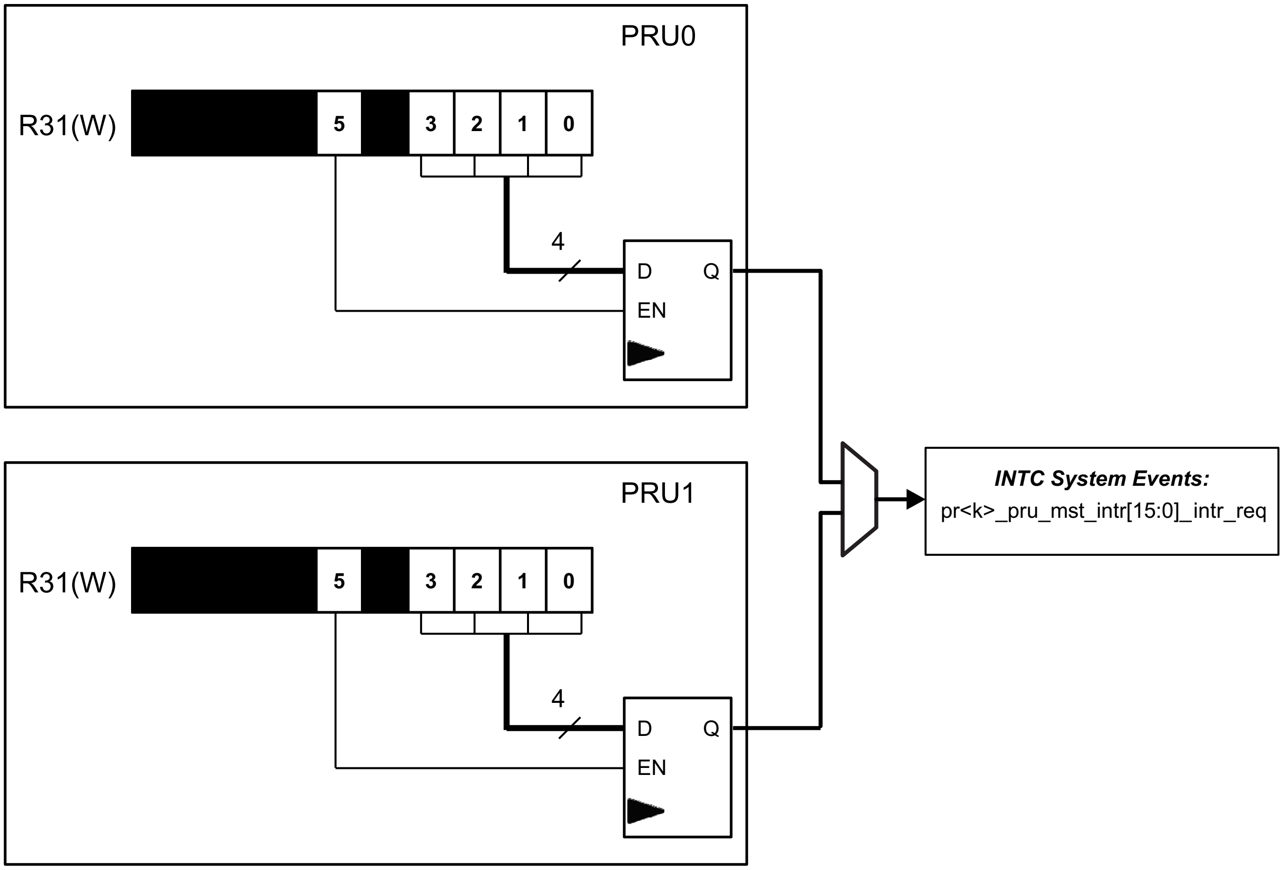


Table 30-9. Event Interface Mapping (R31) Field Descriptions

Bit	Field	Value	Description
31-6	Reserved		
5	pru<n>_r31_vec_valid		Valid strobe for vector output
4	Reserved		
3-0	pru<n>_r31_vec[3:0]		Vector output

Simultaneously writing a '1' to pru<n>_r31_vec_valid (R31 bit 5) and a channel number from 0-15 to pru<n>_r31_vec[3:0] (R31 bits 3:0) creates a pulse on the output of the corresponding pr<k>_pru_mst_intr[x]_intr_req INTC system event. For example, writing '100000' will generate a pulse on pr<k>_pru_mst_intr[0]_intr_req, writing '100001' will generate a pulse on pr<k>_pru_mst_intr[1]_intr_req, and so on to where writing '101111' will generate a pulse on pr<k>_pru_mst_intr[15]_intr_req and writing '0xxxxx' will not generate any system event pulses. The output values from both PRU cores in a subsystem are ORed together.

The output channels 0-15 are connected to the PRU-ICSS INTC system events 16-31, respectively. This allows the PRU to assert one of the system events 16-31 by writing to its own R31 register. The system event is used to either post a completion event to one of the host CPUs (ARM) or to signal the other PRU. The host to be signaled is determined by the system event to interrupt channel mapping (programmable). The 16 events are named as pr<k>_pru_mst_intr<15:0>_intr_req. For more details, see Section 30.4.2, *Interrupt Controller*.

30.4.1.2.3 General-Purpose Inputs (R31): Enhanced PRU GP Module

The PRU-ICSS implements an enhanced General-Purpose Input/Output (GPIO) module that supports the following general-purpose input modes: direct input, 16-bit parallel capture, and 28-bit serial shift in. Register R31 serves as an interface with the general-purpose inputs.

Table 30-10 describes the input modes in detail.

NOTE: Each PRU core can only be configured for one GPI mode at a time. Each mode uses the same R31 signals and internal register bits for different purposes. A summary is found in Table 30-11

NOTE: The GPCFGn register, bit PR1_PRU0_GP_MUX_SEL in the PRU-ICSS CFG register space needs to be set to 0x0 for GP mode. For a given PRU core, the following IO modes are mutually exclusive: GP mode, Sigma Delta mode, and Peripheral I/F mode.

Table 30-10. PRU R31 (GPI) Modes

Mode	Function	Configuration
Direct input	GPI[29:0] feeds directly into the PRU R31	Default mode
16-bit parallel capture	DATAIN[0:15] is captured by the posedge or negedge of CLOCKIN	• Enabled by CFG_GPCFGn register • CLOCKIN edge selected by CFG_GPCFGn register
28-bit shift in	DATAIN is sampled and shifted into a 28-bit shift register. Shift Counter (Cnt_16) feature uses ... • Shift Counter (Cnt_16) feature is mapped to pru<n>_r31_status[28]. • SB (Start Bit detection) feature is mapped to pru<n>_r31_status[29]	• Enabled by CFG_GPCFGn register • Cnt_16 is self clearing and is connected to the PRU INTC • Start Bit (SB) is cleared by CFG_GPCFGn register
MII_RT	mii_rt_r31_status [29:0] internally driven by the MII_RT module	Enabled by CFG

Table 30-11. PRU GPI Signals and Configurations(1)

Pad Names at Device Level	GPI Modes		
	Direct input	Parallel Capture	28-Bit Shift In
pr<k>_pru<n>_gpi0	GPI0	DATAIN0	DATAIN
pr<k>_pru<n>_gpi1	GPI1	DATAIN1	
pr<k>_pru<n>_gpi2	GPI2	DATAIN2	
pr<k>_pru<n>_gpi3	GPI3	DATAIN3	
pr<k>_pru<n>_gpi4	GPI4	DATAIN4	
pr<k>_pru<n>_gpi5	GPI5	DATAIN5	
pr<k>_pru<n>_gpi6	GPI6	DATAIN6	
pr<k>_pru<n>_gpi7	GPI7	DATAIN7	
pr<k>_pru<n>_gpi8	GPI8	DATAIN8	
pr<k>_pru<n>_gpi9	GPI9	DATAIN9	
pr<k>_pru<n>_gpi10	GPI10	DATAIN10	
pr<k>_pru<n>_gpi11	GPI11	DATAIN11	
pr<k>_pru<n>_gpi12	GPI12	DATAIN12	
pr<k>_pru<n>_gpi13	GPI13	DATAIN13	
pr<k>_pru<n>_gpi14	GPI14	DATAIN14	
pr<k>_pru<n>_gpi15	GPI15	DATAIN15	

---

(1) These pins also being used for Sigma Delta or Peripheral I/F mode.

Table 30-11. PRU GPI Signals and Configurations[(1)] (continued)

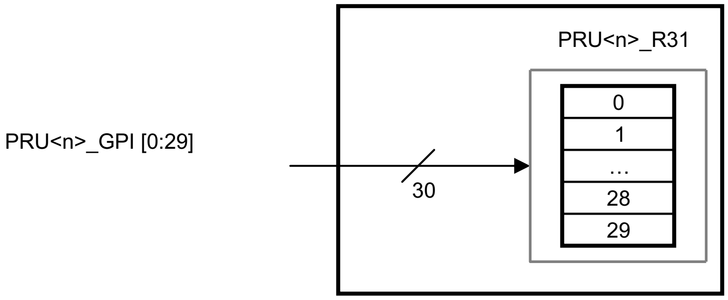
pr<k>_pru<n>_gpi16	GPI16	CLOCKIN	
pr<k>_pru<n>_gpi17	GPI17		
pr<k>_pru<n>_gpi18	GPI18		
pr<k>_pru<n>_gpi19	GPI19		
pr<k>_pru<n>_gpi20	GPI20		
pr<k>_pru<n>_gpi21	GPI21		
pr<k>_pru<n>_gpi22	GPI22		
pr<k>_pru<n>_gpi23	GPI23		
pr<k>_pru<n>_gpi24	GPI24		
pr<k>_pru<n>_gpi25	GPI25		
pr<k>_pru<n>_gpi26	GPI26		
pr<k>_pru<n>_gpi27	GPI27		
pr<k>_pru<n>_gpi28	GPI28		
pr<k>_pru<n>_gpi29	GPI29		

NOTE: Some devices may not pin out all 30 bits of R31. For which pins are available on this device, see Section 30.2.3, *PRU-ICSS Pin List*. See the device data sheet for device-specific pin mapping.

30.4.1.2.3.1 Direct Input

The pru<n>_r31_status [0:29] bits of the internal PRU register file are mapped to device-level, general purpose input pins (PRU<n>_GPI [0:29]). In GPI Direct Input mode, PRU<n>_GPI [0:29] feeds directly to pru<n>_r31_status [0:29]. There are 30 possible general purpose inputs per PRU core; however, some devices may not pin out all of these signals. See the device-specific data sheet for device-specific pin mapping.

Figure 30-6. PRU R31 (GPI) Direct Input Mode Block Diagram

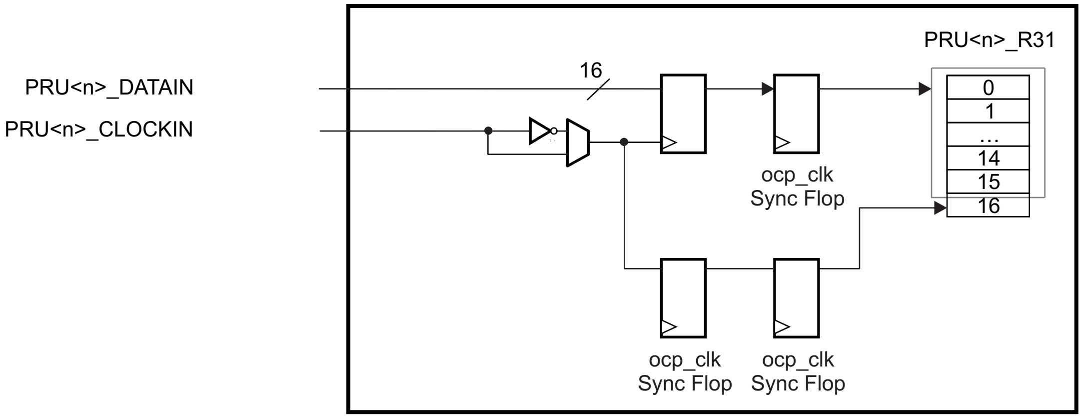


30.4.1.2.3.2 16-Bit Parallel Capture

The pru<n>_r31_status [0:15] and pru<n>_r31_status [16] bits of the internal PRU register file mapped to device-level, general purpose input pins (PRU<n>_DATAIN [0:15] and PRU<n>_CLOCKIN, respectively). PRU<n>_CLOCKIN is designated for an external strobe clock, and is used to capture PRU<n>_DATAIN [0:15].

The PRU<n>_DATAIN can be captured either by the positive or the negative edge of PRU<n>_CLOCK, programmable through the PRU-ICSS CFG register space. If the clocking is configured through the PRU-ICSS CFG register to be positive, then it will equal PRU<n>_CLOCK; however, if the clocking is configured to be negative, then it will equal PRU<n>_CLOCK inverted.

Figure 30-7. PRU R31 (GPI) 16-Bit Parallel Capture Mode Block Diagram



30.4.1.2.3.3 28-Bit Shift In

In 28-bit shift in mode, the device-level, general-purpose input pin PRU<n>_DATAIN is sampled and shifted into a 28-bit shift register on an internal clock pulse. The register fills in lsb order (from bit 0 to 27) and then overflows into a bit bucket. The 28-bit register is mapped to pru<n>_r31_status [0:27] and can be cleared in software through the PRU-ICSS CFG register space.

Note, the PRU will continually capture and shift the DATAIN input when the GPI mode has been set to 28-bit shift in.

The shift rate is controlled by the effective divisor of two cascaded dividers applied to the ocp_clk. These cascaded dividers can each be configured through the PRU-ICSS CFG register space to a value of {1, 1.5, ..., 16}. Table 30-12 shows sample, effective clock values and the divisor values that can be used to generate these clocks.

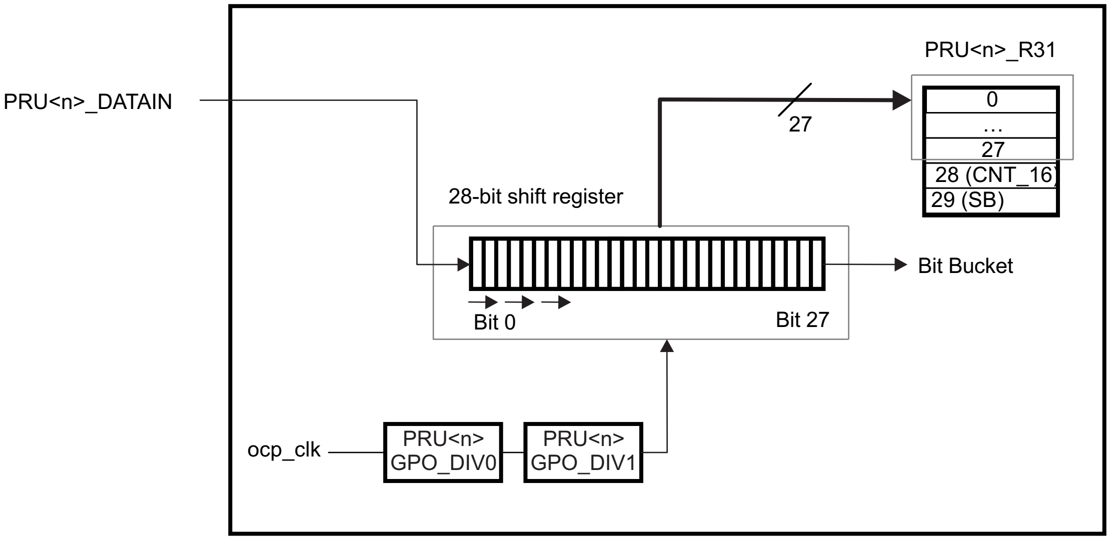
Table 30-12. Effective Clock Values

Generated clock	PRU<n>_GPI_DIV0	PRU<n>_GPI_DIV1
8-MHz	12.5 (0x17)	2 (0x02)
10-MHz	10 (0x12)	2 (0x02)
16-MHz	16 (0x1e)	1 (0x00)
20-MHz	10 (0x12)	1 (0x00)

The 28-bit shift in mode also supports the following features:

- SB (Start Bit detection) is mapped to pru<n>_r31_status[29] and is set when the first 1 is captured on PRU<n>_DATAIN. The SB flag in pru<n>_r31_status[29] is cleared in software through the CFG_GPCFGn registor of the PRU-ICSS CFG register space.
- CNT_16 (Shift Counter) is mapped to pru<n>_r31_status[28] and is set on every 16 shift clock samples after the Start Bit has been received.CNT_16 is self clearing and is connected to the PRU-ICSS INTC. See Section 30.4.2,*Interrupt Controller (INTC)*, for more details.

Figure 30-8. PRU R31 (GPI) 28-Bit Shift In Mode



30.4.1.2.4 General-Purpose Outputs (R30): Enhanced PRU GP Module

The PRU-ICSS implements an enhanced General Purpose Input Output (GPIO) module that supports two general-purpose output modes: direct output and shift out.

Table 30-13 describes these modes in detail.

NOTE: Note, each PRU core can only be configured for one GPO mode at a time. Each mode uses the same R30 signals and internal register bits for different purposes. A summary is found in Table 30-13.

NOTE: The GPCFGn register, bit PR1_PRU0_GP_MUX_SEL in the PRU-ICSS CFG register space needs to be set to 0x0 for GP mode. For a given PRU core, the following IO modes are mutually exclusive: GP mode, Sigma Delta mode, and Peripheral I/F mode.

Table 30-13. PRU R30 (GPO) Output Mode

Mode	Function	Configuration
Direct output	pru<n>_r30[0:31] feeds directly to GPO[0:31]	Default mode
Shift Out	• pru<n>_r30[0] is shifted out on DATAOUT on every rising edge of pru<n>_r30[1] (CLOCKOUT). • LOAD_GPO_SH0 (Load Shadow Register 0) is mapped to pru<n>_r30[29]. • LOAD_GPO_SH1 (Load Shadow Register 1) is mapped to pru<n>_r30[30]. • ENABLE_SHIFT is mapped to pru<n>_r30[31].	Enabled by CFG_GPCFGn register

---

NOTE: Some devices may not pin out all 32 bits of R30. For which pins are available on this device, see Section 30.2.3, *PRU-ICSS Pin List*. See the device data sheet for device-specific pin mapping.

Note: R30 may not be initialized after reset. To avoid outside effect, R30 should be initialized before PINMUX configuration.

Table 30-14. GPO Mode Descriptions(1)(2)

Pad names at device level	GPO Modes	
	Direct output	Shift out
pr<k>_pru<n>_gpo0	GPO0	DATAOUT
pr<k>_pru<n>_gpo1	GPO1	CLOCKOUT
pr<k>_pru<n>_gpo2	GPO2	
pr<k>_pru<n>_gpo3	GPO3	
pr<k>_pru<n>_gpo4	GPO4	
pr<k>_pru<n>_gpo5	GPO5	
pr<k>_pru<n>_gpo6	GPO6	
pr<k>_pru<n>_gpo7	GPO7	
pr<k>_pru<n>_gpo8	GPO8	
pr<k>_pru<n>_gpo9	GPO9	
pr<k>_pru<n>_gpo10	GPO10	
pr<k>_pru<n>_gpo11	GPO11	
pr<k>_pru<n>_gpo12	GPO12	
pr<k>_pru<n>_gpo13	GPO13	
pr<k>_pru<n>_gpo14	GPO14	
pr<k>_pru<n>_gpo15	GPO15	
pr<k>_pru<n>_gpo16	GPO16	
pr<k>_pru<n>_gpo17	GPO17	
pr<k>_pru<n>_gpo18	GPO18	
pr<k>_pru<n>_gpo19	GPO19	
pr<k>_pru<n>_gpo20	GPO20	
pr<k>_pru<n>_gpo21	GPO21	
pr<k>_pru<n>_gpo22	GPO22	
pr<k>_pru<n>_gpo23	GPO23	
pr<k>_pru<n>_gpo24	GPO24	
pr<k>_pru<n>_gpo25	GPO25	
pr<k>_pru<n>_gpo26	GPO26	
pr<k>_pru<n>_gpo27	GPO27	
pr<k>_pru<n>_gpo28	GPO28	
pr<k>_pru<n>_gpo29	GPO29	
pr<k>_pru<n>_gpo30	GPO30	
pr<k>_pru<n>_gpo31	GPO31	

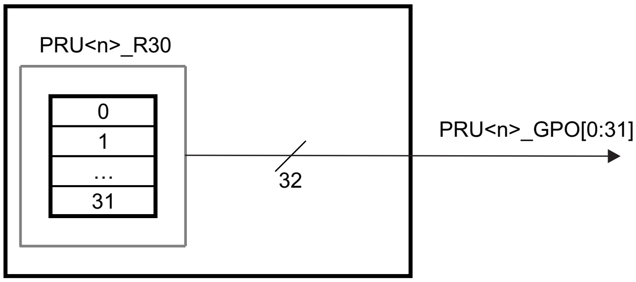
(1) These pins are also used for Sigma Delta or Peripheral I/F mode.

(2) Some devices may not pin out all 30 bits of R31. For which pins are available on this device, see Section 30.2.3, PRU-ICSS Pin List. See the device data sheet for device-specific pin mapping.

30.4.1.2.4.1 Direct Output

The pru<n>_r30 [0:31] bits of the internal PRU register files are mapped to device-level, general-purpose output pins (PRU<n>_GPO[0:31]). In GPO Direct Output mode, pru<n>_r30[0:31] feed directly to PRU<n>_GPO[0:31]. There are 32 possible general-purpose outputs for each PRU core; however, some devices may not pin out all of these signals. See the device-specific data sheet for device-specific pin mapping.

Figure 30-9. PRU R30 (GPO) Direct Output Mode Block Diagram



NOTE: R30 is not initialized after reset. To avoid unintended output signals, R30 should be initialized before pinmux configuration of PRU signals.

30.4.1.2.4.2 Shift Out

In shift out mode, data is shifted out of pru<n>_r30[0] (on PRU<n>_DATAOUT) on every rising edge of pru<n>_r30[1] (PRU<n>_CLOCKOUT). The shift rate is controlled by the effective divisor of two cascaded dividers applied to the ocp_clk. These cascaded dividers can each be configured through the PRU-ICSS CFG register space to a value of {1, 1.5, …, 16}. Table 30-15 shows sample effective clock values and the divisor values that can be used to generate these clocks. Note that the shift out clock is a free-running clock that is always running internally. This clock will be output to PRU<n>_CLOCKOUT (or pr <k>_pru<n>_pru_r30_1) when the PRU GPO mode is set to shift out mode.

Table 30-15. Effective Clock Values

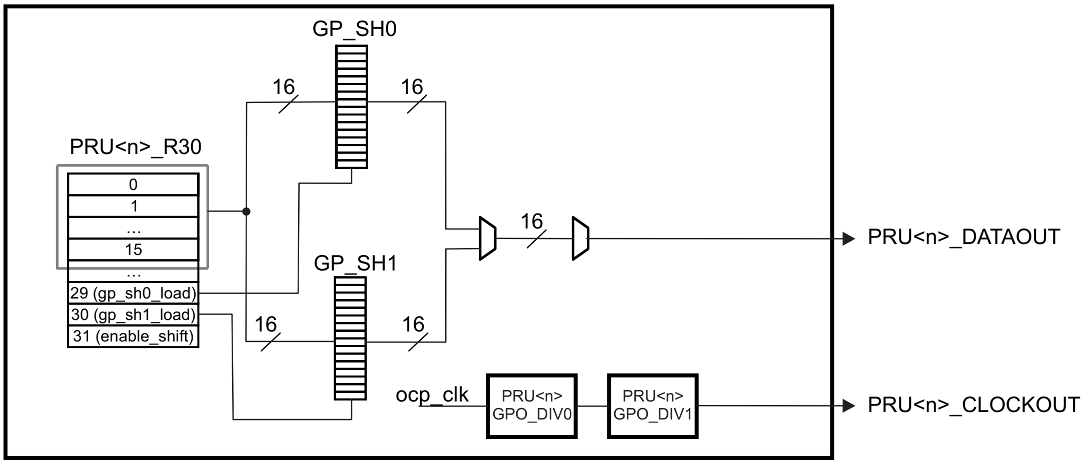
Generated Clock	PRU<n>_GPO_DIV0	PRU<n>_GPO_DIV1
8 MHz	12.5 (0x17)	2 (0x02)
10 MHz	10 (0x12)	2 (0x02)
16 MHz	16 (0x1e)	1 (0x00)
20 MHz	10 (0x12)	1 (0x00)

Shift out mode uses two 16-bit shadow registers (gpo_sh0 and gpo_sh1) to support ping-pong buffers. Each shadow register has independent load controls programmable through pru<n>_r30[29:30] (PRU<n>_LOAD_GPO_SH [0:1]). While PRU<n>_LOAD_GPO_SH [0/1] is set, the contents of pru<n>_r30[0:15] are loaded into gpo_sh0/1.

NOTE: If any device-level pins mapped to pru<n>_r30[2:15] are configured for the pru<n>_r30 [2:15] pinmux mode, then these pins will reflect the shadow register value written to pru<n>_r30. Any pin configured for a different pinmux setting will not reflect the shadow register value written to pru<n>_r30.

The data shift will start from the lsb of gpo_sh0 when pru<n>_r30[31] (PRU<n>_ENABLE_SHIFT) is set. Note that if no new data is loaded into gpo_shn<n> after shift operation, the shift operation will continue looping and shifting out the pre-loaded data. When PRU<n>_ENABLE_SHIFT is cleared, the shift operation will finish shifting out the current shadow register, stop, and then reset.

Figure 30-10. PRU R30 (GPO) Shift Out Mode Block Diagram



Follow these steps to use the GPO shift out mode:

Step One: Initialization

1. Load 16-bits of data into gpo_sh0:
   - i. Set R30[29] = 1 (PRU<n>_LOAD_GPO_SH0)
   - ii. Load data in R30[15:0]
   - iii. Clear R30[29] to turn off load controller
2. Load 16-bits of data into gpo_sh1:
   - i. Set R30[30] = 1 (PRU<n>_LOAD_GPO_SH1)
   - ii. Load data in R30[15:0]
   - iii. Clear R30[30] to turn off load controller
3. Start shift operation:
   - i. Set R30[31] = 1 (PRU<n>_ENABLE_SHIFT)

Step 2: Shift Loop:

1. Monitor when a shadow register has finished shifting out data and can be loaded with new data:
   - i. Poll PRU<n>_GPI_SH_SEL bit of the GPCFG<n> register
   - ii. Load new 16-bits of data into gpo_sh0 if PRU<n>_GPI_SH_SEL = 1
   - iii. Load new 16-bits of data into gpo_sh1 if PRU<n>_GPI_SH_SEL = 0
2. If more data to be shifted out, loop to Shift Loop
3. if there's not more data to be shifted out, exit loop

Step 3: Exit:

1. End shift operation:
   - i. Clear R30[31] to turn off shift operation

NOTE: Until the shift operation is disabled, the shift loop will continue looping and shifting out the pre-loaded data if no new data has been loaded into gpo_sh<n>.

30.4.1.2.5 Sigma Delta (SD) Decimation Filtering H/W

Sigma-delta Sinc filtering is achieved by the combination of PRU hardware and firmware. PRU hardware provides hardware integrators that do the accumulation part of Sinc filtering, while the differentiation part is done in firmware.

The integrator serves to count the number of 1’s per clock event. Each channel has three cascaded counters, which are the accumulators for the Sinc3 filter. Each counter is 24 bits, giving a maximum count of 16,777,215. Each channel has a free running rollover clock counter. This sample counter updates the count value on the effective clock event for that channel. Each channel also contains a programmable counter compare block, and the compare register has a size of 8 bits. However, the minimum value is 4 and maximum value is 202 due to the 24-bit accumulator. Once sample counter compare value is reached, the shadow register copy is updated and the shadow register copy flag is set.

Features of the integrators in PRUs SD Demodulator:

- Up to nine channels with concurrent counting
- Flexible clock source configuration for each channel; option of independent clock source for each channel or one clock source for three channels
- Programmable, 8-bit sample counter compare register; used to set the OSR of Sinc filter
- Three 24-bit cascaded counters per channel for accumulation, only Sinc3 and Sinc2 modes supported
- Common channel enable (all channels are active or none are active)

30.4.1.2.5.1 Block Diagram and Signals

The Sigma Delta’s I/Os are multiplexed with the PRU GPI/GPO signals, as shown in Table 30-17. Note the PR<k>_PRU<n>_GP_MUX_SEL bitfield in the GPCFG<n> register must be set to 0x3 for configure the GPI/GPO signals for SD mode.

Table 30-16. PRU GPI Signals and Configurations for Sigma Delta

Pad Names at Device Level(1)(2)	Sigma Delta (SD) Mode (GPCFGn[PR1_PRUn_GP_MUX_SEL] = 0x3)
pr<k>_pru<n>_gpi0	SD0_CLK
pr<k>_pru<n>_gpi1	SD0_D
pr<k>_pru<n>_gpi2	SD1_CLK
pr<k>_pru<n>_gpi3	SD1_D
pr<k>_pru<n>_gpi4	SD2_CLK
pr<k>_pru<n>_gpi5	SD2_D
pr<k>_pru<n>_gpi6	SD3_CLK
pr<k>_pru<n>_gpi7	SD3_D
pr<k>_pru<n>_gpi8	SD4_CLK
pr<k>_pru<n>_gpi9	SD4_D
pr<k>_pru<n>_gpi10	SD5_CLK
pr<k>_pru<n>_gpi11	SD5_D
pr<k>_pru<n>_gpi12	SD6_CLK
pr<k>_pru<n>_gpi13	SD6_D
pr<k>_pru<n>_gpi14	SD7_CLK
pr<k>_pru<n>_gpi15	SD7_D
pr<k>_pru<n>_gpi16	SD8_CLK
pr<k>_pru<n>_gpi17	SD8_D
pr<k>_pru<n>_gpi18	
pr<k>_pru<n>_gpi19	
pr<k>_pru<n>_gpi20	
pr<k>_pru<n>_gpi21	
pr<k>_pru<n>_gpi22	
pr<k>_pru<n>_gpi23	
pr<k>_pru<n>_gpi24	

(1) Note these signals are shared with the GP and Peripheral I/Fs. To configure for Sigma Delta, GPCFGn[PR1_PRU0_GP_MUX_SEL] needs to be set to 0x3 for SD mode.

(2) NOTE: Some devices may not pin out all 32 bits of R30. For which pins are available on this device, see Section 30.2.3, PRU-ICSS Pin List. See the device data sheet for device-specific pin mapping.

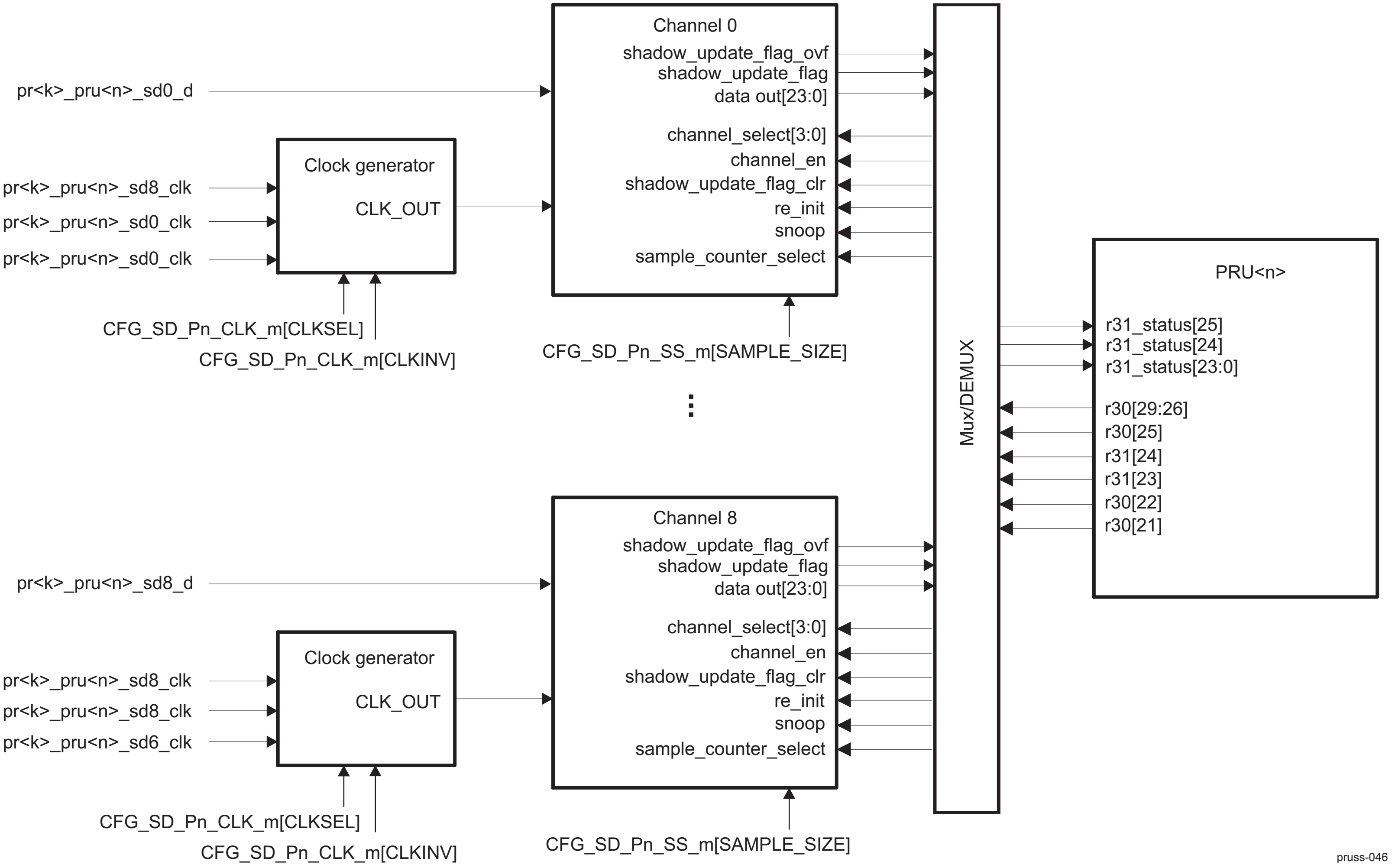
Table 30-16. PRU GPI Signals and Configurations for Sigma Delta (continued)

Pad Names at Device Level[(1)][(2)]	Sigma Delta (SD) Mode (GPCFGn[PR1_PRUn_GP_MUX_SEL] = 0x3)
pr<k>_pru<n>_gpi25	
pr<k>_pru<n>_gpi26	
pr<k>_pru<n>_gpi27	
pr<k>_pru<n>_gpi28	
pr<k>_pru<n>_gpi29	

The pr<k>_pru0_gpo1 signal (muxed with SD0_D) can be used as SD_CLKOUT when PRU-ICSS generates clock. This is a trade-off as PRU application will lose one SD channel. SD_CLKOUT needs to go through a clock generator chip if driving multiple sigma delta modulators and also be looped back into PRU-ICSS as SD_CLKIN, typically pru_gpi16.

Note to output the SD clock on pr<k>_pru0_gpo1, this device requires that the PRU core be configured for SD and shift out mode (CFG_GPCFG<n> [PR1_PRU<n>_GP_MUX_SEL] = 0x3 and CFG_GPCFG<n> [PRU<n>_GPO_MODE] = 0x1). Be sure to configure the shift out mode's clock divisors before enabling shift out mode (CFG_GPCFG<n> [PRU<n>_GPO_MODE] = 0x1). Additionally, the PRU-ICSS0 PRU0 SD clock is routed to both pr0_pru0_gpo1 and pr0_pru1_gpo1. Figure 30-11 shows a block diagram of the Sigma Delta implementation. Full description of the PRU R30 and R31 registers are shown in Table 30-18 and Table 30-19.

Figure 30-11. Sigma Delta Block Diagram



Note each channel can independently be configured to use one of three external clock sources. Table 30-17 shows the clock source options, selectable through CFG_SD_P<n>_CLK_<m> [CLKSEL].

Table 30-17. External Clock Sources

CLKSEL value	Clock Source
0	pru<n>_sd8_clk
1	pru<n>_sd<m>_clk
2	pru<n>_sd0_clk for sd0, sd1, and sd2; pru<n>_sd3_clk for sd3, sd4, and sd5; pru<n>_sd6_clk for sd6, sd7, and sd8

30.4.1.2.5.2 PRU R30 / R31 Interface

The PRU uses the R30 and R31 registers to interface with the Sigma Delta interface. Table 30-17 shows the R31 and R30 interface for the Sigma Delta mode. Note that only the parameters and data for one channel can be viewed at a time. The channel to be viewed is determined by the r30[29-26] (channel_select).

Table 30-18. PRU R31: SD Output Interface
Delta Sigma PRU registers: R31Sigma Delta PRU Registers: R31

Bits	Field Name	Description
29-26	Reserved	
25	shadow_update_flag_ovf / shadow_update_flag_ovf_clr	Shadow update flag overflow, set when over sample count equals over sample size and shadow_update_flag is still set. Set this bit to clear the flag.
24	shadow_update_flag / shadow_update_flag_clr	Shadow update flag overflow, set when over sample count equals over sample size and shadow_update_flag is still set. Set this bit to clear the flag.
23	re_init/data_out[23]	re_init (write): Set to reset all counters, flags, and shadow copy. Updates over_sample_size based on the current PRU_ICSS_CFG_SD_P<n>_SS<i> register on the selected channel. data_out[23](read): most-significant bit of sample data
22-0	data_out[22:0]	Selected sample data excluding most-significant bit

Table 30-19. PRU R30: SD Input Interface
Delta Sigma PRU registers: R30Sigma Delta PRU Registers: R30

Bits	Field Name	Description
31-30	Reserved	
29-26	channel_select[3:0]	Select Channel. Note PRU firmware should add at least 1 PRU cycle delay after selecting a channel. 0x0 = Channel 0 ….. 0x8 = Channel 8 0x9-0xF = Unused
25	channel_en	Global Channel enable (effects all 9 channels). 0x0 = All channels disabled. Counters/flags are cleared. 0x1 = All channels enabled.
24-23	Reserved	
22	snoop	Enable snoop (i.e. fetch data) on the selected channel. 0x0 = acc2/acc3 shadow copy 0x1 = current acc2/acc3
21	sample_counter_select	Read sample counter. 0x0 = Not selected 0x1 = Sample count selected

Table 30-19. PRU R30: SD Input Interface Delta Sigma PRU registers: R30Sigma Delta PRU Registers: R30 (continued)

Bits	Field Name	Description
20-0	Reserved	

The PRU_ICSS CFG register space has additional MMRs for controlling the SD demodulator module:

- CFG_SD_P<n>_CLKSEL_<m> [acc2_sel] - Selects accumulator 2 as source
- CFG_SD_P<n>_CLKSEL_<m> [clk_inv] - Inverts clock
- CFG_SD_P<n>_CLKSEL_<m> [clk_sel] - Selects clock source
- CFG_SD_P<n>_SS_<m> [sample_size] - Selects number of samples to read before giving output

30.4.1.2.5.3 Sigma Delta Description

Figure 30-12 shows a block diagram of the Sigma Delta hardware integrators and integration with the PRU R30 / R31 interface for a single channel.

Figure 30-12. Sigma Delta Hardware Integrators Block Diagram (snoop = 0)

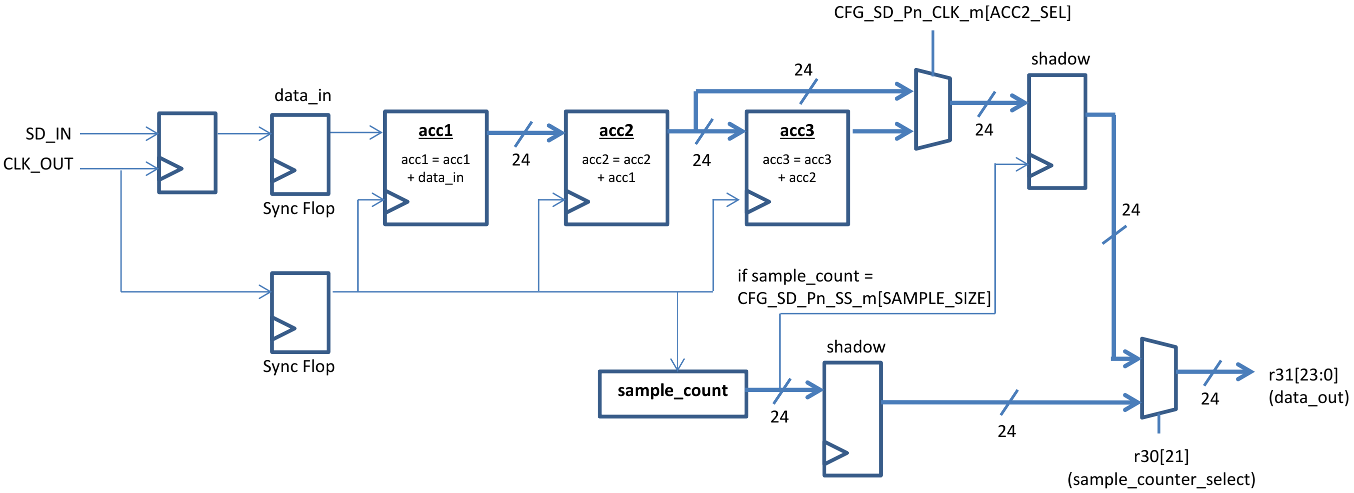
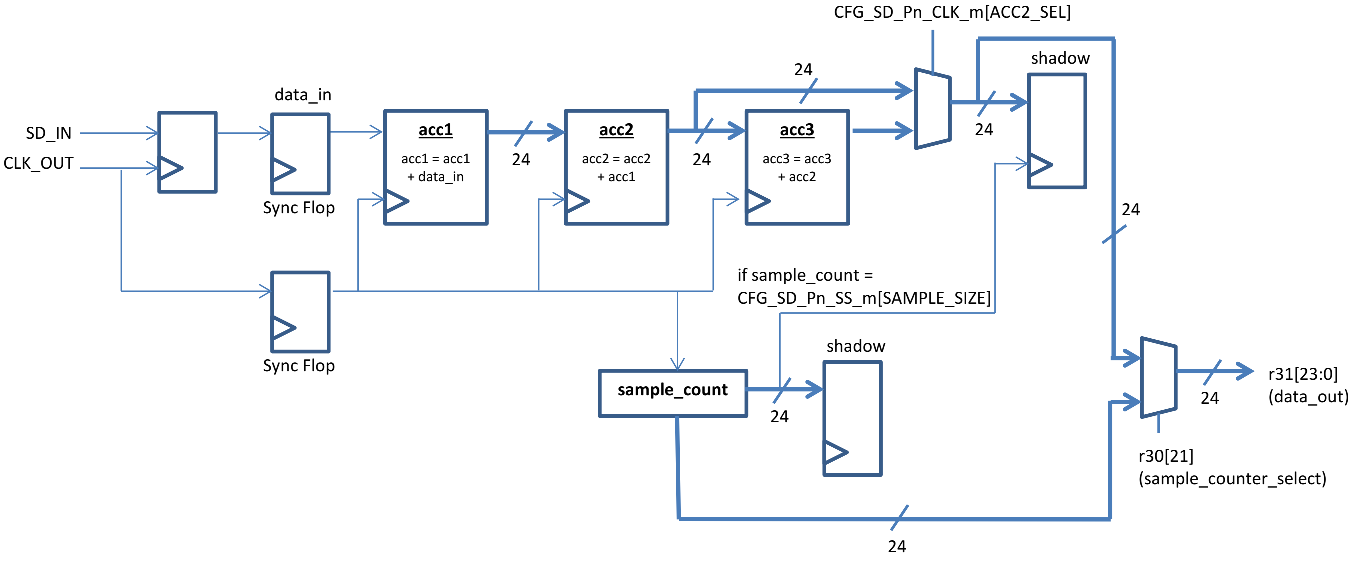


Figure 30-13. Sigma Delta Hardware Integrators Block Diagram (snoop = 1)



The three accumulators (acc1-acc3) for each channel are simple 24 bit adders. The input for acc1 is 1-bit, while the inputs for acc2 and acc3 are 24-bits. On each positive edge of the CLK_OUT, all three 24-bit counters (acc1-acc3) and the sample counter for each channel will get updated as follows:

```
acc1 = acc1 + data_in
acc2 = acc2 + acc1
acc3 = acc3 + acc2
sample_count = sample_count + 1
```

Each accumulator will rollover at 0xFF_FFFF. For example if acc2 = 0x10 and acc3 = 0xFF_FFFF, then acc3 will update to 0x00_0000F on the next clock event. Sample_count will rollover when it equals the defined sample size (CFG_SD_P<n>_SS_<m> [sample_size]).

Note that while the channels are not enabled , no operations are performed and all flags and counters are cleared. If a new sample size is to be loaded, the PRU firmware should assert re_init (r31[23]), and all stored count values are cleared to 0.

The Sigma Delta interface has two status flags:

- Shadow update flag (r31[24])
- Shadow update flag overflow (r31[25])

When sample_count equals the defined sample size (CFG_SD_P<n>_SS_<m> [sample_size]), then the acc2/acc3 shadow register copy will be updated, the shadow_update_flag (r31[24]) will be set, and sample_count will rollover to 0. The PRU firmware can clear this flag by writing ‘1’ to shadow_update_flag_clr (r31[24]). If sample_count equals the defined sample size and the shadow_update_flag is still set, then shadow_update_flag_ovf (r31[25]) will be set. Similarly, the PRU firmware can clear this flag by writing ‘1’ to shadow_update_flag_ovf_clr (r31[25]). Note that the clear operation for both flags has a higher priority than the set event.

The PRU firmware can monitor the acc2/acc3 and sample_count values through data_out[23:0] (r31[23:0]). Table 30-20 shows the configuration options for data_out[23:0].

Table 30-20. Data_out[23:0] Configuration Options

snoop (r30[22])	sample_counter_select (r30[21])	data_out (r31[23:0])
0	0	Reads acc2/acc3 shadow register copy
1	0	Reads acc2/acc3 directly
0	1	Reads sample_count shadow register copy
1	1	Reads sample_count directly

30.4.1.2.5.4 Basic Programming Example

The following programming example assumes that the PRU is configured for Sigma Delta Mode (CFG_GPCFG<n> [PR1_PRU<n>_GP_MUX_SEL] = 3).

1. Configure clock sources, accumulator source, and sample size:
    a. CFG_SD_P<n>_CLKSEL_<m> [clk_sel] for clock source
    b. CFG_SD_P<n>_CLKSEL_<m> [clk_inv] for clock polarity
    c. CFG_SD_P<n>_CLKSEL_<m> [acc2_sel] for accumulator source
    d. CFG_SD_P<n>_SS_<m> [sample_size] for sample size
2. Reinitialize all channels whose sample size was configured
    a. Select channel by writing to channel_select (r30[29-26])
    b. Delay at least 1 PRU cycle before executing re_int in step 2c.
    c. Reinitialize selected channel by writing to re_init (r31[23])
    d. Repeat steps 2a & 2b for all configured channels
3. Enable all channels by writing ‘1’ to channel_en (r30[25])
4. Select channel by writing to channel_select (r30[29-26])
    a. Poll shadow_update_flag (r31[24]) to detect when acc2/acc3 shadow register copy data is ready to

be ready

   b. Delay at least 1 PRU cycle before polling shadow_update_flag in Step 4c.
   c. Read data_out[23:0] (r31[23:0])
   d. Clear shadow_update_flag by writing ‘1’ to r31[24]

5. Repeat step 4 for new channel

30.4.1.2.6 Peripheral Interface

The Peripheral Interface supports functionality for operations utilizing protocols like EnDat 2.2 and BiSS.

This module supports the following features:

- 3 channels with baud range from 100 kHz to 16 MHz
- uart_clk (default) or ocp_clk master clock is an input to independent div16fr clock dividers to produce a 1X clock (ENDAT<m>_CLK) and oversampling clock
- Half-duplex (TX and RX are not supported concurrently)
- TX FIFO size of 32 bits
- RX FIFO size of 32 bits
- Configurable shift size/oversampling on RX
- Optional RX frame size auto shut off
- Programmable HW delay 1 (wire delay, controlling when the clock signal is first driven low) and delay 2 (tst delay, controlling when the clock signal is first driven high) on TX operation
- Optional programmable TX termination
- Individual TX channel start trigger (tx_channel_go) or simultaneous TX start trigger for all channels (tx_global_go)
- Flexible HW assisted clock output generation to allow free running, stop high and stop low (after last RX data), or stop high (after last TX data) operation with optional software clock override feature
- Optional SW direct snoop of data input

30.4.1.2.6.1 Block Diagram and Signal Configuration

The Peripheral Interface’s I/Os are multiplexed with the PRU GPI/GPO signals, as shown in Table 30-21. The PR<k>_PRI<n>_GP_MUX_SEL bitfield in the GPCFG<n> register must be set to 0x1 for configure the GPI/GPO signals for Peripheral I/F mode.

Table 30-21. PRU GPI/GPO Signals and Configurations for Peripheral I/F(1)

Pad Names at Device Level(2)(3)	Peripheral I/F Mode (GPCFGn[PR1_PRUn_GP_MUX_SEL] = 0x1)
pr<k>_pru<n>_gpi0	
pr<k>_pru<n>_gpi1	
pr<k>_pru<n>_gpi2	
pr<k>_pru<n>_gpi3	
pr<k>_pru<n>_gpi4	
pr<k>_pru<n>_gpi5	
pr<k>_pru<n>_gpi6	
pr<k>_pru<n>_gpi7	
pr<k>_pru<n>_gpi8	
pr<k>_pru<n>_gpi9	ENDAT0_IN
pr<k>_pru<n>_gpi10	ENDAT1_IN

(1) Usage of the Peripheral Interface signals are not restricted to only ENDAT interfaces.
(2) Note these signals are shared with the GP and Sigma Delta modes. To configure for Periph I/F, GPCFGn[PR1_PRU0_GP_MUX_SEL] needs to be set to 0x1 for Periph I/F mode.
(3) Some devices may not pin out all 29 bits of R31 and all 32 bits of R30. For which pins are available on this device, see Section 30.2.3, PRU-ICSS Pin List. See the device data sheet for device-specific pin mapping.

Table 30-21. PRU GPI/GPO Signals and Configurations for Peripheral I/F[(1)] (continued)

Pad Names at Device Level[(2)][(3)]	Peripheral I/F Mode (GPCFGn[PR1_PRUn_GP_MUX_SEL] = 0x1)
pr<k>_pru<n>_gpi11	ENDAT2_IN
pr<k>_pru<n>_gpi12	
pr<k>_pru<n>_gpi13	
pr<k>_pru<n>_gpi14	
pr<k>_pru<n>_gpi15	
pr<k>_pru<n>_gpi16	
pr<k>_pru<n>_gpi17	
pr<k>_pru<n>_gpi18	
pr<k>_pru<n>_gpi19	
pr<k>_pru<n>_gpi20	
pr<k>_pru<n>_gpi21	
pr<k>_pru<n>_gpi22	
pr<k>_pru<n>_gpi23	
pr<k>_pru<n>_gpi24	
pr<k>_pru<n>_gpi25	
pr<k>_pru<n>_gpi26	
pr<k>_pru<n>_gpi27	
pr<k>_pru<n>_gpi28	
pr<k>_pru<n>_gpi29	
pr<k>_pru<n>_gpo0	ENDAT0_CLK
pr<k>_pru<n>_gpo1	ENDAT0_OUT
pr<k>_pru<n>_gpo2	ENDAT0_OUT_EN
pr<k>_pru<n>_gpo3	ENDAT1_CLK
pr<k>_pru<n>_gpo4	ENDAT1_OUT
pr<k>_pru<n>_gpo5	ENDAT1_OUT_EN
pr<k>_pru<n>_gpo6	ENDAT2_CLK
pr<k>_pru<n>_gpo7	ENDAT2_OUT
pr<k>_pru<n>_gpo8	ENDAT2_OUT_EN
pr<k>_pru<n>_gpo9	
pr<k>_pru<n>_gpo10	
pr<k>_pru<n>_gpo11	
pr<k>_pru<n>_gpo12	
pr<k>_pru<n>_gpo13	
pr<k>_pru<n>_gpo14	
pr<k>_pru<n>_gpo15	
pr<k>_pru<n>_gpo16	
pr<k>_pru<n>_gpo17	
pr<k>_pru<n>_gpo18	
pr<k>_pru<n>_gpo19	
pr<k>_pru<n>_gpo20	
pr<k>_pru<n>_gpo21	
pr<k>_pru<n>_gpo22	
pr<k>_pru<n>_gpo23	
pr<k>_pru<n>_gpo24	
pr<k>_pru<n>_gpo25	
pr<k>_pru<n>_gpo26	
pr<k>_pru<n>_gpo27	

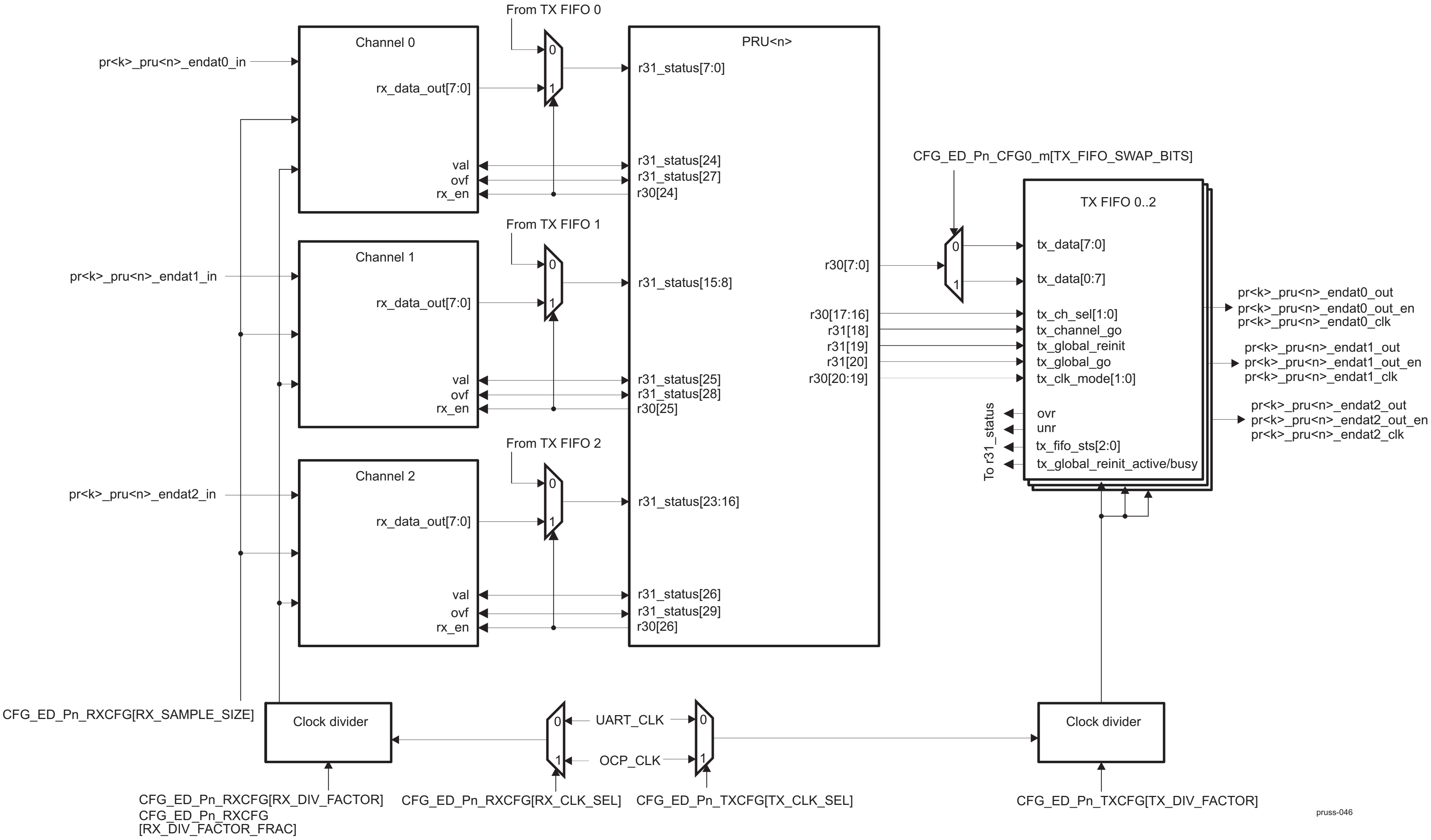
Table 30-21. PRU GPI/GPO Signals and Configurations for Peripheral I/F[(1)] (continued)

Pad Names at Device Level[(2)(3)]	Peripheral I/F Mode (GPCFGn[PR1_PRUn_GP_MUX_SEL] = 0x1)
pr<k>_pru<n>_gpo28	
pr<k>_pru<n>_gpo29	
pr<k>_pru<n>_gpo30	
pr<k>_pru<n>_gpo31	

A block diagram for the Peripheral I/F is included in Figure 30-14. As shown, each channel is composed of four I/Os:

- ENDAT<m>_IN - RX input data
- ENDAT<m>_CLK - Clock (CLK_OUT) generated by the 1x (or TX) clock. The default value is 1.
- ENDAT<m>_OUT - TX output data. The default value is 0.
- ENDAT<m>_OUT_EN - TX enable output (1 = TX mode, 0 = RX mode). The default value is 0. Note this signal is auto controlled by hardware.

Figure 30-14. Peripheral I/F Block Diagram



30.4.1.2.6.2 PRU R30 and R31 Interface

The PRU uses the R30 and R31 registers to interface with the Peripheral I/F. Table 30-22 shows the R31 and R30 interface for the Peripheral I/F RX mode, and Table 30-23 shows the comparable interface for the TX mode.

Table 30-22. Peripheral I/F: RX

Register	Bits	Field Name	Description
R31	31-30	Reserved	PRU Host Interrupts 1/0 from local INTC
	29	ovf2	Overflow Flag for Channel 2. Write 1 to clear.
	28	ovf1	Overflow Flag for Channel 1. Write 1 to clear.
	27	ovf0	Overflow Flag for Channel 0. Write 1 to clear.
	26	val2	Valid Flag for Channel 2. Write 1 to clear.
	25	val1	Valid Flag for Channel 1. Write 1 to clear.
	24	val0	Valid Flag for Channel 0. Write 1 to clear.
	23-16	rx_data_out2	Oversampled Data Output for Channel 2. Note these bits are shared with the TX Interface. When TX_FIFO has stopped transmission, RX data will be selected.
	15-8	rx_data_out1	Oversampled Data Output for Channel 1. Note these bits are shared with the TX Interface. When TX_FIFO has stopped transmission, RX data will be selected.
	7-0	rx_data_out0	Oversampled Data Output for Channel 0. Note these bits are shared with the TX Interface. When TX_FIFO has stopped transmission, RX data will be selected.
R30	31-27	Reserved	
	26	rx_en2	RX Enable for Channel 2. 0: Channel not enabled, all counters/flags will get reset 1: Channel is enabled
	25	rx_en1	RX Enable for Channel 1. 0: Channel not enabled, all counters/flags will get reset 1: Channel is enabled
	24	rx_en0	RX Enable for Channel 0. 0: Channel not enabled, all counters/flags will get reset 1: Channel is enabled
	23-0	Reserved	

Table 30-23. Peripheral I/F: TX

Register	Bits	Field Name	Description
R31	31-30	Reserved	
	29:22	Reserved	
	21	tx_global_reinit_active/busy2	If tx_global_reinit has been set, this status shows if the reinit has been complete. 1 = Active 0 = Done If tx_global_reinit has not been set, this status shows if the last bit is on the TX wire (i.e. ENDAT<m>_OUT). Note it does not mean the clock is off. 1 = Last bit is not done 0 = Last bit on tx wire Note that by using RX auto arm feature (i.e. CFG_ED_P<n>_CFG1_<m> [RX_EN_COUNTER] > 0), the observation of tx_global_reinit_active/busy is lost when rx_en<m> = 1. However, tx_global_reinit_active/busy can be used to determine when to enable RX during non-auto arm case.
	20	tx_global_go	TX global start of all channels. Note: FIFO must not be empty. If empty, transmit will not start.
	19	tx_global_reinit	Reinit all channels into default mode. This clears all flags and state machines for all channels. Note: Sequence should be assert tx_global_reinit then de-assert rx_en. This will ensure TX and RX are in reset/default state. User must assert this after the frame has been sent and TX is not busy.
	18	tx_channel_go	TX start the channel transmit (selected by tx_ch_sel). Note: FIFO must not be empty.
	17	unr2	Under Run Flag for Channel 2. This flag is only set when the tx_frame_count is nonzero and FIFO is empty at time to send data.
	16	ovr2	Over Run Flag for Channel 2
	15-14	Reserved	
	13	tx_global_reinit_active/busy1	If tx_global_reinit has been set, this status shows if the reinit has been complete. 1 = Active 0 = Done If tx_global_reinit has not been set, this status shows if the last bit is on the TX wire (i.e. ENDAT<m>_OUT). Note it does not mean the clock is off. 1 = Last bit is not done 0 = Last bit on tx wire Note that by using RX auto arm feature (i.e. CFG_ED_P<n>_CFG1_<m> [RX_EN_COUNTER] > 0), the observation of tx_global_reinit_active/busy is lost when rx_en<m> = 1. However, tx_global_reinit_active/busy can be used to determine when to enable RX during non-auto arm case.
	12:10	tx_fifo_sts1	TX FIFO occupancy status for Channel 1 0 = Empty 1 = 1 word 2 = 2 words 3 = 3 words 4 = Full 5-7 = Reserved
	9	unr1	Under Run Flag for Channel 1. This flag is only set when the tx_frame_count is nonzero and FIFO is empty at time to send data.
	8	ovr1	Over Run Flag for Channel 1
	7:6	Reserved	

Table 30-23. Peripheral I/F: TX (continued)

Register	Bits	Field Name	Description
R31	5	tx_global_reinit_active/busy0	If tx_global_reinit has been set, this status shows if the reinit has been complete. 1 = Active 0 = Done If tx_global_reinit has not been set, this status shows if the last bit is on the TX wire (i.e. ENDAT<m>_OUT). Note it does not mean the clock is off. 1 = Last bit is not done 0 = Last bit on tx wire Note that by using RX auto arm feature (i.e. CFG_ED_P<n>_CFG1_<m> [RX_EN_COUNTER] > 0), the observation of tx_global_reinit_active/busy is lost when rx_en<m> = 1. However, tx_global_reinit_active/busy can be used to determine when to enable RX during non-auto arm case.
	4:2	tx_fifo_sts0	TX FIFO occupancy status for Channel 0 0 = Empty 1 = 1 word 2 = 2 words 3 = 3 words 4 = Full 5-7 = Reserved
	1	unr0	Under Run Flag. This flag is only set when the tx_frame_count is nonzero and FIFO is empty at time to send data.
	0	ovr0	Over Run Flag for Channel 0

Table 30-23. Peripheral I/F: TX (continued)

Register	Bits	Field Name	Description
R30	31:21	Reserved	
	20:19	clk_mode	CLK_OUT mode. 0 = Free-running/stop-low. Clock will remain free-running until the receive module has received the number of bits indicated in rx_frame_counter and then the clock will stop low. 1 (default) = Free-running/stop-high. Clock will remain free-running until the receive module has received the number of bits indicated in rx_frame_counter and then the clock will stop high. Note this is the default/reset state, and a hardware reset or reinit will return clk_mode to this state. Note the initial state of the clock will be high, but the clock will not start until TX GO event. 2 = Free-run. NOTE: You must do a reinit to get out of this clock mode then you can update clk_mode to a different mode. Also if you do multiple TX GO, the 2nd go should have tst_delay and wire_delay zero since the clock is free running after the first go. 3= Stop high after transmit. Clock will run until the last TX bit is sent and stops high.
	18	Reserved	
	17:16	tx_ch_sel	TX channel select. 0 = Channel 0 1 = Channel 1 2 = Channel 2 3 = Reserved
	15:9	Reserved	
	7:0	tx_data	TX data for FIFO. Notes: FIFO transmits MSB first and is 32-bits deep. TX_FIFO_SWAP_BITS bit in the PRU-ICSS CFG register space can be used to flip the load order of bits. The FIFO has 2 modes of operation: 1. Preload and Go. This should be done for EnDAT and frames less than 32-bits. 2. Continuous mode. This should be done for frames bigger than 32-bits. In continuous mode, software needs to keep up with the line rate and ensure that the FIFO is never empty. When the FIFO is at 2 byte level, software needs to load the next 2 bytes. If software waits till the end of the empty state, it is possible to get the TX into a bad state. The FIFO state can be recovered via re-init.

30.4.1.2.6.3 Clock Generation

30.4.1.2.6.3.1 Configuration

The Peripheral I/F module has two source clock options, uart_clk (default) and ocp_clk. There are two independent clock dividers (div16) for the 1x and oversampling (OS) clocks, and each clock divider is configurable by two cascading dividers:

- TX_DIV_FACTOR and TX_DIV_FACTOR_FRAC for the 1x clock
- RX_DIV_FACTOR and RX_DIV_FACTOR_FRAC for the OS clock

The 1x clock is output on the ENDAT<m>_CLK signal. In TX mode, the output data is read from the TX FIFO at this 1x clock rate. The default value of this clock is high and the start and stop conditions for this clock are described in Section 30.4.1.2.6.3.2 and Section 30.4.1.2.6.3.3.

In RX mode, the input data is sampled at the OS clock rate. Note the OS clock rate divided by the 1x clock rate must equal RX_SAMPLE_SIZE.

Example clock rates and divisor values relative to the 192-MHz uart_clk clock source are shown in Table 30-24.

Table 30-24. Clock Rate Examples for 192-MHz uart_clk Clock Source

TX_DIV_FACTOR	1x Clock	RX_DIV_FACTOR	RX_DIV_FACTOR_FRAC	OS Clock	Oversample Factor
12	16 MHz	1	1.5	128 MHz	8x
16	12 MHz	2	1	96 MHz	8x
24	8 MHz	3	1	64 MHz	8x
32	6 MHz	4	1	48 MHz	8x
48	4 MHz	6	1	32 MHz	8x
96	2 MHz	12	1	16 MHz	8x
192	1 MHz	24	1	8 MHz	8x

30.4.1.2.6.3.2 Clock Output Start Conditions

This section describes the configurable start conditions for the ENDAT<m>_CLK. The software can completely control via CFG_ED_P_<n>_CFG0_<m> PRUn_EDm_CLK when ED_CLK_OUT_OVERRIDE_EN = 1. By default however, the PRU hardware will control the clocks as described in the following sections.

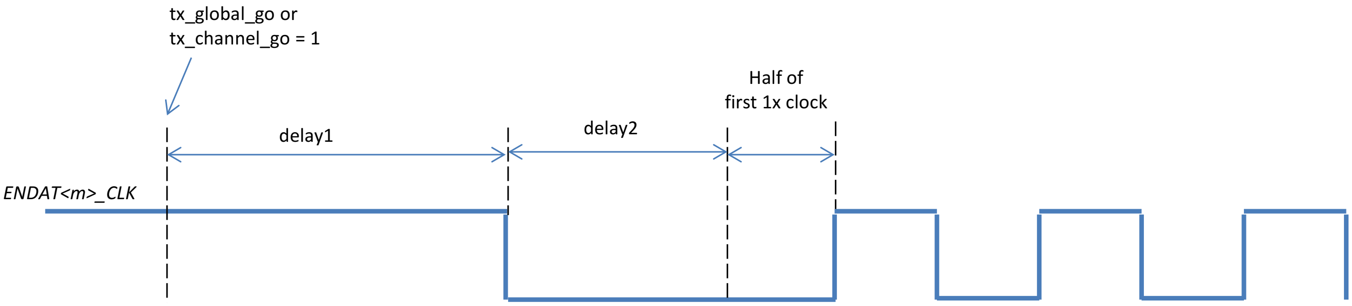
30.4.1.2.6.3.2.1 TX Mode (RX_EN = 0)

In TX mode, the ENDAT<m>_CLK begins after the firmware loads the TX FIFO and sets either r30[20] (tx_global_go) or r30[17:16] (tx_channel_go) to 1. After the “go” bit is set, the delay1 (wire delay) compensation counter for each channel begins. After delay1 is complete, ENDAT<m>_CLK is driven low and then the delay2 (tst) counter begins. After the delay2 counter expires, the ENDAT<m>>_CLK starts running (first low and then high). Therefore, first rising edge of ENDAT<m>_CLK (measured from the go bit) = delay1 (tx wire delay) + delay2 (tst_counter delay) + half of the 1x clock frequency (since the clock starts low).

Figure 30-15 shows the start condition for TX mode. As shown in the figure, the default value of clock is high. The PRU-ICSS CFG register space has additional MMRs for controlling the TX start timing delay values:

- Delay 1: CFG_ED_P<n>_CFG0<m> [TX_WIRE_DLY]
- Delay 2: CFG_ED_P<n>_CFG1<m> [TST_DELAY_COUNTER]

Figure 30-15. TX Mode Start Condition



30.4.1.2.6.3.2.2 RX Mode (RX_EN = 1)

In RX mode, the ENDAT<m>_CLK will start running whenever the RX_EN is set. Note that the PRU firmware in this mode is responsible for any delay conditions.

The hardware can also auto-enable RX mode at the end of a TX transaction. The CFG_ED_P<n>_CFG1<m> [RX_EN_COUNTER] is used to program a delay between the last TX bit sent and when the RX_EN is set.

30.4.1.2.6.3.3 Stop Conditions

The r30[20:19] (clk_mode[1:0]) value determines the stop condition for ENDAT<m>_CLK. There are 4 options available:

clk_mode_value	Description
0	Stop low on last RX frame
1	Stop high on last RX frame
2	Run continuously
3	Stop high on last TX bit

The last RX frame is configured by CFG_ED_P<n>_CFG0_<m> [rx_frame_size], and the last TX bit is configured by CFG_ED_P<n>_CFG0_<m> [tx_frame_size]. Each stop condition is shown in Figure 30-16 through Figure 30-19.

Figure 30-16. ENDAT<m>_CLK Stop High on Last RX Frame

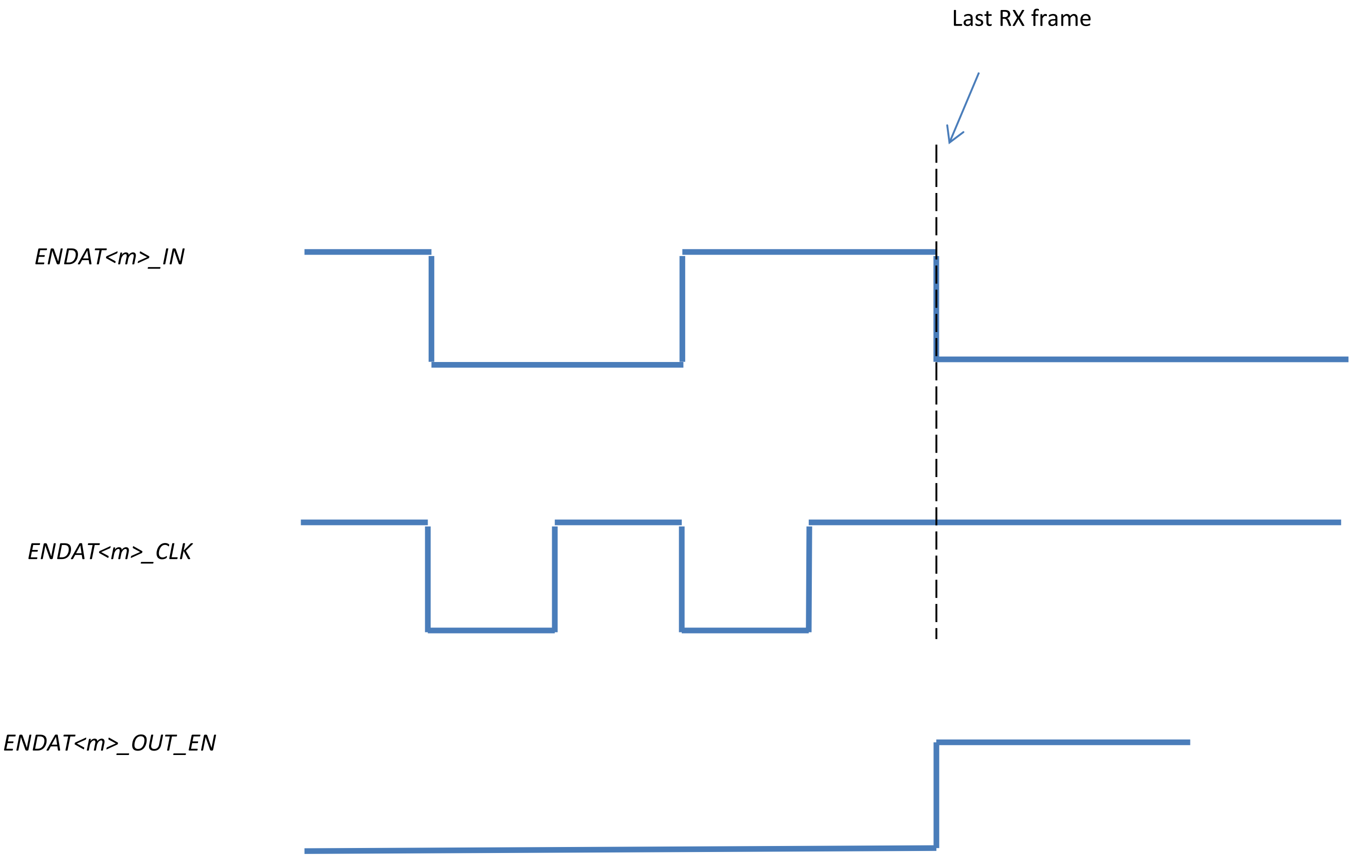


Figure 30-17. ENDAT<m>_CLK Stop Low on Last RX Frame

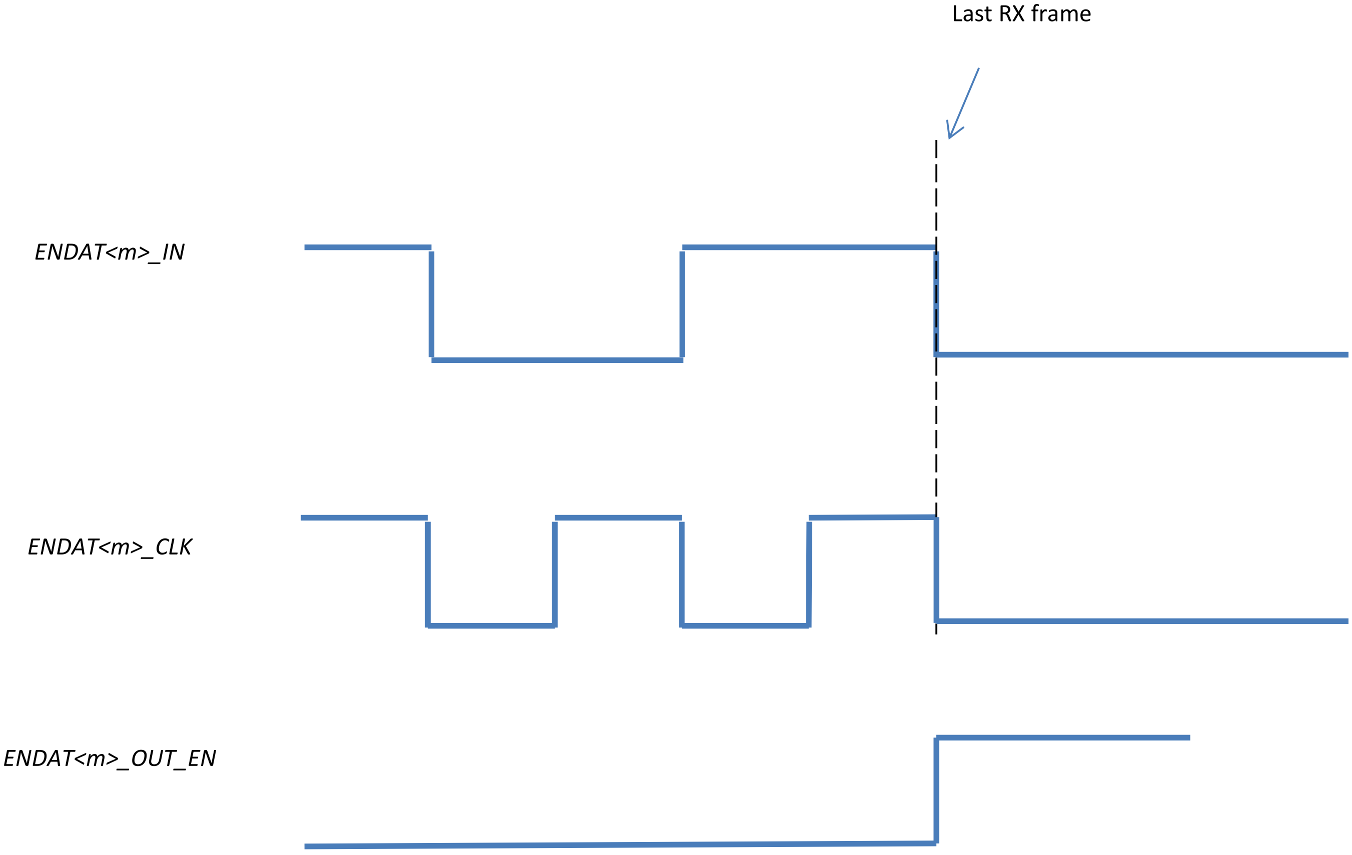


Figure 30-18. ENDAT<m>_CLK Run Continuously

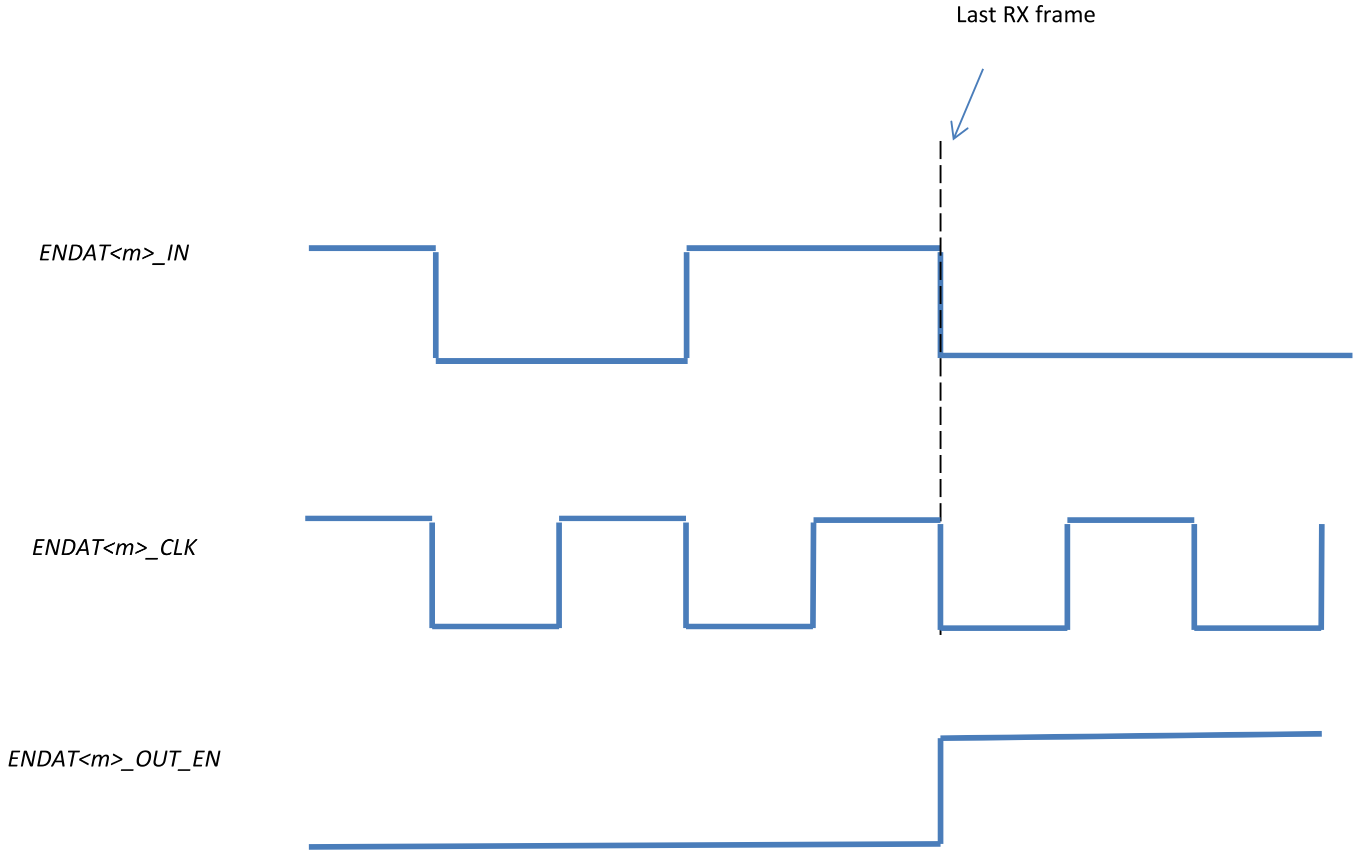
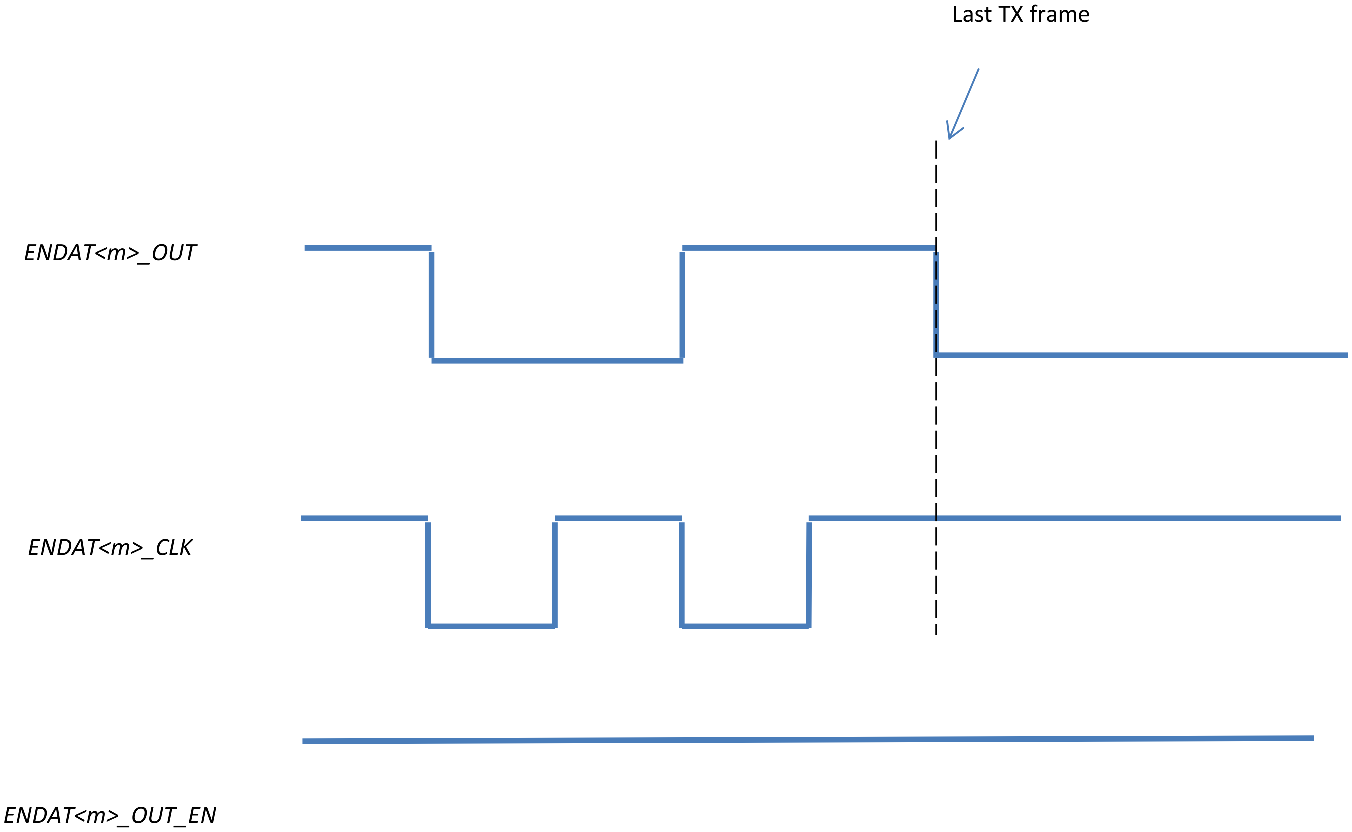


Figure 30-19. ENDAT<m>_CLK Stop High on Last TX Bit



30.4.1.2.6.4 Basic Programming Model

The following programming models assume that the PRU is configured for 3 Peripheral Mode (CFG_GPCFG<n> [PR1_PRI<n>_GP_MUX_SEL] = 1).

30.4.1.2.6.4.1 Clock Generation

Follow these steps to configure Peripheral I/F clocks using the HW control of the clock:

1. Select TX and RX clock sources:
   a. CFG_ED_P<n>TXCFG [TX_CLK_SEL] for the TX clock source
   b. CFG_ED_P<n>RXCFG [RX_CLK_SEL] for the RX clock source
2. Configure the 1x (TX) clock frequency:
   a. Write Division Factor to CFG_ED_P<n>_TXCFG {TX_DIV_FACTOR]
   b. Write Fraction division factor to CFG_ED_P<n>_TXCFG [TX_DIV_FACTOR_FRAC]
3. Configure the oversampling (RX) frequency and oversample size:
   a. Write Division Factor to CFG_ED_P<n>_RXCFG [RX_DIV_FACTOR]
   b. Write Fraction division factor to CFG_ED_P<n>_RXCFG [RX_DIV_FACTOR_FRAC]
   c. Write RX oversample size to CFG_ED_P<n>_RXCFG[RX_SAMPLE_SIZE]
4. Select the clk_mode to configure how the ENDAT<m>_CLK signal ends after TX/RX:
   a. Write to r30[20:19] (clk_mode). Note the clk_mode setting can also be changed per transaction.
5. Configure the wire, tst, and rx_en_counter delay values:
   a. CFG_ED_P<n>_CFG0_<m> [TX_WIRE_DLY] for wire delay
   b. CFG_ED_P<n>_CFG0_<m> [TST_DELAY_COUNTER] for tst delay

c. CFG_ED_P<n>_CFG1_<m> [RX_EN_COUNTER] for auto-delay between TX and RX

30.4.1.2.6.4.2 TX - Single Shot

Follow these steps to configure the Peripheral I/F channel(s) for a single shot transmission:

1. (Optional) Configure TX FIFO for MSB (default) or LSB:
    1. CFG_ED_P<n>_CFG0_<m> [TX_FIFO_SWAP_BITS]
2. Pre-load TX FIFO:
    1. Select TX channel by writing the desired channel number to R30[17:16] (tx_ch_sel)
    2. Write 1-4 bytes of data to r30[7:0] (tx_data). At each r30[7:0] write, data will be pushed into the FIFO.
    3. Repeat Steps 2a and 2b for all desired channels.
3. Configure TX frame size if less than 4 full bytes loaded into FIFO:
    1. CFG_ED_P<n>_CFG0_<m> [TX_FRAME_SIZE]
4. Push TX FIFO data to ENDAT<m>_OUT (see Section 30.4.1.2.6.3.2 for the ENDAT<m>_CLK and ENDAT<m>_OUT start time relationship);
    1. To start TX on all channels, set r31[20] = 1 (tx_global_go).
    2. To start TX on individual channel:
        - i. Select TX channel by writing the desired channel number to R30[17:16] (tx_ch_sel)
        - ii. Set R31[18] = 1 (tx_channel_go)
5. If CFG_ED_P<n>_CFG1_<m> [RX_EN_COUNTER] > 0, then the channel will automatically switch into RX mode. See Section 30.4.1.2.6.4.4 for an example of how to program and configure RX content.
6. If CFG_ED_P<n>_CFG1_<m> [RX_EN_COUNTER] = 0, poll either r31[21, 13, or 5] (tx_global_reinit_active/busy[2,1,0]) or CFG_ED_P<n>_TXCFG [TX_BUSY<m>] for when TX is complete

NOTE: The ENDAT<m>_CLK Peripheral I/F requires that ENDAT<m>_CLK be in a high state at the beginning of a new transaction. If the clock ended the single shot transmission in low state, then the clock needs to be reset before sending more data. The steps to reset ENDAT<m>_CLK are:

1. Set R31[19] = 1 (tx_global_reinit) to reset clock high
2. Wait until tx_busy<m> is cleared
3. Re-configure R30[20:19] (clk_mode), since reinit will reset the clk_mode to "Free-running/stop-high" mode

30.4.1.2.6.4.3 TX - Continuous FIFO Loading

Follow these steps to configure the Peripheral I/F channel(s) for a continuous loading transmission:

1. (Optional) Configure TX FIFO for MSB (default) or LSB:
    - a. CFG_ED_P<n>_CFG0_<m> [TX_FIFO_SWAP_BITS]
2. Pre-load TX FIFO:
    - a. Select TX channel by writing the desired channel number to r30[17:16] (tx_ch_sel)
    - b. Write 1-4 bytes of data to r30[7:0] (tx_data). At each r30[7:0] write, data will be pushed into the FIFO.
    - c. Repeat Steps 2a and 2b for all desired channels.
3. Configure TX frame size to continuously transmit the TX FIFO until empty:
    - a. Set CFG_ED_P<n>_CFG0_<m> [TX_FRAME_SIZE] = 0
4. Push TX FIFO data to ENDAT<m>_OUT (see Section 30.4.1.2.6.3.2 for the ENDAT<m>_CLK and ENDAT<m>_OUT start time relationship):
    - a. To start TX on all channels, set r31[20] = 1 (tx_global_go).

   b. To start TX on individual channel:
      i. Select TX channel by writing the desired channel number to r30[17:16] (tx_ch_sel)
      ii. Set r31[18] = 1 (tx_channel_go)
5. Monitor line rate and reload FIFO:
   a. Polling r31[xx, 12:10, 4:2] (tx_fifo_sts<m>)
   b. When FIFO level is at 2 bytes, load next 2 bytes of data (see Step 2). Do not let the FIFO get close to 0. Once the FIFO runs empty, the hardware will assume the PRU has reached end of the last transmit. Any new writes to the FIFO will NOT be sent until the software sends another tx_channel_go bit. Note there are also underrun and overrun error flags that can be monitored.
6. To end TX operation, do not send any new data to FIFO.
   a. If CFG_ED_P<n>_CFG1_<m> [RX_EN_COUNTER] > 0, then the channel will automatically switch into RX mode. See Section 30.4.1.2.6.4.4 for an example of how to program and configure RX content.
   b. If CFG_ED_P<n>_CFG1_<m> [RX_EN_COUNTER] = 0, poll either r31[21, 13, or 5] (tx_global_reinit_active/busy[2,1,0]) or CFG_ED_P<n>_TXCFG [TX_BUSY<m>] for when TX is complete

NOTE: The ENDAT<m>_CLK Peripheral I/F requires that ENDAT<m>_CLK be in a high state at the beginning of a new transaction. If the clock ended the continuous loading transmission in low state, then the clock needs to be reset before sending more data. The steps to reset ENDAT<m>_CLK are:

1. Set R31[19] = 1 (tx_global_reinit) to reset clock high
2. Wait until tx_busy<m> is cleared
3. Re-configure R30[20:19] (clk_mode), since reinit will reset the clk_mode to "Free-running/stop-high" mode

30.4.1.2.6.4.4 RX - Auto Arm or Non-Auto Arm

Follow these steps to configure the Peripheral I/F channel(s) to receive data:

1. Configure RX and frame size:
   a. CFG_ED_P<n>_CFG_<m> [RX_FRAME_SIZE]
2. To start ENDAT<m>_CLK:
   a. For the non-auto arm use case, set r30[26, 25, 24] = 1 (rx_en<m>)
   b. For the auto arm use case, rx_en<m> will be automatically enabled at the end of a TX operation when CFG_ED_P<n>_CFG1_<m> [RX_EN_COUNTER] > 0
3. RX FIFO will start filling on the first start bit (ENDAT<m>_IN = 1). The data will be captured on the positive edge of the ENDAT<m>_CLK and shifted into the LSB position of the 8-bit shadow register.
4. Poll for r31[26, 25, 24] (val<m>) assertion. The valid flag will be asserted when n bits of data (determined by CFG_ED_P<n>_RXCFG [RX_SAMPLE_SIZE]) have been collected.
5. Fetch data by reading r31[23-16, 15-8, 7-0] (rx_data_out<m>). The data will remain constant for one data frame, and PRU must read data and clear valid flag within this time. Otherwise, an overflow will occur – r31[29, 28, 27] (ovf<m>) = 1 - indicating that val<m> has been continuously asserted for longer than one data frame.
6. The clock will be stopped based on the r30[20:19] (clk_mode) configured before the start of the RX operation.
7. Clear r30[26, 25, 24] (rx_en<m>) to disable RX mode. All counters and flags will be reset.

30.4.1.3 Multiplier With Optional Accumulation (MPY/MAC)

Each PRU core has a designated unsigned multiplier with optional accumulation (MPY/MAC). The MAC has two modes of operation: Multiply Only or Multiply and Accumulate. The MAC is directly connected with the PRU internal registers R25–R29 and uses the broadside load/store PRU interface and XFR instructions to both control the mode of the MAC and import the multiplication results into the PRU.

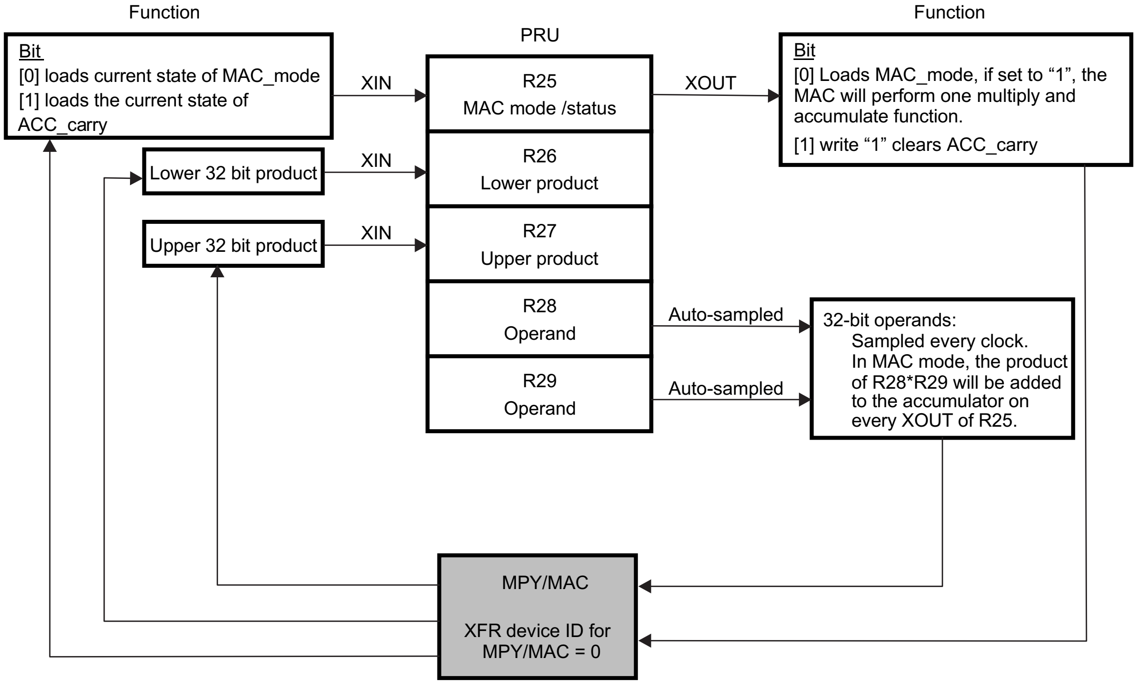
30.4.1.3.1 Features

- Configurable Multiply Only and Multiply and Accumulate functionality via PRU register R25
- 32-bit operands with direct connection to PRU registers R28 and R29
- 64-bit result (with carry flag) with direct connection to PRU registers R26 and R27
- PRU broadside interface and XFR instructions (XIN, XOUT) allow for importing multiplication results and initiating accumulate function

30.4.1.3.2 PRU and MPY/MAC Interface

The MAC directly connects with the PRU internal registers R25–R29 through use of the PRU broadside interface and XFR instructions. Figure 30-20 shows the functionality of each register.

Figure 30-20. Integration of the PRU and MPY/MAC



The XFR instructions (XIN and XOUT) are used to load/store register contents between the PRU core and the MAC. These instructions define the start, size, direction of the operation, and device ID. The device ID number corresponding to the MPY/MAC is shown in Table 30-25.

Table 30-25. MPY/MAC XFR ID

Device ID	Function
0	Selects MPY/MAC

The PRU register R25 is mapped to the MAC_CTRL_STATUS register (Table 30-26). The MAC's current status (MAC_mode and ACC_carry states) is loaded into R25 using the XIN command on R25. The PRU sets the MAC's mode and clears the ACC_carry using the XOUT command on R25.

Table 30-26. MAC_CTRL_STATUS Register (R25) Field Descriptions

Bit	Field	Value	Description
7-2	Reserved		Reserved
1	ACC_carry		Write 1 to clear
		0	64-bit accumulator carry has not occurred
		1	64-bit accumulator carry occurred
0	MAC_mode	0	Accumulation mode disabled and accumulator is cleared
		1	Accumulation mode enabled

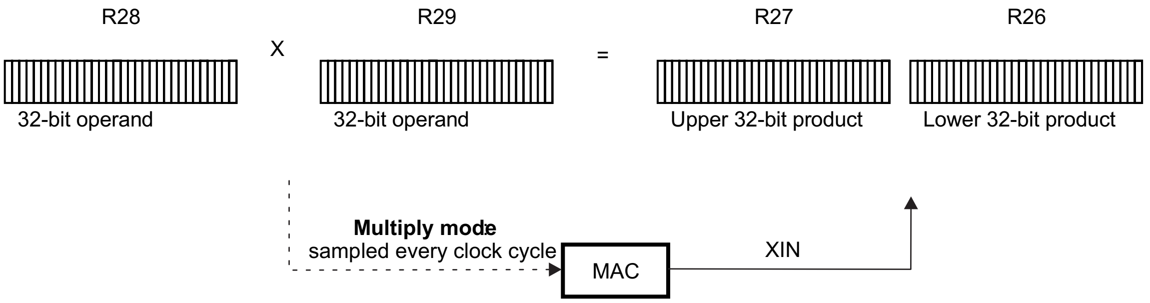
The two 32-bit operands for the multiplication are loaded into R28 and R29. These registers have a direction connection with the MAC, and the MAC samples these registers every clock cycle. Note, XOUT is not required to load the MAC. In multiply and accumulate mode, the product of R28*R29 will be added to the accumulator on every XOUT of R25.

The product from the MAC is linked to R26 (lower 32 bits) and R27 (upper 32 bits). The product is loaded into register R26 and R27 using XIN.

30.4.1.3.2.1 Multiply-Only Mode (Default State), MAC_mode = 0

On every clock cycle, the MAC multiplies the contents of R28 and R29.

Figure 30-21. Multiply-Only Mode Functional Diagram



The following steps are performed by the PRU firmware for multiply-only mode:

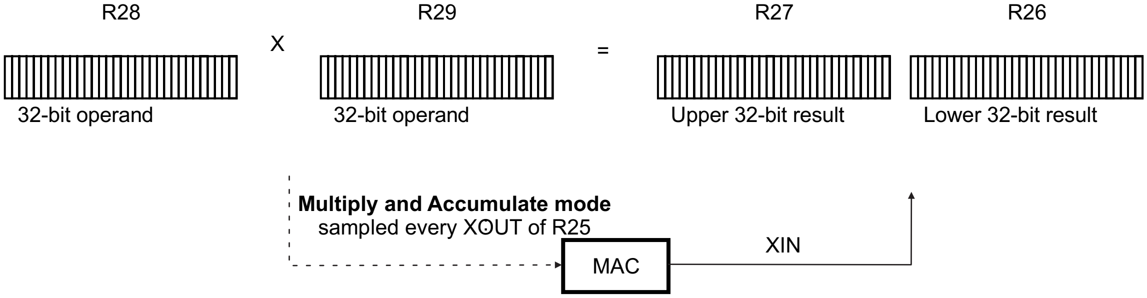
1. Enable multiply only MAC_mode.
   a. Clear R25[0] for multiply only mode.
   b. Store MAC_mode to MAC using XOUT instruction with the following parameters:
      - Device ID = 0
      - Base register = R25
      - Size = 1
2. Load operands into R28 and R29.
3. Delay at least 1 PRU cycle before executing XIN in step 4.
4. Load product into PRU using XIN instruction on R26, R27.

Repeat steps 2–4 for each new operand.

30.4.1.3.2.2 Multiply and Accumulate Mode, MAC_mode = 1

On every XOUT R25_reg[7:0] transaction, the MAC multiplies the contents of R28 and R29, adds the product to its accumulated result, and sets ACC_carry if an accumulation overflow occurs.

Figure 30-22. Multiply and Accumulate Mode Functional Diagram



The following steps are performed by the PRU firmware for multiply and accumulate mode:

1. Enable multiply and accumulate MAC_mode.
   a. Set R25[1:0] = 1 for accumulate mode.
   b. Store MAC_mode to MAC using XOUT instruction with the following parameters:
      - Device ID = 0
      - Base register = R25
      - Size = 1
2. Clear accumulator and carry flag.
   a. Set R25[1:0] = 3 to clear accumulator (R25[1]=1) and preserve accumulate mode (R25[0]=1).
   b. Store accumulator to MAC using XOUT instruction on R25.
3. Load operands into R28 and R29.
4. Multiply and accumulate, XOUT R25[1:0] = 1

   Repeat step 4 for each multiply and accumulate using same operands.

   Repeat step 3 and 4 for each multiply and accumulate for new operands.
5. Load the accumulated product into R26, R27, and the ACC_carry status into R25 using the XIN instruction.

Note: Steps one and two are required to set the accumulator mode and clear the accumulator and carry flag.

30.4.1.4 CRC16/32

The PRU0 and PRU1 cores of PRU-ICSS1/PRU-ICSS2 each have a designated CRC16/32 module.

In general, CRC adds error detection capability to communication systems. The CRC encoder appends redundant bits (or CRC bits) to the systematic data message. During reception of the data message, the received data is also encoded with the same CRC encoder. The 2 sets of CRC bits are compared together. If they atch, there were no transmission errors; and if they don’t match, a transmission error has been detected.

30.4.1.4.1 Features

CRC16/32 supports the following features:CRC16/32 supports the following features:

- Supports CRC32:
  – $x_{32} + x_{26} + x_{23} + x_{22} + x_{16} + x_{12} + x_{11} + x_{10} + x_8 + x_7 + x_5 + x_4 + x_2 + x + 1$
- Supports CRC16:
  – $x_{16} + x_{15} + x_2 + 1$
- PRU broadside interface and XFR instructions (XIN, XOUT) allow for importing CRC results and

executing accumulation function

30.4.1.4.2 PRU and CRC16/32 Interface

The CRC16/32 module directly connects with the PRU internal registers R25-R29 through use of the PRU broadside interface and XFR instructions. Table 30-27 shows the functionality of each register.

The XFR instructions (XIN and XOUT) are used to load/store register contents between the PRU core and the CRC16/32 module. These instructions define the start, size, direction of the operation, and device ID. The XFR device ID number corresponding to the CRC16/32 module is 1.

Table 30-27. CRC Register to PRU Port Mapping

CRC Register	R/W	Description	PRU Mapping
CRC_CFG	W	bit [0] CRC32_ENABLE: 0: CRC16 mode is selected. Hardware will auto-set init state of CRC_SEED to 0x0000_0000. Note CRC16 result value is only 16- bits 1: CRC32 mode is selected. Hardware will auto-set init state of CRC_SEED will be 0xffff_ffff. Always write all 4 bytes.	R25_reg
CRC_DATA_8_BFLIP	R	8-bit flip of CRC_DATA. CRC_DATA_8_BFLIP has the same byte order as CRC_DATA[31:0], but each byte has all bits flipped. CRC_DATA_32_FLIP[7:0] = CRC_DATA[0:7] CRC_DATA_32_FLIP[15:8] = CRC_DATA[8:15] CRC_DATA_32_FLIP[23:16] = CRC_DATA[16:23] CRC_DATA_32_FLIP[31:24] = CRC_DATA[24:31] For CRC16, only CRC_DATA_8_BFLIP[15:0] are valid. No auto reset on CRC_DATA_8_BFLIP read.	R27_reg
CRC_SEED	W	CRC SEED value. Hardware will auto-initialize the CRC_SEED value to 0x0000_0000 for CRC16 and 0xFFFF_FFFF for CRC32. Software only needs to initialize CRC_SEED if a different default value is required. Always write 4 bytes. Note when CRC_CFG[CRC32_ENABLE] is enabled, the hardware will switch the CRC_SEED value to 0xFFFF_FFFF. Reading the CRC_DATA register will reset the CRC value to the CRC_SEED state.	R28_reg
CRC_DATA_32_BFLIP	R	Full 32-bit flip of CRC_DATA CRC_DATA_32_BFLIP[0] = CRC_DATA[31] … CRC_DATA_32_BFLIP[31] = CRC_DATA[0] For CRC16, only CRC_DATA_32_BFLIP[31:16] are valid. No auto reset on CRC_DATA_32_BFLIP read.	R28_reg
CRC_DATA	RW	For Write, must use a fixed width throughout the session. The CRC module supports lower 8-bit, or lower 16-bit, or full 32-bit data widths. For Read, LSB or CRC_DATA[0] is first bit on the wire. Note for CRC16, only CRC_DATA[15:0] is valid. Hardware will delay CRC_DATA read operation up to 1 clock if it occurs back to back with a CRC_DATA write.	R29_reg

30.4.1.4.3 Programming Model

The following steps are performed by the PRU firmware to use the CRC module:

Step 1: Configuration (optional)

1. Configure CRC type: For CRC32 operation, set CRC32_ENABLE using XOUT instruction with the following parameters:
   - Device ID = 1
   - Base register = R25
   - Size = 1
2. Update CRC_SEED, if required using XOUT with the following parameters:
   - Device ID = 1

- Base Register = R28
- Size = 1 to 4

Step 2:

1. Load new CRC data into R29
2. Push CRC data to the CRC16/32 module using XOUT with the following parameters:
   - Device ID = 1
   - Base Register = R29
   - Size = 1 to 4
3. Load the accumulated CRC result into the PRU using the XIN instruction with the following parameters:
   - Device ID = 1
   - Base register = R29
   - Size = 4

Repeat Step 2, numbers 1 and 2 for each new CRC data.

NOTE: When a session starts, the PRU firmware must use the same write data width throughout the session.

30.4.1.5 PRU0/1 Scratch Pad

The PRU-ICSS supports a scratch pad with three independent banks accessible by the PRU cores. The PRU cores interact with the scratch pad through broadside load/store PRU interface and XFR instructions. The scratch pad can be used as a temporary place holder for the register contents of the PRU cores. Direct connection between the PRU cores is also supported for transferring register contents directly between the cores.

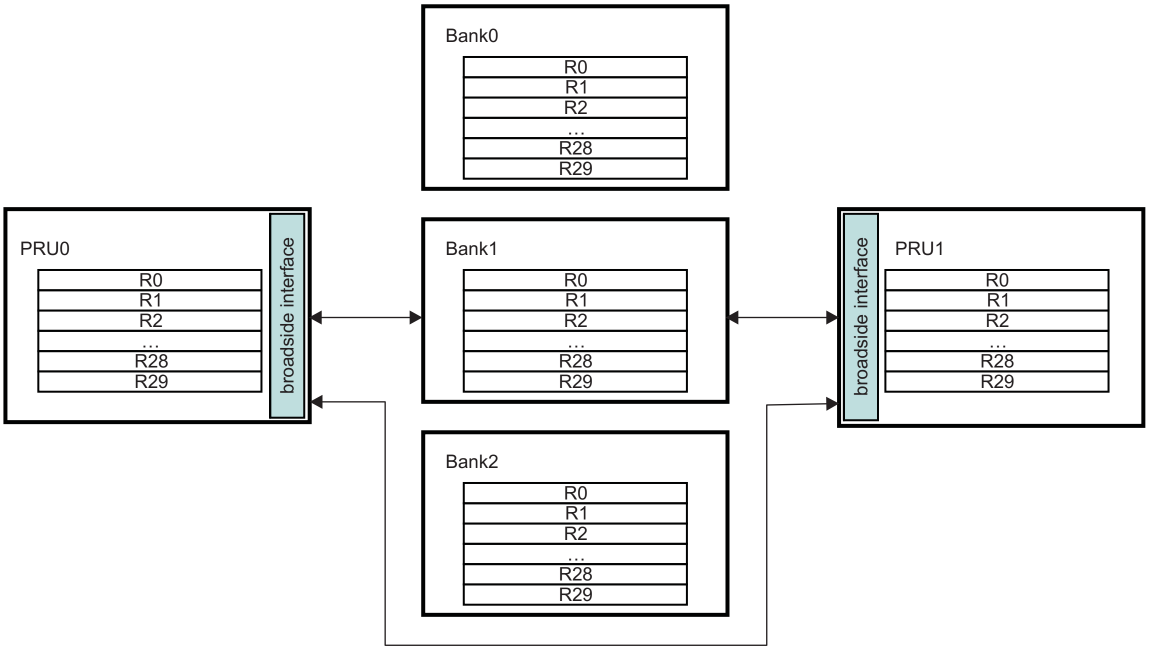
NOTE: Scratch pad may not be available for all subsystem instances. For a full list of available features for each subsystem instance, see Section 30.1.1, *Features*.

30.4.1.5.1 Features

The PRU-ICSS scratch pad supports the following features:

- Three scratch pad banks of 30, 32-bit registers (R29:0)
- Flexible load/store options
  - User-defined start byte and length of the transfer
  - Length of transfer ranges from one byte of a register to the entire register content (R29 to R0)
  - Simultaneous transactions supported between PRU0 ↔ Bank<n> and PRU1 ↔ Bank<m>
  - Direct connection of PRU0 → PRU1 or PRU1 → PRU0 for all registers R29–R0
- XFR instructions operate in one clock cycle
- Optional XIN/XOUT shift functionality allows remapping of registers (R<n> → R<m>) during load/store operation

Figure 30-23. Integration of PRU and Scratch Pad



30.4.1.5.2 Implementations and Operations

XFR instructions are used to load/store register contents between the PRU cores and the scratch pad banks. These instructions define the start, size, direction of the operation, and device ID. The device ID corresponds to the external source or destination (either a scratch pad bank or the other PRU core). The device ID numbers are shown in Table 30-28. Note the direct connect mode (device ID 14) can be used to synchronize the PRU cores. This mode requires the transmitting PRU core to execute XOUT and the receiving PRU core to execute XIN.

Table 30-28. Scratch Pad XFR ID

Device ID	Function
10	Selects Bank0
11	Selects Bank1
12	Selects Bank2
13	Reserved
14	Selects other PRU core (Direct connect mode)

A collision occurs when two XOUT commands simultaneously access the same asset or device ID. Table 30-29 shows the priority assigned to each operation when a collision occurs. In direct connect mode (device ID 14), any PRU transaction will be terminated if the stall is greater than 1024 cycles. This will generate the event pr<k>_xfr_timeout that is connected to INTC.

Table 30-29. Scratch Pad XFR Collision and Stall Conditions

Operation	Collision and Stall Handling
PRU<n> XOUT (→) bank[j]	If both PRU cores access the same bank simultaneously, PRU0 is given priority. PRU1 will temporarily stall until the PRU0 operation completes.
PRU<n> XOUT (→) PRU<m>	Direct connect mode requires the transmitting core (PRU<n>) to execute XOUT and the receiving core (PRU<m>) to execute XIN. If PRU<n> executes XOUT before PRU<m> executes XIN, then PRU<n> will stall until either PRU<m> executes XIN or the stall is greater than 1024 cycles.
PRU<m> XIN (←) PRU<n>	Direct connect mode requires the transmitting core (PRU<n>) to execute XOUT and the receiving core (PRU<m>) to execute XIN. If PRU<m> executes XIN before PRU<n> executes XOUT, then PRU<m> will stall until either PRU<n> executes XOUT or the stall is greater than 1024 cycles.

30.4.1.5.2.1 Optional XIN/XOUT Shift

The optional XIN/XOUT shift functionality allows register contents to be remapped or shifted within the destination's register space. For example, the contents of PRU0 R6-R8 could be remapped to Bank1 R10-12. The XIN/XOUT shift feature is not supported for direct connect mode, only for transfers between a PRU core and scratch pad bank.

The shift feature is enabled or disabled through the SPP register of the PRU-ICSS CFG register space. When enabled, R0[4:0] (internal to the PRU) defines the number of 32-bit registers in which content is shifted in the scratch pad bank. Note that scratch pad banks do not have registers R30 or R31.

30.4.1.5.2.2 Example Scratch Pad Operations

The following PRU firmware examples demonstrate the shift functionality. Note these assume the SHIFT_EN bit of the SPP register of the PRU-ICSS CFG register space has been set.

XOUT Shift By 4 Registers

Store R4:R7 to R8:R11 in bank0:

- Load 4 into R0.b0
- XOUT using the following parameters:
  - Device ID = 10
  - Base register = R4
  - Size = 16

XOUT Shift By 9 Registers, With Wrap Around

Store R25:R29 to R4:R8 in bank1:

- Load 9 into R0.b0
- XOUT using the following parameters:
  - Device ID = 11
  - Base register = R25
  - Size = 20

XIN Shift By 10 Registers

Load R14:R16 from bank2 to R4:R6

- Load 10 into R0.b0
- XIN using the following parameters:
  - Device ID = 12
  - Base register = R4
  - Size = 12

30.4.2 Interrupt Controller (INTC)

The PRU-ICSS interrupt controller (INTC) is an interface between interrupts coming from different parts of the system (referred to as system events; see Section 30.4.2.2) and the PRU-ICSS interrupt interface.

The PRU-ICSS INTC has the following features:

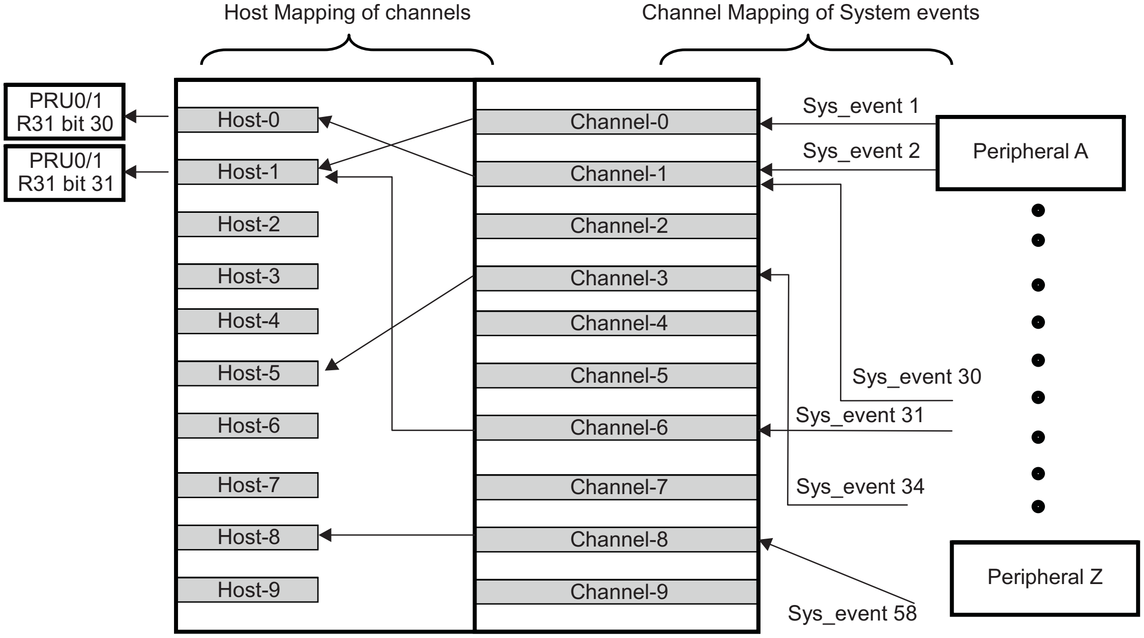
- Capturing up to 64 System Events
- Supports up to 10 interrupt channels.
- Generation of 10 Host Interrupts
  - 2 Host Interrupts for the PRUs.
  - 7 Host Interrupts exported from the PRU-ICSS for signaling the ARM interrupt controllers.
  - 1 Host Interrupt for the other PRU-ICSS.
- Each system event can be enabled and disabled.
- Each host event can be enabled and disabled.
- Hardware prioritization of events.

30.4.2.1 INTC Overview

The PRU-ICSS INTC supports up to 64 system events from different peripherals and PRUs to be mapped to 10 channels inside the INTC (see Figure 30-24). Interrupts from these 10 channels are further mapped to 10 Host Interrupts.

- Any of the 64 system events can be mapped to any of the 10 channels.
- Multiple interrupts can be mapped to a single channel.
- An interrupt should not be mapped to more than one channel.
- Any of the 10 channels can be mapped to any of the 10 host interrupts. It is recommended to map channel “x” to host interrupt “x”, where x is from 0 to 9
- A channel should not be mapped to more than one host interrupt
- For channels mapping to the same host interrupt, lower number channels have higher priority.
- For interrupts on same channel, priority is determined by the hardware interrupt number. The lower the interrupt number, the higher the priority.
- Host Interrupt 0 is connected to bit 30 in register 31 of PRU0 and PRU1.
- Host Interrupt 1 is connected to bit 31 in register 31 of PRU0 and PRU1.
- Host Interrupts 2-6 and 8-9 are exported from PRU-ICSS for signaling ARM interrupt controllers or other machines like EDMA.
- Host Interrupt 7 is exported from PRU-ICSS for signaling the other PRU-ICSS

Figure 30-24. Interrupt Controller Block Diagram



30.4.2.2 PRU-ICSS System Events

The PRU-ICSS system events are described as follows.

Table 30-30. PRU-ICSS0 System Events

Event Number	Signal Name (Standard Mode)	Source	Signal Name (MII_RT Mode)[1][2]
63	tpcc_int_pend_po1	EDMA3CC	
62	tpcc_errint_pend_po	EDMA3CC	
61	tptc_erint_pend_po	EDMA3TC0	
60	initiator_sinterrupt_q_n1	MAILBOX0- mail_u1_irq (mailbox interrupt for pru0)	
59	initiator_sinterrupt_q_n2	MAILBOX1 - mail_u2_irq (mailbox interrupt for pru1)	
58	Emulation Suspend Signal (software use)	DEBUGSS	
57	pwm_trip_zone	PWMSS3, PWMSS4, PWMSS5, EPWM	
56	pr1_host_intr5_intr_pend	PRU-ICSS1 Host Interrupt 7 (exported as pr1_host[5])	
55	mcasp_x_intr_pend	MCASP0, TX	(pr1_mii1_col and pr1_mii1_txen) (external)
54	mcasp_r_intr_pend	MCASP0, RX	PRU-ICSS0 PRU1_RX_EOF
53	gen_intr_pend	ADC0	PRU-ICSS0 MDIO_MII_LINK[1]
52	nirq	UART2	PRU-ICSS0 PORT1_TX_OVERFLOW
51	nirq	UART0	PRU-ICSS0 PORT1_TX_UNDERFLOW
50	c2_rx_thresh_pend	CPSW	PRU-ICSS0 PRU1_RX_OVERFLOW

---

[1] MII_RT mode is selected through the MII_RT register in the PRU-ICSS0 CFG register space.

[2] Signals 63-56 and 31-0 for MII_RT Mode are the same as for Standard Mode.

Table 30-30. PRU-ICSS0 System Events (continued)

Event Number	Signal Name (Standard Mode)	Source	Signal Name (MII_RT Mode)[1][2]
49	c2_rx_pend	CPSW	PRU-ICSS0 PRU1_RX_NIBBLE_ODD
48	c2_tx_pend	CPSW	PRU-ICSS0 PRU1_RX_CRC
47	c2_misc_pend	CPSW	PRU-ICSS0 PRU1_RX_SOF
46	epwm_intr_intr_pend	PWMSS4_EPWM	PRU-ICSS0 PRU1_RX_SFD
45	eqep_intr_intr_pend	PWMSS0_EQEP	PRU-ICSS0 PRU1_RX_ERR32
44	SINTERRUPTN	MCSPI0	PRU-ICSS0 PRU1_RX_ERR
43	epwm_intr_intr_pend	PWMSS3_EPWM	(pr1_mii0_col and pr1_mii0_txen) (external)
42	ecap_intr_intr_pend	PWMSS0_ECAP	PRU-ICSS0 PRU0_RX_EOF
41	POINTRPEND	I2C0	PRU-ICSS0 MDIO_MII_LINK[0]
40	dcan_intr	DCAN0	PRU-ICSS0 PORT0_TX_OVERFLOW
39	dcan_int1	DCAN0	PRU-ICSS0 PORT0_TX_UNDERFLOW
38	dcan_uerr	DCAN0	PRU-ICSS0 PRU0_RX_OVERFLOW
37	epwm_intr_intr_pend	PWMSS5_EPWM	PRU-ICSS0 PRU0_RX_NIBBLE_ODD
36	ecap_intr_intr_pend	PWMSS2_ECAP	PRU-ICSS0 PRU0_RX_CRC
35	ecap_intr_intr_pend	PWMSS1_ECAP	PRU-ICSS0 PRU0_RX_SOF
34	gen_intr_pend	ADC1	PRU-ICSS0 PRU0_RX_SFD
33	intr_pend	QSPI	PRU-ICSS0 PRU0_RX_ERR32
32	nirq	UART3	PRU-ICSS0 PRU0_RX_ERR

Table 30-30. PRU-ICSS0 System Events (continued)

Event Number	Signal Name (Standard Mode)	Source	Signal Name (MII_RT Mode)[1][2]
31	pr0_pru_mst_intr15_intr_req	PRU_ICSS0_PRU0, PRU_ICSS0_PRU1	PRU-ICSS0 Internal Interrupts
30	pr0_pru_mst_intr14_intr_req	PRU_ICSS0_PRU0, PRU_ICSS0_PRU1	
29	pr0_pru_mst_intr13_intr_req	PRU_ICSS0_PRU0, PRU_ICSS0_PRU1	
28	pr0_pru_mst_intr12_intr_req	PRU_ICSS0_PRU0, PRU_ICSS0_PRU1	
27	pr0_pru_mst_intr11_intr_req	PRU_ICSS0_PRU0, PRU_ICSS0_PRU1	
26	pr0_pru_mst_intr10_intr_req	PRU_ICSS0_PRU0, PRU_ICSS0_PRU1	
25	pr0_pru_mst_intr9_intr_req	PRU_ICSS0_PRU0, PRU_ICSS0_PRU1	
24	pr0_pru_mst_intr8_intr_req	PRU_ICSS0_PRU0, PRU_ICSS0_PRU1	
23	pr0_pru_mst_intr7_intr_req	PRU_ICSS0_PRU0, PRU_ICSS0_PRU1	
22	pr0_pru_mst_intr6_intr_req	PRU_ICSS0_PRU0, PRU_ICSS0_PRU1	
21	pr0_pru_mst_intr5_intr_req	PRU_ICSS0_PRU0, PRU_ICSS0_PRU1	
20	pr0_pru_mst_intr4_intr_req	PRU_ICSS0_PRU0, PRU_ICSS0_PRU1	
19	pr0_pru_mst_intr3_intr_req	PRU_ICSS0_PRU0, PRU_ICSS0_PRU1	
18	pr0_pru_mst_intr2_intr_req	PRU_ICSS0_PRU0, PRU_ICSS0_PRU1	
17	pr0_pru_mst_intr1_intr_req	PRU_ICSS0_PRU0, PRU_ICSS0_PRU1	
16	pr0_pru_mst_intr0_intr_req	PRU_ICSS0_PRU0, PRU_ICSS0_PRU1	
15	pr0_ecap_intr_req	PRU_ICSS0_ECAP	
14	sync0_out_pend	PRU_ICSS0 IEP_SYNC	
13	sync1_out_pend	PRU_ICSS0 IEP_SYNC	
12	latch0_in (input to PRU-ICSS)	PRU_ICSS0 IEP_SYNC	
11	latch1_in (input to PRU-ICSS)	PRU_ICSS0 IEP_SYNC	
10	pdi_wd_exp_pend	PRU_ICSS0 IEP_WD	
9	pd_wd_exp_pend	PRU_ICSS0 IEP_WD	
8	pr0_digio_event_req	PRU_ICSS0_IEP_DIGIO	
7	pr0_iep_tim_cap_cmp_pend	PRU_ICSS0_IEP_TIMER	
6	pr0_uart1_uint_intr_req	PRU_ICSS0_UART	
5	pr0_uart1_utxevt_intr_req	PRU_ICSS0_UART	
4	pr0_uart1_urxevt_intr_req	PRU_ICSS0_UART	
3	pr0_xfr_timeout	PRU_ICSS0_XFR_TIMEOUT	
2	pr0_pru1_r31_status_cnt16	PRU_ICSS0_PRU1	
1	pr0_pru0_r31_status_cnt16	PRU_ICSS0_PRU0	
0	pr0_parity_err_intr_pend	PRU_ICSS0_MEM	

Table 30-31. PRU-ICSS1 System Events

Event Number	Signal Name (Standard Mode)	Source	Signal Name (MII_RT Mode)(1)(2)
63	tpcc_int_pend_po1	EDMA3CC	
62	tpcc_errint_pend_po	EDMA3CC	
61	tptc_erint_pend_po	EDMA3TC0	
60	initiator_sinterrupt_q_n1	MAILBOX0- mail_u1_irq (mailbox interrupt for pru0)	
59	initiator_sinterrupt_q_n2	MAILBOX1 - mail_u2_irq (mailbox interrupt for pru1)	
58	Emulation Suspend Signal (software use)	DEBUGSS	
57	pwm_trip_zone	PWMSS3, PWMSS4, PWMSS5, EPWM	
56	pr0_host_intr5_intr_pend	PRU-ICSS0 Host Interrupt 7 (exported as pr0_host[5])	
55	mcasp_x_intr_pend	MCASP0, TX	pr1_mii1_crs(external)
54	mcasp_r_intr_pend	MCASP0, RX	PRU-ICSS1 PRU1_RX_EOF
53	gen_intr_pend	ADC0	PRU-ICSS1 MDIO_MII_LINK[1]
52	nirq	UART2	PRU-ICSS1 PORT1_TX_OVERFLOW
51	nirq	UART0	PRU-ICSS1 PORT1_TX_UNDERFLOW
50	c2_rx_thresh_pend	CPSW	PRU-ICSS1 PRU1_RX_OVERFLOW
49	c2_rx_pend	CPSW	PRU-ICSS1 PRU1_RX_NIBBLE_ODD
48	c2_tx_pend	CPSW	PRU-ICSS1 PRU1_RX_CRC
47	c2_misc_pend	CPSW	PRU-ICSS1 PRU1_RX_SOF
46	epwm_intr_intr_pend	PWMSS1_EPWM	PRU-ICSS1 PRU1_RX_SFD
45	eqep_intr_intr_pend	PWMSS0_EQEP	PRU-ICSS1 PRU1_RX_ERR32
44	SINTERRUPTN	MCSPI0	PRU-ICSS1 PRU1_RX_ERR
43	epwm_intr_intr_pend	PWMSS0_EPWM	pr1_mii0_crs(external)
42	ecap_intr_intr_pend	PWMSS0_ECAP	PRU-ICSS1 PRU0_RX_EOF
41	POINTRPEND	I2C0	PRU-ICSS1 MDIO_MII_LINK[0]
40	dcan_intr	DCAN0	PRU-ICSS1 PORT0_TX_OVERFLOW
39	dcan_int1	DCAN0	PRU-ICSS1 PORT0_TX_UNDERFLOW
38	dcan_uerr	DCAN0	PRU-ICSS1 PRU0_RX_OVERFLOW
37	epwm_intr_intr_pend	PWMSS2_EPWM	PRU-ICSS1 PRU0_RX_NIBBLE_ODD
36	ecap_intr_intr_pend	PWMSS2_ECAP	PRU-ICSS1 PRU0_RX_CRC
35	ecap_intr_intr_pend	PWMSS1_ECAP	PRU-ICSS1 PRU0_RX_SOF
34	gen_intr_pend	ADC1	PRU-ICSS1 PRU0_RX_SFD
33	intr_pend	QSPI	PRU-ICSS1 PRU0_RX_ERR32
32	nirq	UART3	PRU-ICSS1 PRU0_RX_ERR

---

(1) MII_RT mode is selected through the PRU-ICSS1 MII_RT register in the CFG register space.

(2) Signals 63-56 and 31-0 for MII_RT Mode are the same as for Standard Mode.

Table 30-31. PRU-ICSS1 System Events (continued)

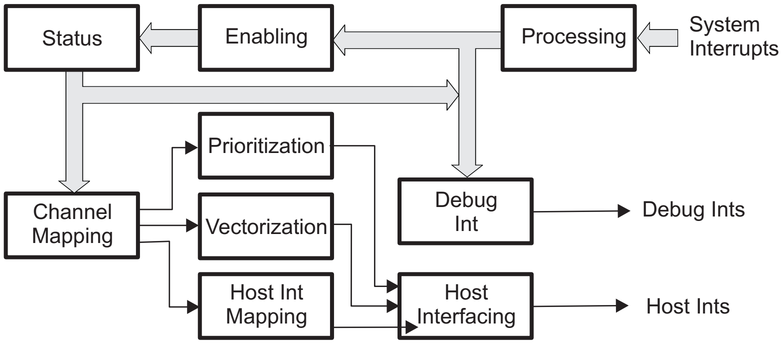
Event Number	Signal Name (Standard Mode)	Source	Signal Name (MII_RT Mode)[(1)][(2)]
31	pr1_pru_mst_intr15_intr_req	PRU_ICSS1_PRU0, PRU_ICSS1_PRU1	PRU-ICSS1 Internal Interrupts
30	pr1_pru_mst_intr14_intr_req	PRU_ICSS1_PRU0, PRU_ICSS1_PRU1	
29	pr1_pru_mst_intr13_intr_req	PRU_ICSS1_PRU0, PRU_ICSS1_PRU1	
28	pr1_pru_mst_intr12_intr_req	PRU_ICSS1_PRU0, PRU_ICSS1_PRU1	
27	pr1_pru_mst_intr11_intr_req	PRU_ICSS1_PRU0, PRU_ICSS1_PRU1	
26	pr1_pru_mst_intr10_intr_req	PRU_ICSS1_PRU0, PRU_ICSS1_PRU1	
25	pr1_pru_mst_intr9_intr_req	PRU_ICSS1_PRU0, PRU_ICSS1_PRU1	
24	pr1_pru_mst_intr8_intr_req	PRU_ICSS1_PRU0, PRU_ICSS1_PRU1	
23	pr1_pru_mst_intr7_intr_req	PRU_ICSS1_PRU0, PRU_ICSS1_PRU1	
22	pr1_pru_mst_intr6_intr_req	PRU_ICSS1_PRU0, PRU_ICSS1_PRU1	
21	pr1_pru_mst_intr5_intr_req	PRU_ICSS1_PRU0, PRU_ICSS1_PRU1	
20	pr1_pru_mst_intr4_intr_req	PRU_ICSS1_PRU0, PRU_ICSS1_PRU1	
19	pr1_pru_mst_intr3_intr_req	PRU_ICSS1_PRU0, PRU_ICSS1_PRU1	
18	pr1_pru_mst_intr2_intr_req	PRU_ICSS1_PRU0, PRU_ICSS1_PRU1	
17	pr1_pru_mst_intr1_intr_req	PRU_ICSS1_PRU0, PRU_ICSS1_PRU1	
16	pr1_pru_mst_intr0_intr_req	PRU_ICSS1_PRU0, PRU_ICSS1_PRU1	
15	pr1_ecap_intr_req	PRU_ICSS1_ECAP	
14	pr1_sync0_out_pend	PRU_ICSS1_IEP_SYNC	
13	pr1_sync1_out_pend	PRU_ICSS1_IEP_SYNC	
12	pr1_latch0_in	PRU_ICSS1_IEP_SYNC	
11	pr1_latch1_in	PRU_ICSS1_IEP_SYNC	
10	pr1_pdi_wd_exp_pend	PRU_ICSS1_IEP_WD	
9	pr1_pd_wd_exp_pend	PRU_ICSS1_IEP_WD	
8	pr1_digio_event_req	PRU_ICSS1_IEP_DIGIO	
7	pr1_iep_tim_cap_cmp_pend	PRU_ICSS1_IEP_TIMER	
6	pr1_uart1_uint_intr_req	PRU_ICSS1_UART	
5	pr1_uart1_utxevt_intr_req	PRU_ICSS1_UART	
4	pr1_uart1_urxevt_intr_req	PRU_ICSS1_UART	
3	pr1_xfr_timeout	PRU_ICSS1_XFR_TIMEOUT	
2	pr1_pru1_r31_status_cnt16	PRU_ICSS1_PRU1	
1	pr1_pru0_r31_status_cnt16	PRU_ICSS1_PRU0	
0	pr1_parity_err_intr_pend	PRU_ICSS1_MEM	

30.4.2.3 INTC Methodology

The INTC module controls the system event mapping to the host interrupt interface. System events are generated by the device peripherals or PRUs. The INTC receives the system events and maps them to internal channels. The channels are used to group interrupts together and to prioritize them. These channels are then mapped onto the host interrupts. Interrupts from the system side are active high in polarity. They are also pulse type of interrupts.

The INTC encompasses many functions to process the system events and prepare them for the host interface. These functions are: processing, enabling, status, channel mapping, host interrupt mapping, prioritization, and host interfacing. Figure 30-25 illustrates the flow of system events through the functions to the host. The following subsections describe each part of the flow.

Figure 30-25. Flow of System Events to Host



30.4.2.3.1 Interrupt Processing

This block does the following tasks:

- Synchronization of slower and asynchronous interrupts
- Conversion of polarity to active high
- Conversion of interrupt type to pulse interrupts

After the processing block, all interrupts will be active high pulses.

30.4.2.3.2 Interrupt Enabling

The next stage of INTC is to enable system events based on programmed settings. The following sequence is to be followed to enable interrupts:

- Enable required system events: System events that are required to get propagated to host are to be enabled individually by writing to IDX field in the system event enable indexed set register (EISR). The event to enable is the index value written. This sets the Enable Register bit of the given index.
- Enable required host interrupts: By writing to the IDX field in the host interrupt enable indexed set register (HIEISR), enable the required host interrupts. The host interrupt to enable is the index value written. This enables the host interrupt output or triggers the output again if that host interrupt is already enabled.
- Enable all host interrupts: By setting the EN bit in the global enable register (GER) to 1, all host interrupts will be enabled. Individual host interrupts are still enabled or disabled from their individual enables and are not overridden by the global enable.

30.4.2.3.3 Interrupt Status Checking

The next stage is to capture which system events are pending. There are two kinds of pending status: raw status and enabled status. Raw status is the pending status of the system event without regards to the enable bit for the system event. Enabled status is the pending status of the system events with the enable bits active. When the enable bit is inactive, the enabled status will always be inactive. The enabled status of system events is captured in system event status enabled/clear registers (SECR1-SECR2).

Status of system event 'N' is indicated by the Nth bit of SECR1-SECR2. Since there are 64 system events, two 32-bit registers are used to capture the enabled status of events. The pending status reflects whether the system event occurred since the last time the status register bit was cleared. Each bit in the status register can be individually cleared.

30.4.2.3.4 Interrupt Channel Mapping

The INTC has 10 internal channels to which enabled system events can be mapped. Channel 0 has highest priority and channel 9 has the lowest priority. Channels are used to group the system events into a smaller number of priorities that can be given to a host interface with a very small number of interrupt inputs.

When multiple system events are mapped to the same channel their interrupts are ORed together so that when one or more is active the output is active. The channel map registers (CMRm) define the channel for each system event. There is one register per 4 system events; therefore, there are 16 channel map registers for a system of 64 events. The channel for each system event can be set using these registers.

30.4.2.3.4.1 Host Interrupt Mapping

The hosts can be the PRUs or ARM CPU. The 10 channels from the INTC can mapped to any of the 10 Host interrupts. The Host map registers (HMR*m*) define the channel for each system event. There is one register per 4 channels; therefore, there are 3 host map registers for 10 channels. When multiple channels are mapped to the same host interrupt, then prioritization is done to select which event is in the highest-priority channel and which should be sent first to the host.

30.4.2.3.4.2 Interrupt Prioritization

The next stage of the INTC is prioritization. Since multiple events can feed into a single channel and multiple channels can feed into a single host interrupt, it is to read the status of all system events to determine the highest priority event that is pending. The INTC provides hardware to perform this prioritization with a given scheme so that software does not have to do this. There are two levels of prioritizations:

- The first level of prioritization is between the active channels for a host interrupt. Channel 0 has the highest priority and channel 9 has the lowest. So the first level of prioritization picks the lowest numbered active channel.
- The second level of prioritization is between the active system events for the prioritized channel. The system event in position 0 has the highest priority and system event 63 has the lowest priority. So the second level of prioritization picks the lowest position active system event.

This is the final prioritized system event for the host interrupt and is stored in the global prioritized index register (GPIR). The highest priority pending event with respect to each host interrupts can be obtained using the host interrupt prioritized index registers (HIPIRn).

30.4.2.3.5 Interrupt Nesting

The INTC can also perform a nesting function in its prioritization. Nesting is a method of enabling certain interrupts (usually higher-priority interrupts) when an interrupt is taken so that only those desired interrupts can trigger to the host while it is servicing the current interrupt. The typical usage is to nest on the current interrupt and disable all interrupts of the same or lower priority (or channel). Then the host will only be interrupted from a higher priority interrupt.

The nesting is done in one of three methods:

1. Nesting for all host interrupts, based on channel priority: When an interrupt is taken, the nesting level is set to its channel priority. From then, that channel priority and all lower priority channels will be disabled from generating host interrupts and only higher priority channels are allowed. When the interrupt is completely serviced, the nesting level is returned to its original value. When there is no interrupt being serviced, there are no channels disabled due to nesting. The global nesting level register (GNLR) allows the checking and setting of the global nesting level across all host interrupts. The nesting level is the channel (and all of lower priority channels) that are nested out because of a current interrupt.
2. Nesting for individual host interrupts, based on channel priority: Always nest based on channel priority

for each host interrupt individually. When an interrupt is taken on a host interrupt, then, the nesting level is set to its channel priority for just that host interrupt, and other host interrupts do not have their nesting affected. Then for that host interrupt, equal or lower priority channels will not interrupt the host but may on other host interrupts if programmed. When the interrupt is completely serviced the nesting level for the host interrupt is returned to its original value. The host interrupt nesting level registers (HINLR1 and HINLR2) display and control the nesting level for each host interrupt. The nesting level controls which channel and lower priority channels are nested. There is one register per host interrupt.

3. Software manually performs the nesting of interrupts. When an interrupt is taken, the software will disable all the host interrupts, manually update the enables for any or all the system events, and then re-enable all the host interrupts. This now allows only the system events that are still enabled to trigger to the host. When the interrupt is completely serviced the software must reverse the changes to re-enable the nested out system events. This method requires the most software interaction but gives the most flexibility if simple channel-based nesting mechanisms are not adequate.

30.4.2.3.6 Interrupt Status Clearing

After servicing the event (after execution of the ISR), event status is to be cleared. If a system event status is not cleared, then another host interrupt may not be triggered or another host interrupt may be triggered incorrectly. It is also essential to clear all system events before the PRU is halted as the PRU does not power down unless all the event status bits are cleared. For clearing the status of an event, whose event number is N, write a 1 to the Nth bit position in the system event status enabled/clear registers (SECR1-SECR2). System event N can also be cleared by writing the value N into the system event status indexed clear register (SICR).

30.4.2.4 Interrupt Disabling

At any time, if any event is not to be propagated to the host, then that event should be disabled. For disabling an event whose event number is N, write a 1 to the Nth bit in the system event enable clear registers (ECR1-ECR2). system event N can also be disabled by writing the value N in the system event enable indexed clear register (EICR).

30.4.2.5 INTC Basic Programming Model

Follow these steps to configure the interrupt controller.

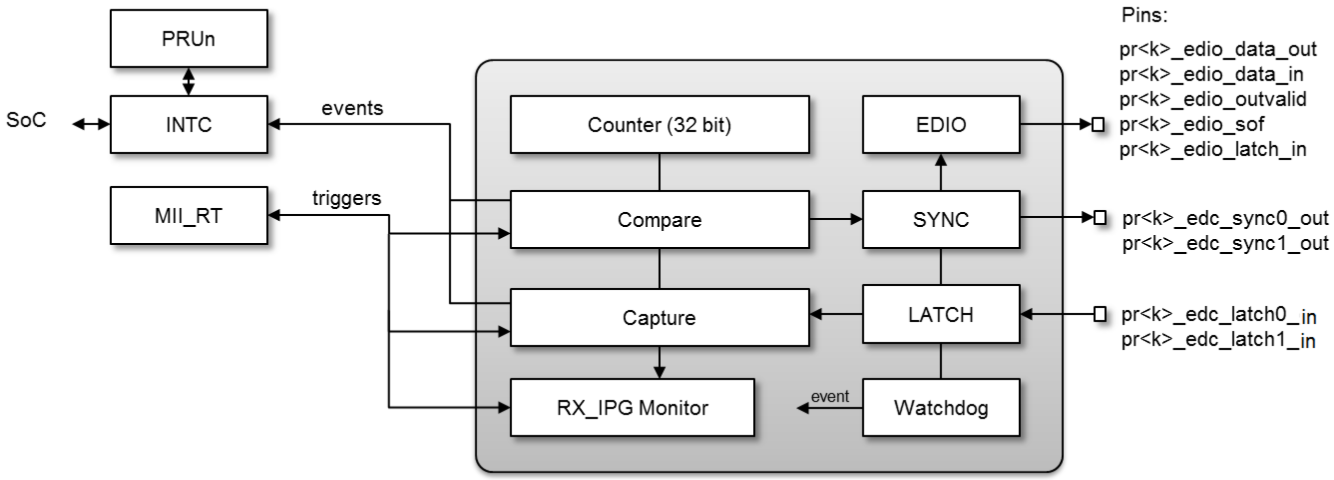
1. Set polarity and type of system event through the System Event Polarity Registers (SIPR1 and SPIR2) and the System Event Type Registers (SITR1 and SITR2). Polarity of all system events is always high. Type of all system events is always pulse.
2. Map system event to INTC channel through CHANMAP registers.
3. Map channel to host interrupt through HOSTMAP registers. Recommend channel “x” be mapped to host interrupt “x”.
4. Clear system event by writing 1s to SECR registers.
5. Enable host interrupt by writing index value to HIER register.
6. Enable interrupt nesting if desired.
7. Globally enable all interrupts through GER register.

30.4.3 Industrial Ethernet Peripheral (IEP)

The industrial ethernet peripheral (IEP) performs hardware work required for industrial ethernet functions. The IEP module features an industrial ethernet timer with 16 compare events, industrial ethernet sync generator and latch capture, industrial ethernet watchdog timer, and a digital I/O port (DIGIO). The IEP functional block diagram is shown in Figure 30-26.

The industrial ethernet peripheral (IEP) is intended to do the hardware work required for industrial ethernet functions of the timer/system timer, sync 0/1 generation, latch0/1 capture, watchdog timer (WD), and Digital I/O port (DIGIO). The IEP functional block diagram is shown in Figure 30-26. The following chapter will introduce these in register view.

Figure 30-26. Industrial Ethernet Peripheral Block Diagram



(1) Internal signal wire

(2) External pin input/output

A For SYNC0 trigger of start time.

B For SYNC1 trigger only in independent mode.

30.4.3.1 IEP Clock Source

The IEP has a selectable module input clock. The clock source is selected by the state of the IEPCLK.OCP_EN bit within the PRU-ICSS CFG register space. Two clock sources are supported for the IEP input clock:

- iep_clk (default): Runs at 200 MHz
- ocp_clk

Switching from iep_clk to ocp_clk is done by writing 1 to the IEPCLK.OCP_EN bit. This is a one time configuration step before enabling the IEP function. Switching back from ocp_clk to iep_clk is only supported through a hardware reset of the PRU-ICSS.

30.4.3.2 Industrial Ethernet Timer

The industrial ethernet timer is a simple 32-bit timer. This timer is intended for use by industrial ethernet functions but can also be leveraged as a generic timer in other applications.

30.4.3.2.1 Features

The industrial ethernet timer supports the following features:

- One master 32-bit count-up counter with an overflow status bit.
  - Runs on iep_clk or ocp_clk
  - Write 1 to clear status.

  - Supports a programmable increment value from 1 to 16 (default 5).
  - An optional compensation method allows the increment value to apply a compensation increment value from 1 to 16, counting up to 2^24 iep_clk/ocp_clk events.
- Sixteen 32-bit compare registers (CMP[15:0], CMP_STAT).
  - Sixteen status bits, write 1 to clear.
  - Sixteen individual event outputs.
  - One global event (any compare event) output for interrupt generation triggered by any compare event.
- Ten 32-bit capture registers (CAPR[5:0], CAPR[7:6], CAPF[7:6]).
  - Eight capture inputs with optional synchronous or asynchronous mode.
    - Six rise-only capture inputs (CAPR[5:0]).
    - Two rise-and-fall capture inputs:
      - CAPR[7] and CAPF[7]
      - CAPR[6] and CAPF[6]
    - One input signal will be used by two capture registers:
      - One register for rising edge.
      - One register for falling edge.
    - One global event (any capture event) output for interrupt generation triggered by any capture event.
- 32 outputs, one high level and one high pulse for each compare hit event.
- CMP[0], if enabled, will reset the counter on the next iep_clk/ocp_clk cycle.

30.4.3.2.2 Basic Programming Model

Follow these basic steps to configure the IEP Timer.

1. Initialize timer to known state (default values).
   a. Disable counter (GLB_CFG.CNT_ENABLE).
   b. Reset Count Register (CNT) by writing 0xFFFFFFFF to clear.
   c. Clear overflow status register (GLB_STS.CNT_OVF).
   d. Clear compare status (STS).
2. Set compare values (CMP0-CMPx).
3. Enable compare events (CMP_CFG.CMP_EN).
4. Set increment value (GLB_CFG.DEFAULT_INC).
5. Set compensation value (COMPEN.COMPEN_CNT).
6. Enable counter (GLB_CFG.CNT_ENABLE).

30.4.3.2.3 Industrial Ethernet Mapping

Some of the capture inputs and compare registers are mapped to specific industrial Ethernet functions in hardware, shown in Table 30-32. All capture inputs are mapped to industrial ethernet functions, and these inputs are not available for any other application. The cmp1 and cmp2 compare registers also function as the start time triggers for SYNC0 and SYNC1, respectively.

Table 30-32. Industrial Ethernet Timer Mode Mapping

Capture Input	IEP Line/Function
cap0, rise only	PRU0_RX_SOF
cap1, rise only	PRU0_RX_SFD
cap2, rise only	PRU1_RX_SOF
cap3, rise only	PRU1_RX_SFD
cap4, rise only	PORT0_TX_SOF
cap5, rise only	PORT1_TX_SOF
cap6, rise and fall	latch0_in, SOC IO inputs
cap7, rise and fall	latch1_in, SOC IO inputs
cmp1	For SYNC0 trigger of start time
cmp2	For SYNC1 trigger of start time only valid in the independent mode
cmp3	For MII TX0 start trigger, if MII register TXCFG0/1[TX_EN_MODE] is enabled.
cmp4	For MII TX1 start trigger, if MII register TXCFG0/1[TX_EN_MODE] is enabled.

30.4.3.3 Industrial Ethernet SYNC0/SYNC1

The industrial ethernet sync block supports the generation of two synchronization signals: SYNC0 and SYNC1. SYNC0 and SYNC1 can be directly mapped to output signals (pr1_edc_sycn0_out and pr1_edc_sync1_out) for external devices to use. They can also be used for internal synchronization within the PRU-ICSS. These signals are also mapped as system events and can therefore be mapped to the ARM core's Host interrupts.

30.4.3.3.1 Features

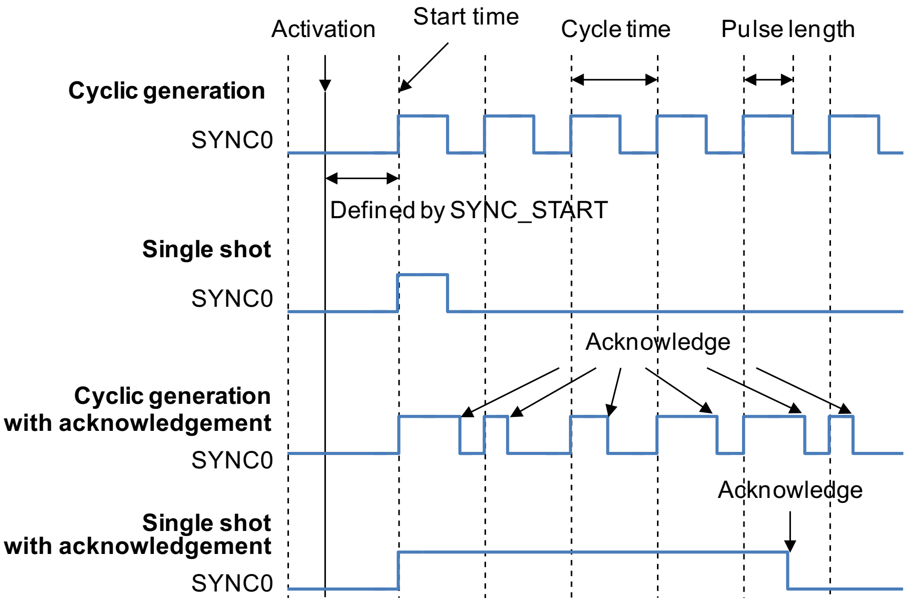
The industrial ethernet sync block supports the following features:

- Two synchronize generation signals (SYNC0, SYNC1)
  - Activation time synchronized with IEP Timer
    - CMP[1] triggers SYNC0 activation time
    - CMP[2] triggers SYNC1 activation time (only valid in the independent mode)
  - Pulse width defined by registers or ack mode (remain asserted until software acknowledged)
  - Cyclic or single-shot operation
  - Option to enable or disable sync generation
- Programmable number of clock cycles between the start of SYNC0 to the start of SYNC1

30.4.3.3.2 Sync Generation Modes

There are four modes of operation for the sync signals: cyclic mode, single shot mode, cyclic with acknowledge mode, and single shot with acknowledge mode. Figure 30-27 shows examples of these modes. The start time is set by the SYNC_START register. The cycle time is configured by the SYNC0_PERIOD register. The pulse length is defined by SYNC_PWIDTH register.

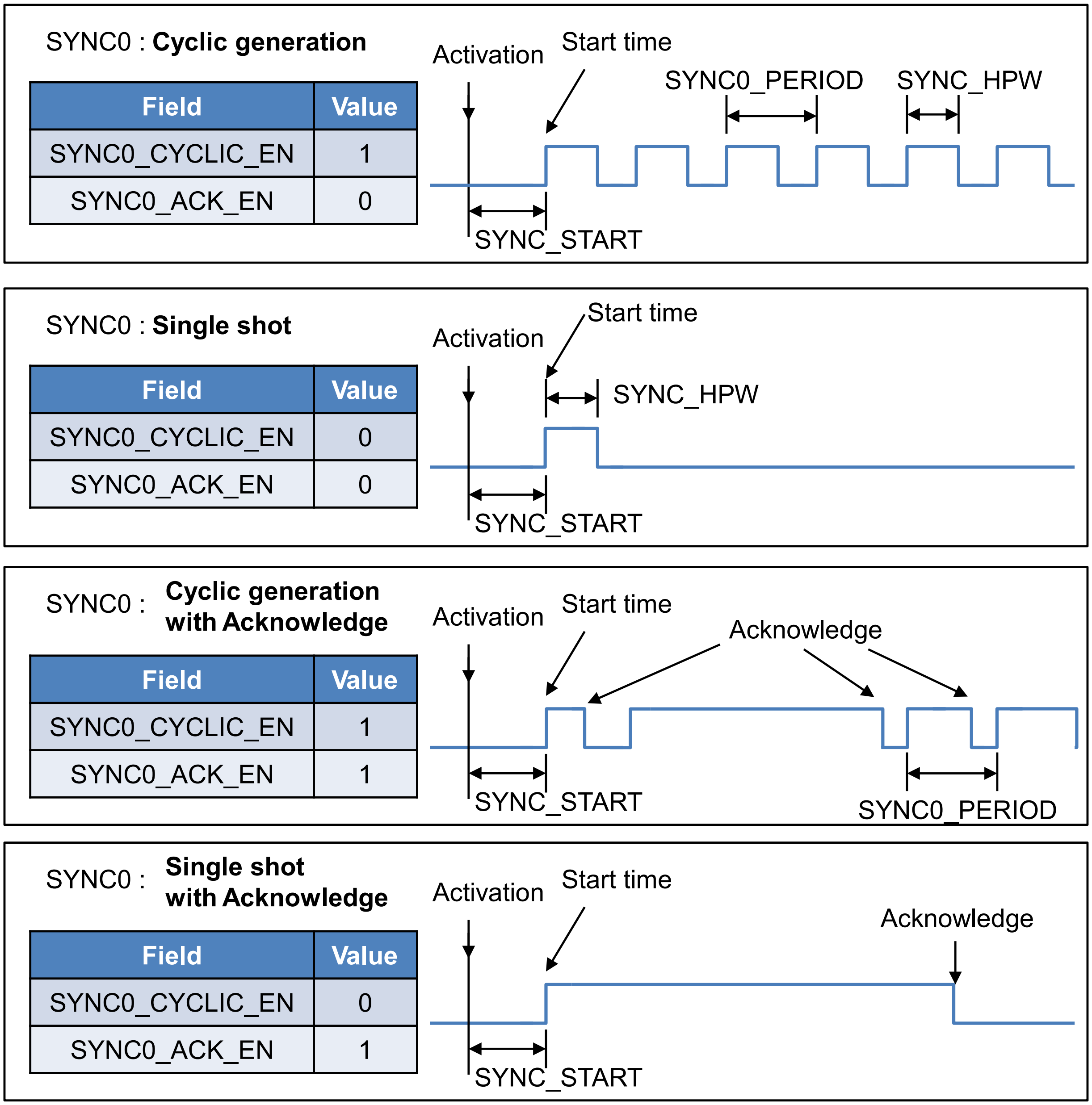
Figure 30-27. Sync Signal Generation Mode



In SYNC1 dependent mode (SYNC_CTRL.sync1_ind_en = 0), SYNC1 depends on SYNC0 and the start time of the SYNC1 can be defined by the SYNC1_DELAY register. Figure 30-28 shows different examples when changing the value in the SYNC1_DELAY register. Note if the SYNC1 delay time is 0, SYNC1 reflects SYNC0.

Cyclic generation cannot be used for network time synchronized applications because only the CMP1/CMP2 hit occurs in the compensated time domain.

Figure 30-28. Examples of the Dependent Mode of SYNC1



SYNC0 : **Cyclic generation**

Field	Value
SYNC0_CYCLIC_EN	1
SYNC0_ACK_EN	0

SYNC0 : **Single shot**

Field	Value
SYNC0_CYCLIC_EN	0
SYNC0_ACK_EN	0

SYNC0 : **Cyclic generation with Acknowledge**

Field	Value
SYNC0_CYCLIC_EN	1
SYNC0_ACK_EN	1

SYNC0 : **Single shot with Acknowledge**

Field	Value
SYNC0_CYCLIC_EN	0
SYNC0_ACK_EN	1

30.4.3.4 Industrial Ethernet Watchdog Timer

In industrial ethernet applications, the watchdog timer (WD) is used as a safety feature to monitor process data communication and to turn off the outputs of the digital input/output (DIGIO) functional block after a set time. The WD will thereby protect the system from errors or faults by timeout or expiration. The expiration is used to initiate corrective action in order to keep the system in a safe state and restore normal operation based on configuration. Therefore, if the system is stable, the watchdog timer should be regularly reset or cleared to avoid timeout or expiration.

30.4.3.4.1 Features

The industrial ethernet watchdog timer supports the following features:

- One 32-bit pre-divider for generating a WD clock (default 100μs) based on iep_clk input
- Two 16-bit Watchdog Timers:
  - PDI_WD for Sync Managers WD, used in conjunction with digital input/output (DIGIO)
  - PD_WD for data link layer user WD, used in conjunction with data link layer or application layer interface actions

30.4.3.5 Industrial Ethernet Digital I/O (DIGIO)

The IEP digital I/O (DIGIO) block provides dedicated I/Os intended for industrial ethernet protocols, but they can also be used as generic I/Os in other applications.

The IEP digital I/O (DIGIO) block provides dedicated I/Os for industrial ethernet protocols. The digital inputs can be sampled when specific events occur or continuously as a raw input. Likewise, driving the digital outputs can be triggered by specific events or controlled by software. The timing, delay cycle clocks, data sources, and data valid of the digital input and outputs are controlled by the DIGIO_CTRL and DIGIO_EXP registers.

Additionally, the IEP DIGIO block can be used as generic I/Os in other applications.

30.4.3.5.1 Features

The industrial ethernet digital I/O supports the following features:

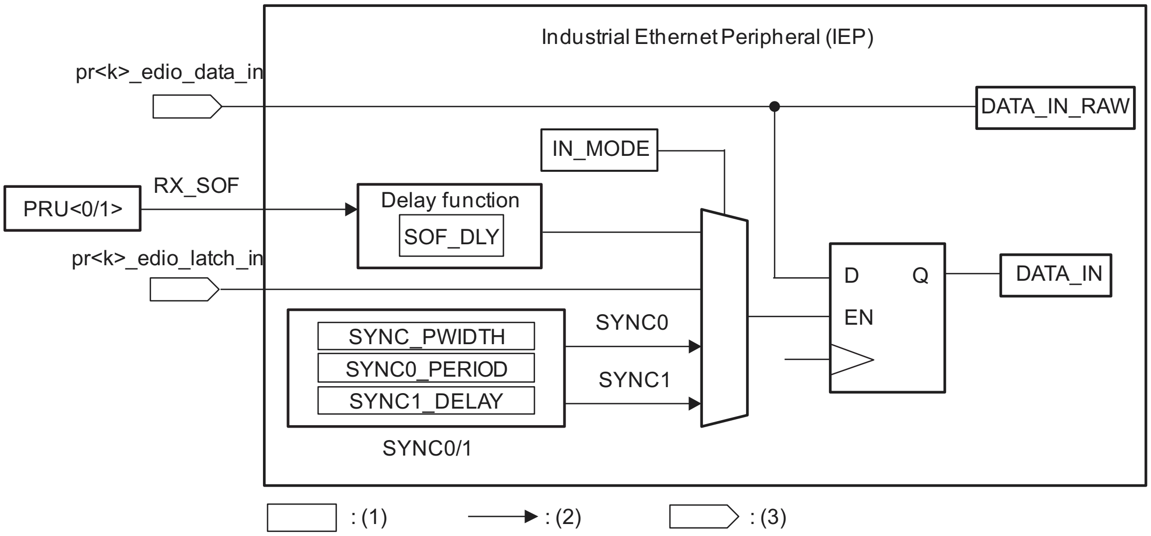
- Digital data output
  - 32 channels (pr<k>_edio_data_out[31:0])
  - Software controls enable signal driving output data output
  - Five event options for driving output data output
    - End of frame event (PRU0/1_RX_EOF)
    - SYNC0 events
    - SYNC1 events
    - Watchdog trigger
    - Software enable
- Digital data out enable (optional tri-state control)
- Digital data input
  - 32 channels (pr<k>_edio_data_in[31:0])
  - DIGIO_DATA_IN_RAW supports direct sampling of pr<k>_edio_data_in
  - External latch event signal (pr<k>_edio_latch_in) triggers a pulse on which pr<k>_edio_data_in is sampled
  - DIGIO_DATA_IN supports four event options to trigger sampling of pr<k>_edio_data_in
    - Start of frame event in start of frame (SOF) mode
    - pr<k>_edio_latch_in_event
    - SYNC0 events
    - SYNC1 events

NOTE: Some devices may not pin out all 32 data signals. See Table 30-3, *PRU-ICSS Pin List*, for the data pins that are available on this device.

30.4.3.5.2 DIGIO Block Diagrams

Figure 30-29 shows the signals and registers for capturing the DIGIO data in. Note that IN_MODE in the DIGIO_CTRL register must be set to 1 for data to be latched on the external pr<k>_edio_latch_in signal.Note in PRU0/1_RX_SOF mode, the delay time of capturing pr1_edio_data_in is programmable through the SOF_DLY bit of the DIGIO_EXT register.

Figure 30-29. IEP DIGIO Data In



(1) Register

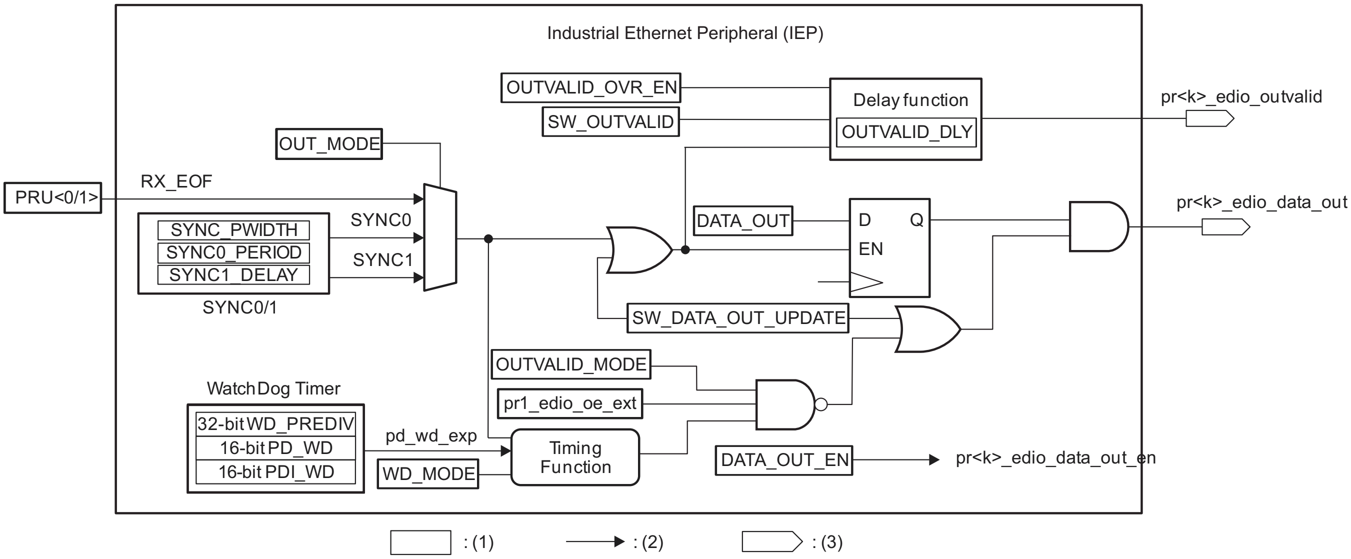
(2) Internal signal wire

(3) External pin input/output

Figure 30-30 shows the signals and registers for driving the DIGIO data out.

Figure 30-30 shows the signals and registers for driving the industrial ethernet data out. The pr1_edio_data_out is immediately forced to zero when OUTVALID_MODE = 1, pr1_edio_oe_ext = 1, and PD_WD_EXP = 1, or the next update hardware post PD_WD_EXP. Delay assertion of pr1_edio_outvalid from pr1_edio_data_out update events are controlled by software (SW_OUTVALID).

Figure 30-30. IEP DIGIO Data Out



(1) Register

(2) Internal signal wire

(3) External pin input/output

30.4.3.5.3 Basic Programming Model

Follow these steps to configure and read the DIGIO Data Input.

1. Read DIGIO_DATA_IN_RAW for raw input data.

   or

1. Enable sampling of pr<k>_edio_data_in[31:0] by setting DIGIO_CTRL.IN_MODE.
2. Read DIGIO_DATA_IN for data sampled by the selected trigger source.

Follow these steps to configure and write to the DIGIO Data Output:

1. Pre-configure DIGIO by setting DIGIO_EXP.OUTVALID_OVR_EN and DIGIO_EXP.SW_DATA_OUT_UPDATE.
2. Write to DIGIO_DATA_OUT to configure output data.
3. To Hi-Z output, set DIGIO_DATA_OUT_EN. (Clear DIGIO_DATA_OUT_EN to drive value stored in DIGIO_DATA_OUT.)

30.4.4 Universal Asynchronous Receiver/Transmitter

30.4.4.1 Introduction

30.4.4.1.1 Purpose of the Peripheral

The PRU UART peripheral within the PRU-ICSS is based on the industry standard TL16C550 asynchronous communications element, which in turn is a functional upgrade of the TL16C450. Functionally similar to the TL16C450 on power up (single character or TL16C450 mode), the UART can be placed in an alternate FIFO (TL16C550) mode. This relieves the CPU of excessive software overhead by buffering received and transmitted characters. The receiver and transmitter FIFOs store up to 16 bytes including three additional bits of error status per byte for the receiver FIFO.

The UART performs serial-to-parallel conversions on data received from a peripheral device and parallel-to-serial conversion on data received from the CPU. The CPU can read the UART status at any time. The UART includes control capability and a processor interrupt system that can be tailored to minimize software management of the communications link.

The UART includes a programmable baud rate generator capable of dividing the PRU_ICSS_UART_CLK input clock by divisors from 1 to 65535 and producing a 16× reference clock or a 13× reference clock for the internal transmitter and receiver logic. For detailed timing and electrical specifications for the UART, see your device-specific data manual.

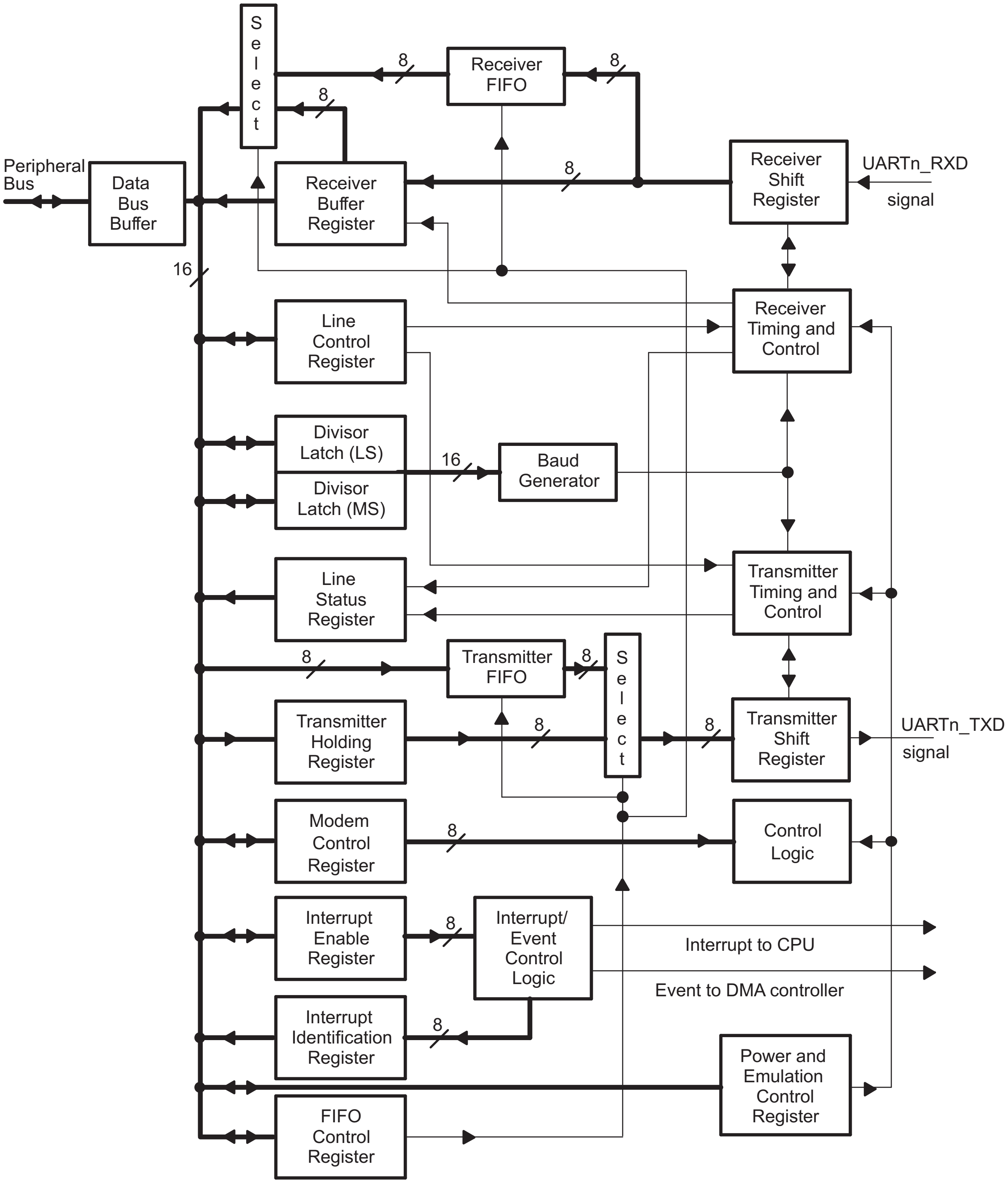
30.4.4.1.2 Functional Block Diagram

A functional block diagram of the UART is shown in Figure 30-31.

30.4.4.1.3 Industry Standard(s) Compliance Statement

The UART peripheral is based on the industry standard TL16C550 asynchronous communications element, which is a functional upgrade of the TL16C450. The information in this chapter assumes you are familiar with these standards.

Figure 30-31. UART Block Diagram



NOTE: The value *n* indicates the applicable UART where there are multiple instances. For the PRU-ICSS, there is only one instance and all UART signals should reflect this (e.g., UART0_TXD instead of UARTn_TXD).

30.4.4.2 Functional Description

30.4.4.2.1 Clock Generation and Control

The UART bit clock is derived from an input clock to the UART. See your device-specific data manual to check the maximum data rate supported by the UART.

Figure 30-32 is a conceptual clock generation diagram for the UART. The PRU_ICSS_UART_CLK is input to the Clock Generator, which uses a programmable divider to produce the UART input clock. The UART contains a programmable baud generator that takes the UART input clock and divides it by a divisor in the range between 1 and (2^{16} - 1) to produce a baud clock (BCLK). The frequency of BCLK is sixteen times (16×) the baud rate (each received or transmitted bit lasts 16 BCLK cycles) or thirteen times (13×) the baud rate (each received or transmitted bit lasts 13 BCLK cycles). When the UART is receiving, the bit is sampled in the 8th BCLK cycle for 16× over sampling mode and on the 6th BCLK cycle for 13× over-sampling mode. The 16× or 13× reference clock is selected by configuring the OSM_SEL bit in the mode definition register (MDR). The formula to calculate the divisor is:

$$\textbf{Divisor} = \frac{\textbf{UART input clock frequency}}{\textbf{Desired baud rate} \times \textbf{16}} \qquad [\textbf{MDR.OSM\_SEL} = \textbf{0}]$$

$$\textbf{Divisor} = \frac{\textbf{UART input clock frequency}}{\textbf{Desired baud rate} \times \textbf{13}} \qquad [\textbf{MDR.OSM\_SEL} = \textbf{1}]$$

Two 8-bit register fields (DLH and DLL), called divisor latches, hold this 16-bit divisor. DLH holds the most significant bits of the divisor, and DLL holds the least significant bits of the divisor. For information about these register fields, see the UART register descriptions. These divisor latches must be loaded during initialization of the UART in order to ensure desired operation of the baud generator. Writing to the divisor latches results in two wait states being inserted during the write access while the baud generator is loaded with the new value.

Figure 30-33 summarizes the relationship between the transferred data bit, BCLK, and the UART input clock. Note that the timing relationship depicted in Figure 30-33 shows that each bit lasts for 16 BCLK cycles . This is in case of 16x over-sampling mode. For 13× over-sampling mode each bit lasts for 13 BCLK cycles .

Example baud rates and divisor values relative to a 192-MHz UART input clock and 16× over-sampling mode are shown in Table 30-33.

Figure 30-32. UART Clock Generation Diagram

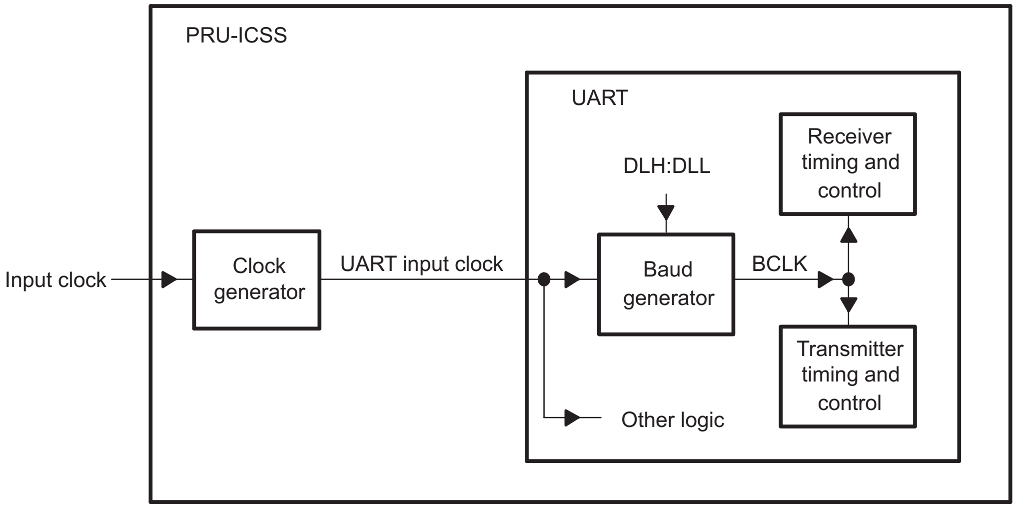


Figure 30-33. Relationships Between Data Bit, BCLK, and UART Input Clock

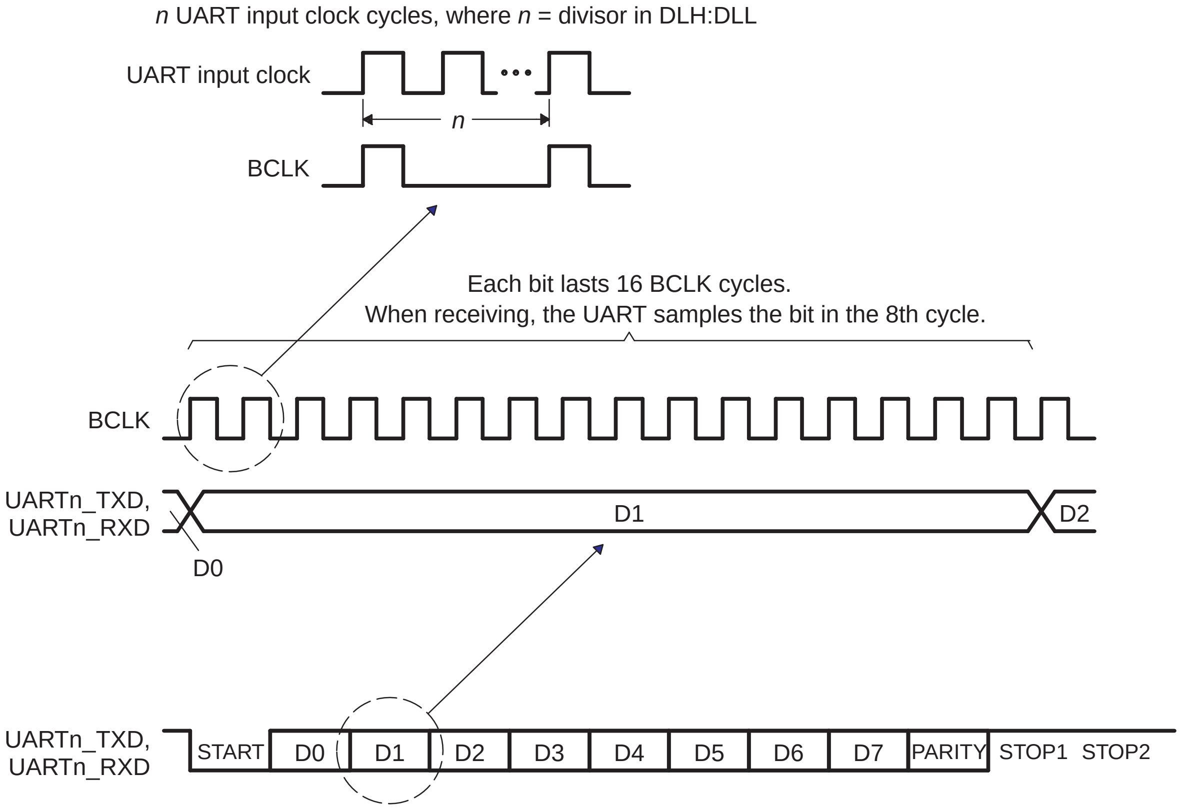


Table 30-33. Baud Rate Examples for 192-MHZ UART Input Clock and 16× Over-sampling Mode

Baud Rate	Divisor Value	Actual Baud Rate	Error (%)
2400	5000	2400	0.00
4800	2500	4800	0.00
9600	1250	9600	0.00
19200	625	19200	0.00
38400	313	38338.658	–0.16
56000	214	56074.766	0.13
128000	94	127659.574	–0.27
300000	40	300000	0.00

Table 30-34. Baud Rate Examples for 192-MHZ UART Input Clock and 13× Over-sampling Mode

Baud Rate	Divisor Value	Actual Baud Rate	Error (%)
2400	6154	2399.940	–0.0025
4800	3077	4799.880	–0.0025
9600	1538	9602.881	0.03
19200	769	19205.762	0.03
38400	385	38361.638	–0.10
56000	264	55944.056	–0.10
128000	115	128428.094	0.33
300000	49	301412.873	0.47

30.4.4.2.2 Signal Descriptions

The UARTs utilize a minimal number of signal connections to interface with external devices. The UART signal descriptions are included in Table 30-35. Note that the number of UARTs and their supported features vary on each device. See your device-specific data manual for more details.

Table 30-35. UART Signal Descriptions

Signal Name[1]	Signal Type	Function
UARTn_TXD	Output	Serial data transmit
UARTn_RXD	Input	Serial data receive
$\overline{\text{UARTn\_CTS}}$ [2]	Input	Clear-to-Send handshaking signal
$\overline{\text{UARTn\_RTS}}$ [2]	Output	Request-to-Send handshaking signal

[1] The value *n* indicates the applicable UART; that is, UART0, UART1, etc.

[2] This signal is not supported in all UARTs. See your device-specific data manual to check if it is supported.

30.4.4.2.3 Pin Multiplexing

Extensive pin multiplexing is used to accommodate the largest number of peripheral functions in the smallest possible package. Pin multiplexing is controlled using a combination of hardware configuration at device reset and software programmable register settings. See your device-specific data manual to determine how pin multiplexing affects the UART.

30.4.4.2.4 Protocol Description

30.4.4.2.4.1 Transmission

The UART transmitter section includes a transmitter hold register (THR) and a transmitter shift register (TSR). When the UART is in the FIFO mode, THR is a 16-byte FIFO. Transmitter section control is a function of the UART line control register (LCR). Based on the settings chosen in LCR, the UART transmitter sends the following to the receiving device:

- 1 START bit
- 5, 6, 7, or 8 data bits
- 1 PARITY bit (optional)
- 1, 1.5, or 2 STOP bits

30.4.4.2.4.2 Reception

The UART receiver section includes a receiver shift register (RSR) and a receiver buffer register (RBR). When the UART is in the FIFO mode, RBR is a 16-byte FIFO. Receiver section control is a function of the UART line control register (LCR). Based on the settings chosen in LCR, the UART receiver accepts the following from the transmitting device:

- 1 START bit
- 5, 6, 7, or 8 data bits
- 1 PARITY bit (optional)
- 1 STOP bit (any other STOP bits transferred with the above data are not detected)

30.4.4.2.4.3 Data Format

The UART transmits in the following format:

1 START bit + data bits (5, 6, 7, 8) + 1 PARITY bit (optional) + STOP bit (1, 1.5, 2)

It transmits 1 START bit; 5, 6, 7, or 8 data bits, depending on the data width selection; 1 PARITY bit, if parity is selected; and 1, 1.5, or 2 STOP bits, depending on the STOP bit selection.

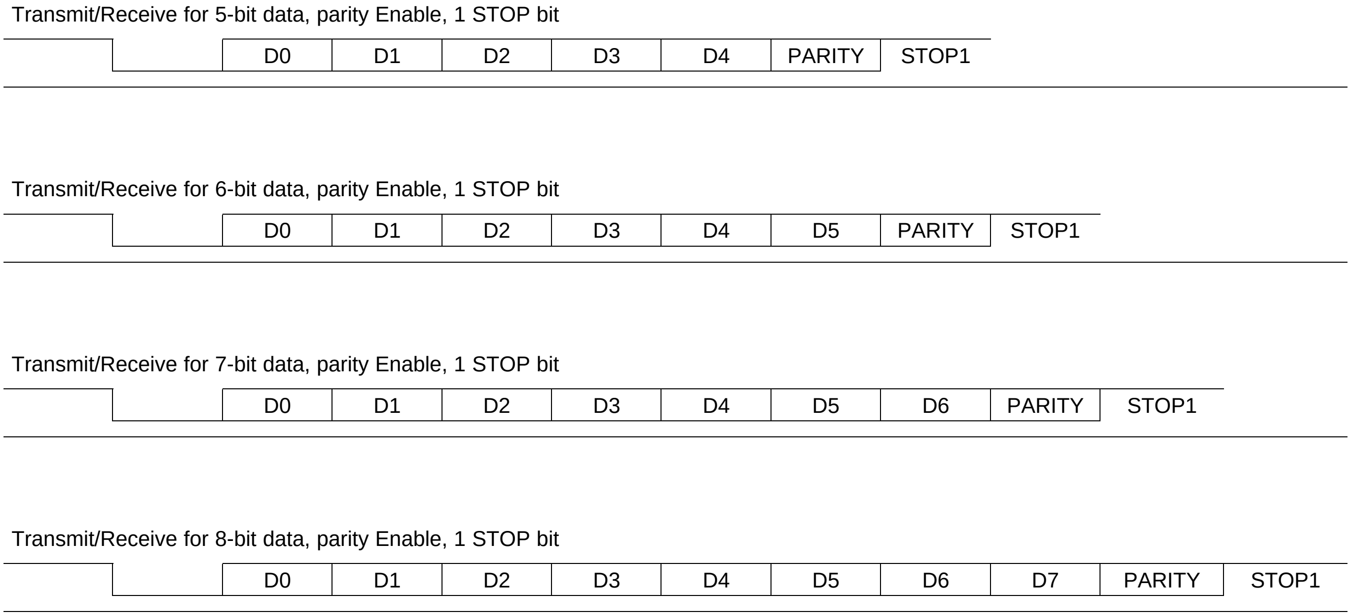
The UART receives in the following format:

1 START bit + data bits (5, 6, 7, 8) + 1 PARITY bit (optional) + 1 STOP bit

It receives 1 START bit; 5, 6, 7, or 8 data bits, depending on the data width selection; 1 PARITY bit, if parity is selected; and 1 STOP bit.

The protocol formats are shown in Figure 30-34.

Figure 30-34. UART Protocol Formats



30.4.4.2.5 Operation

30.4.4.2.5.1 Transmission

The UART transmitter section includes a transmitter hold register (THR) and a transmitter shift register (TSR). When the UART is in the FIFO mode, THR is a 16-byte FIFO. Transmitter section control is a function of the UART line control register (LCR). Based on the settings chosen in LCR, the UART transmitter sends the following to the receiving device:

- 1 START bit
- 5, 6, 7, or 8 data bits
- 1 PARITY bit (optional)
- 1, 1.5, or 2 STOP bits

THR receives data from the internal data bus, and when TSR is ready, the UART moves the data from THR to TSR. The UART serializes the data in TSR and transmits the data on the UARTn_TXD pin.

In the non-FIFO mode, if THR is empty and the THR empty (THRE) interrupt is enabled in the interrupt enable register (IER), an interrupt is generated. This interrupt is cleared when a character is loaded into THR or the interrupt identification register (IIR) is read. In the FIFO mode, the interrupt is generated when the transmitter FIFO is empty, and it is cleared when at least one byte is loaded into the FIFO or IIR is read.

30.4.4.2.5.2 Reception

The UART receiver section includes a receiver shift register (RSR) and a receiver buffer register (RBR). When the UART is in the FIFO mode, RBR is a 16-byte FIFO. Timing is supplied by the 16× receiver clock. Receiver section control is a function of the UART line control register (LCR). Based on the settings chosen in LCR, the UART receiver accepts the following from the transmitting device:

- 1 START bit
- 5, 6, 7, or 8 data bits
- 1 PARITY bit (optional)
- 1 STOP bit (any other STOP bits transferred with the above data are not detected)

RSR receives the data bits from the UARTn_RXD pin. Then RSR concatenates the data bits and moves the resulting value into RBR (or the receiver FIFO). The UART also stores three bits of error status information next to each received character, to record a parity error, framing error, or break.

In the non-FIFO mode, when a character is placed in RBR and the receiver data-ready interrupt is enabled in the interrupt enable register (IER), an interrupt is generated. This interrupt is cleared when the character is read from RBR. In the FIFO mode, the interrupt is generated when the FIFO is filled to the trigger level selected in the FIFO control register (FCR), and it is cleared when the FIFO contents drop below the trigger level.

30.4.4.2.5.3 FIFO Modes

The following two modes can be used for servicing the receiver and transmitter FIFOs:

- FIFO interrupt mode. The FIFO is enabled and the associated interrupts are enabled. Interrupts are sent to the CPU to indicate when specific events occur.
- FIFO poll mode. The FIFO is enabled but the associated interrupts are disabled. The CPU polls status bits to detect specific events.

Because the receiver FIFO and the transmitter FIFO are controlled separately, either one or both can be placed into the interrupt mode or the poll mode.

30.4.4.2.5.3.1 FIFO Interrupt Mode

When the receiver FIFO is enabled in the FIFO control register (FCR) and the receiver interrupts are enabled in the interrupt enable register (IER), the interrupt mode is selected for the receiver FIFO. The following are important points about the receiver interrupts:

- The receiver data-ready interrupt is issued to the CPU when the FIFO has reached the trigger level that is programmed in FCR. It is cleared when the CPU or the DMA controller reads enough characters from the FIFO such that the FIFO drops below its programmed trigger level.
- The receiver line status interrupt is generated in response to an overrun error, a parity error, a framing error, or a break. This interrupt has higher priority than the receiver data-ready interrupt. For details, see Section 30.4.4.2.8.
- The data-ready (DR) bit in the line status register (LSR) indicates the presence or absence of characters in the receiver FIFO. The DR bit is set when a character is transferred from the receiver shift register (RSR) to the empty receiver FIFO. The DR bit remains set until the FIFO is empty again.
- A receiver time-out interrupt occurs if all of the following conditions exist:
  - At least one character is in the FIFO,
  - The most recent character was received more than four continuous character times ago. A character time is the time allotted for 1 START bit, *n* data bits, 1 PARITY bit, and 1 STOP bit, where *n* depends on the word length selected with the WLS bits in the line control register (LCR). See Table 30-36.
  - The most recent read of the FIFO has occurred more than four continuous character times before.
- Character times are calculated by using the baud rate.
- When a receiver time-out interrupt has occurred, it is cleared and the time-out timer is cleared when the CPU or the EDMA controller reads one character from the receiver FIFO. The interrupt is also cleared if a new character is received in the FIFO or if the URRST bit is cleared in the power and emulation management register (PWREMU_MGMT).
- If a receiver time-out interrupt has not occurred, the time-out timer is cleared after a new character is received or after the CPU or EDMA reads the receiver FIFO.

When the transmitter FIFO is enabled in FCR and the transmitter holding register empty (THRE) interrupt is enabled in IER, the interrupt mode is selected for the transmitter FIFO. The THRE interrupt occurs when the transmitter FIFO is empty. It is cleared when the transmitter hold register (THR) is loaded (1 to 16 characters may be written to the transmitter FIFO while servicing this interrupt) or the interrupt identification register (IIR) is read.

Table 30-36. Character Time for Word Lengths

Word Length (*n*)	Character Time	Four Character Times
5	Time for 8 bits	Time for 32 bits
6	Time for 9 bits	Time for 36 bits
7	Time for 10 bits	Time for 40 bits
8	Time for 11 bits	Time for 44 bits

30.4.4.2.5.3.2 FIFO Poll Mode

When the receiver FIFO is enabled in the FIFO control register (FCR) and the receiver interrupts are disabled in the interrupt enable register (IER), the poll mode is selected for the receiver FIFO. Similarly, when the transmitter FIFO is enabled and the transmitter interrupts are disabled, the transmitter FIFO is in the poll mode. In the poll mode, the CPU detects events by checking bits in the line status register (LSR):

- The RXFIFOE bit indicates whether there are any errors in the receiver FIFO.
- The TEMT bit indicates that both the transmitter holding register (THR) and the transmitter shift register (TSR) are empty.
- The THRE bit indicates when THR is empty.
- The BI (break), FE (framing error), PE (parity error), and OE (overrun error) bits specify which error or errors have occurred.
- The DR (data-ready) bit is set as long as there is at least one byte in the receiver FIFO.

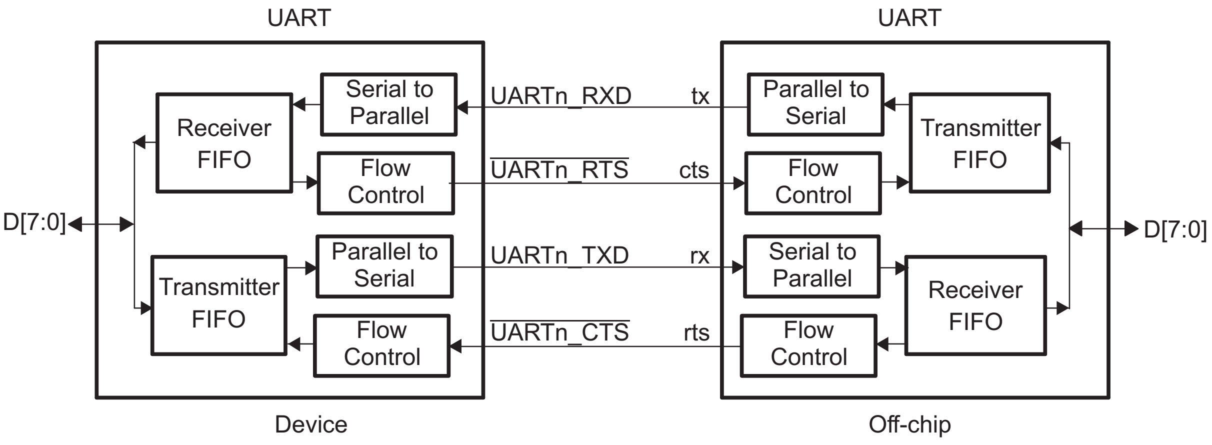
Also, in the FIFO poll mode:

- The interrupt identification register (IIR) is not affected by any events because the interrupts are disabled.
- The UART does not indicate when the receiver FIFO trigger level is reached or when a receiver time-out occurs.

30.4.4.2.5.4 Autoflow Control

The UART can employ autoflow control by connecting the UARTn_CTS and UARTn_RTS signals. Note that all UARTs do not support autoflow control; see your device-specific data manual for supported features. The UARTn_CTS input must be active before the transmitter FIFO can transmit data. The UARTn_RTS output becomes active when the receiver needs more data and notifies the sending device. When UARTn_RTS is connected to UARTn_CTS, data transmission does not occur unless the receiver FIFO has space for the data. Therefore, when two UARTs are connected as shown in Figure 30-35 with autoflow enabled, overrun errors are eliminated.

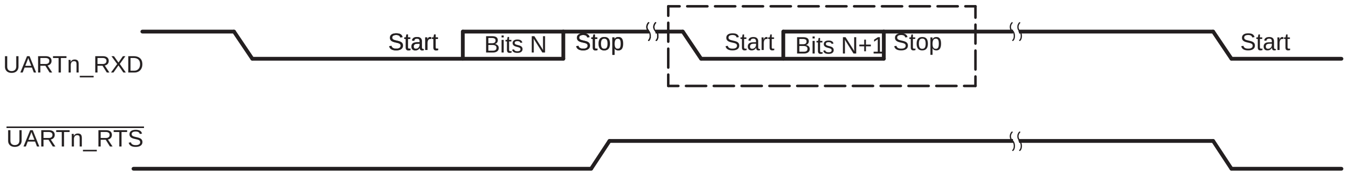
Figure 30-35. UART Interface Using Autoflow Diagram



30.4.4.2.5.4.1 UARTn_RTS Behavior

UARTn_RTS data flow control originates in the receiver block (see Figure 30-31). When the receiver FIFO level reaches a trigger level of 1, 4, 8, or 14 (see Figure 30-36), UARTn_RTS is deasserted. The sending UART may send an additional byte after the trigger level is reached (assuming the sending UART has another byte to send), because it may not recognize the deassertion of UARTn_RTS until after it has begun sending the additional byte. For trigger level 1, 4, and 8, UARTn_RTS is automatically reasserted once the receiver FIFO is emptied. For trigger level 14, UARTn_RTS is automatically reasserted once the receiver FIFO drops below the trigger level.

Figure 30-36. Autoflow Functional Timing Waveforms for UARTn_RTS



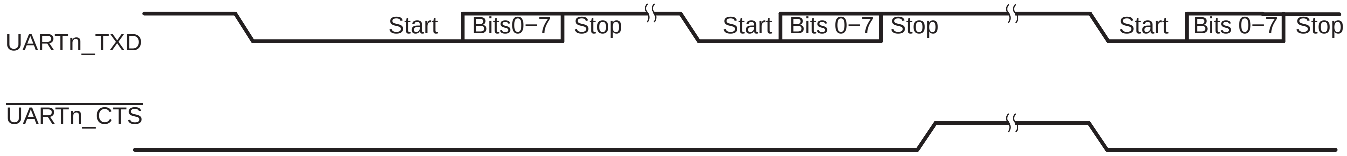
(1) N = Receiver FIFO trigger level.

(2) The block in dashed lines covers the case where an additional byte is sent.

30.4.4.2.5.4.2 UARTn_CTS Behavior

The transmitter checks UARTn_CTS before sending the next data byte. If UARTn_CTS is active, the transmitter sends the next byte. To stop the transmitter from sending the following byte, UARTn_CTS must be released before the middle of the last STOP bit that is currently being sent (see Figure 30-37). When flow control is enabled, UARTn_CTS level changes do not trigger interrupts because the device automatically controls its own transmitter. Without autoflow control, the transmitter sends any data present in the transmitter FIFO and a receiver overrun error may result.

Figure 30-37. Autoflow Functional Timing Waveforms for UARTn_CTS



(1) When UARTn_CTS is active (low), the transmitter keeps sending serial data out.

(2) When UARTn_CTS goes high before the middle of the last STOP bit of the current byte, the transmitter finishes sending the current byte but it does not send the next byte.

(3) When UARTn_CTS goes from high to low, the transmitter begins sending data again.

30.4.4.2.5.5 Loopback Control

The UART can be placed in the diagnostic mode using the LOOP bit in the modem control register (MCR), which internally connects the UART output back to the UART input. In this mode, the transmit and receive data paths, the transmitter and receiver interrupts, and the modem control interrupts can be verified without connecting to another UART.

30.4.4.2.6 Reset Considerations

30.4.4.2.6.1 Software Reset Considerations

Two bits in the power and emulation management register (PWREMU_MGMT) control resetting the parts of the UART:

- The UTRST bit controls resetting the transmitter only. If UTRST = 1, the transmitter is active; if UTRST = 0, the transmitter is in reset.
- The URRST bit controls resetting the receiver only. If URRST = 1, the receiver is active; if URRST = 0, the receiver is in reset.

In each case, putting the receiver and/or transmitter in reset will reset the state machine of the affected portion but does not affect the UART registers.

30.4.4.2.6.2 Hardware Reset Considerations

When the processor RESET pin is asserted, the entire processor is reset and is held in the reset state until the RESET pin is released. As part of a device reset, the UART state machine is reset and the UART registers are forced to their default states. The default states of the registers are shown in , *PRU_ICSS_UART Registers*.

30.4.4.2.7 Initialization

The following steps are required to initialize the UART:

1. Perform the necessary device pin multiplexing setup (see your device-specific data manual).
2. Set the desired baud rate by writing the appropriate clock divisor values to the divisor latch registers (DLL and DLH).
3. If the FIFOs will be used, select the desired trigger level and enable the FIFOs by writing the appropriate values to the FIFO control register (FCR). The FIFOEN bit in FCR must be set first, before the other bits in FCR are configured.
4. Choose the desired protocol settings by writing the appropriate values to the line control register (LCR).
5. If autoflow control is desired, write appropriate values to the modem control register (MCR). Note that all UARTs do not support autoflow control; see your device-specific data manual for supported features.
6. Choose the desired response to emulation suspend events by configuring the FREE bit, and enable the UART by setting the UTRST and URRST bits in the power and emulation management register (PWREMU_MGMT).

30.4.4.2.8 Interrupt Support

30.4.4.2.8.1 Interrupt Events and Requests

The UART generates the interrupt requests described in Table 30-37. All requests are multiplexed through an arbiter to a single UART interrupt request to the CPU, as shown in Figure 30-38. Each of the interrupt requests has an enable bit in the interrupt enable register (IER) and is recorded in the interrupt identification register (IIR).

If an interrupt occurs and the corresponding enable bit is set to 1, the interrupt request is recorded in IIR and is forwarded to the CPU. If an interrupt occurs and the corresponding enable bit is cleared to 0, the interrupt request is blocked. The interrupt request is neither recorded in IIR nor forwarded to the CPU.

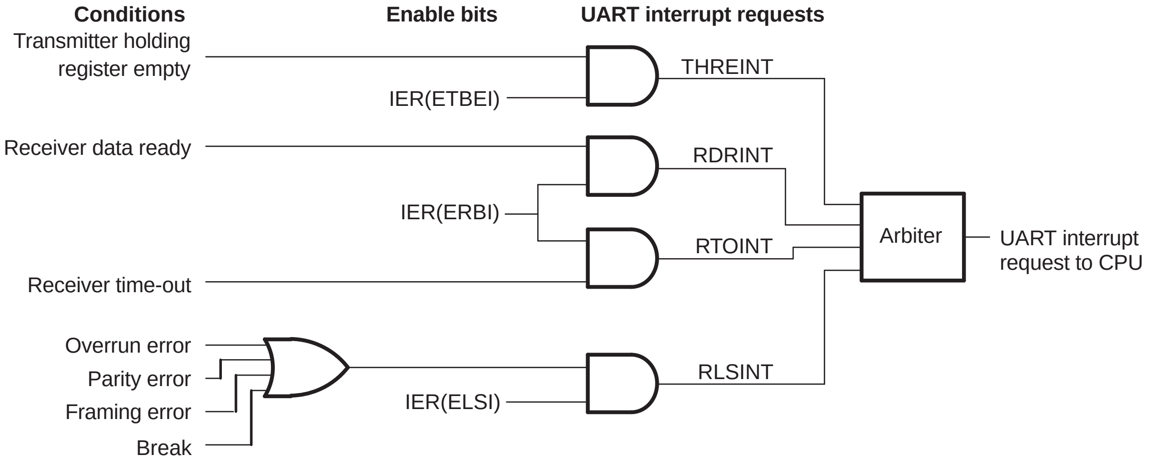
30.4.4.2.8.2 Interrupt Multiplexing

The UARTs have dedicated interrupt signals to the CPU and the interrupts are not multiplexed with any other interrupt source.

Table 30-37. UART Interrupt Requests Descriptions

UART Interrupt Request	Interrupt Source	Comment
THREINT	THR-empty condition: The transmitter holding register (THR) or the transmitter FIFO is empty. All of the data has been copied from THR to the transmitter shift register (TSR).	If THREINT is enabled in IER, by setting the ETBEI bit, it is recorded in IIR. As an alternative to using THREINT, the CPU can poll the THRE bit in the line status register (LSR).
RDAINT	Receive data available in non-FIFO mode or trigger level reached in the FIFO mode.	If RDAINT is enabled in IER, by setting the ERBI bit, it is recorded in IIR. As an alternative to using RDAINT, the CPU can poll the DR bit in the line status register (LSR). In the FIFO mode, this is not a functionally equivalent alternative because the DR bit does not respond to the FIFO trigger level. The DR bit only indicates the presence or absence of unread characters.
RTOINT	Receiver time-out condition (in the FIFO mode only): No characters have been removed from or input to the receiver FIFO during the last four character times (see Table 30-36), and there is at least one character in the receiver FIFO during this time.	The receiver time-out interrupt prevents the UART from waiting indefinitely, in the case when the receiver FIFO level is below the trigger level and thus does not generate a receiver data-ready interrupt. If RTOINT is enabled in IER, by setting the ERBI bit, it is recorded in IIR. There is no status bit to reflect the occurrence of a time-out condition.
RLSINT	Receiver line status condition: An overrun error, parity error, framing error, or break has occurred.	If RLSINT is enabled in IER, by setting the ELSI bit, it is recorded in IIR. As an alternative to using RLSINT, the CPU can poll the following bits in the line status register (LSR): overrun error indicator (OE), parity error indicator (PE), framing error indicator (FE), and break indicator (BI).

Figure 30-38. UART Interrupt Request Enable Paths



30.4.4.2.9 DMA Event Support

In the FIFO mode, the UART generates the following two DMA events:

- **Receive event (URXEVT):** The trigger level for the receiver FIFO (1, 4, 8, or 14 characters) is set with the RXFIFTL bit in the FIFO control register (FCR). Every time the trigger level is reached or a receiver time-out occurs, the UART sends a receive event to the EDMA controller. In response, the EDMA controller reads the data from the receiver FIFO by way of the receiver buffer register (RBR). Note that the receive event is not asserted if the data at the top of the receiver FIFO is erroneous even if the trigger level has been reached.
- **Transmit event (UTXEVT):** When the transmitter FIFO is empty (when the last byte in the transmitter FIFO has been copied to the transmitter shift register), the UART sends an UTXEVT signal to the EDMA controller. In response, the EDMA controller refills the transmitter FIFO by way of the transmitter holding register (THR). The UTXEVT signal is also sent to the DMA controller when the UART is taken out of reset using the UTRST bit in the power and emulation management register (PWREMU_MGMT).

Activity in DMA channels can be synchronized to these events. In the non-FIFO mode, the UART generates no DMA events. Any DMA channel synchronized to either of these events must be enabled at the time the UART event is generated. Otherwise, the DMA channel will miss the event and, unless the UART generates a new event, no data transfer will occur.

30.4.4.2.10 Power Management

The UART peripheral can be placed in reduced-power modes to conserve power during periods of low activity. The power management of the UART peripheral is controlled by the processor Power and Clock Management (PRCM). The PRCM acts as a master controller for power management for all of the peripherals on the device. For detailed information on power management procedures using the PSC, see the chapter *Power, Reset, and Clock Management (PRCM)* in the device reference manual.

30.4.4.2.11 Emulation Considerations

The FREE bit in the power and emulation management register (PWREMU_MGMT) determines how the UART responds to an emulation suspend event such as an emulator halt or breakpoint. If FREE = 0 and a transmission is in progress, the UART halts after completing the one-word transmission; if FREE = 0 and a transmission is not in progress, the UART halts immediately. If FREE = 1, the UART does not halt and continues operating normally.

Note also that most emulator accesses are transparent to UART operation. Emulator read operations do not affect any register contents, status bits, or operating states, with the exception of the interrupt identification register (IIR). Emulator writes, however, may affect register contents and may affect UART operation, depending on which register is accessed and what value is written.

The UART registers can be read from or written to during emulation suspend events, even if the UART activity has stopped.

30.4.4.2.12 Exception Processing

30.4.4.2.12.1 Divisor Latch Not Programmed

Since the processor reset signal has no effect on the divisor latch, the divisor latch will have an unknown value after power up. If the divisor latch is not programmed after power up, the baud clock (BCLK) will not operate and will instead be set to a constant logic 1 state.

The divisor latch values should always be reinitialized following a processor reset.

30.4.4.2.12.2 Changing Operating Mode During Busy Serial Communication

Since the serial link characteristics are based on how the control registers are programmed, the UART will expect the control registers to be static while it is busy engaging in a serial communication. Therefore, changing the control registers while the module is still busy communicating with another serial device will most likely cause an error condition and should be avoided.

30.4.5 ECAP

The PRU ECAP module within the PRU-ICSS is identical to the ECAP module in the AM437x PWMSS. For additional details about the ECAP module, see Section 20.3, *Enhanced Capture (eCAP) Module*.

30.4.6 MII_RT

30.4.6.1 Introduction

The Real-time Media Independent Interface (MII_RT) provides a programmable I/O interface for the PRUs to access and control up to two MII ports. The MII_RT module can also be configured to push and pull data independent of the PRU cores.

NOTE: In order to guarantee the MII_RT IO timing values published in the device data manual, the ocp_clk must be configured for 200MHz (default value) and TXCFG<n>_TX_CLK_DELAY must be set to 6h (non-default value).

30.4.6.1.1 Features

The PRU-ICSS MII_RT module supports:

- Two MII ports
  - Each MII port has:
    - 32-byte RX L1 FIFO
    - 64-byte RX L2 buffer
    - 64-byte TX L1 FIFO
  - Rate decoupling on TX L1 FIFO
  - Configurable pre-amble removal on RX L1 FIFO and insertion on TX L1 FIFO
  - Configurable TX L1 FIFO trigger (10 bits with 40 ns ticks)
- MII port multiplexer per direction to support line/ring structure
  - Link detection through RX_ERR
- Cyclic redundancy check (CRC)
  - CRC32 generation on TX path
  - CRC32 checker on RX path

30.4.6.1.2 Unsupported Features

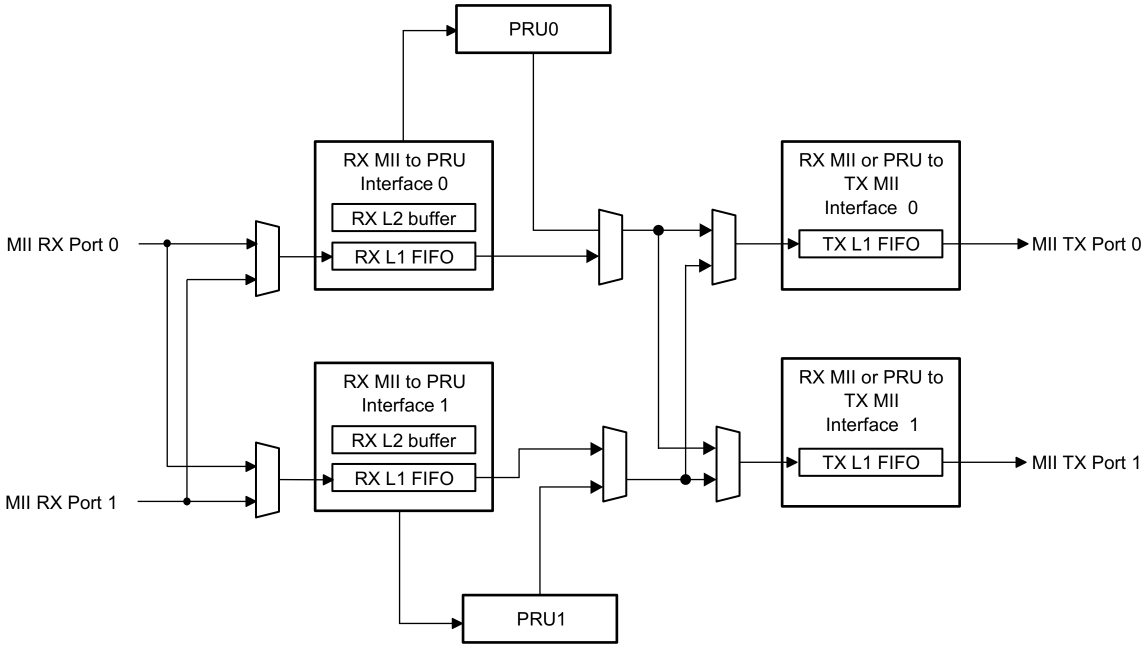
The PRU-ICSS MII_RT module does not support:

- Auto padding in TX L1 FIFO
- Dynamic TX multiplexer switching during packet handling
  - Can allow one PRU to handle both MII interfaces and a second PRU to manage the host and switch functions.

30.4.6.1.3 Block Diagram

Figure 30-39 shows the MII_RT in context of the PRU-ICSS. This diagram is a conceptual block diagram and does not necessarily reflect actual topologies.

Figure 30-39. MII_RT Block Diagram



30.4.6.2 Functional Description

30.4.6.2.1 Data Path Configuration

The MII_RT module supports three basic data path configurations. These configurations are compared in Table 30-38 and described in the following sections.

Table 30-38. Data Path Configuration Comparison

Configuration	PRU Dependency	Data Servicing	Port-to-Port Latency
Auto-forward	Snoop only	One word in flight	Low
8- or 16-bit processing with on-the-fly modifications (RX L1)	Yes	One word or byte in flight	Low
32-byte double buffer or ping-pong processing (RX L2)	Yes	Multi-words in flight	Medium (application-dependent)

30.4.6.2.1.1 Auto-forward with Optional PRU Snoop

Data is automatically forwarded from the MII RX port to the MII TX port without manipulations, as shown in Figure 30-40. This configuration does not depend on the PRU core. However, it does support an option for PRU to snoop or monitor the received data through the RX L2, shown in Figure 30-41. The PRU does not access data and status bits through R31, and it does not modify and push data.

Figure 30-40. Auto-forward

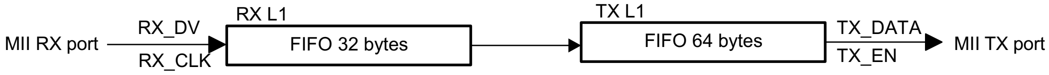
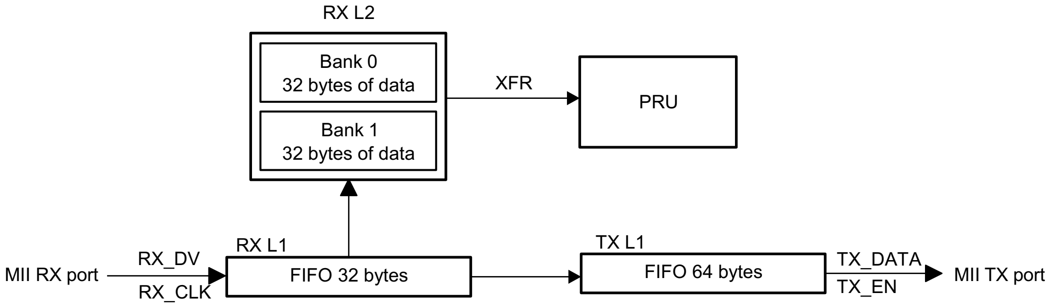


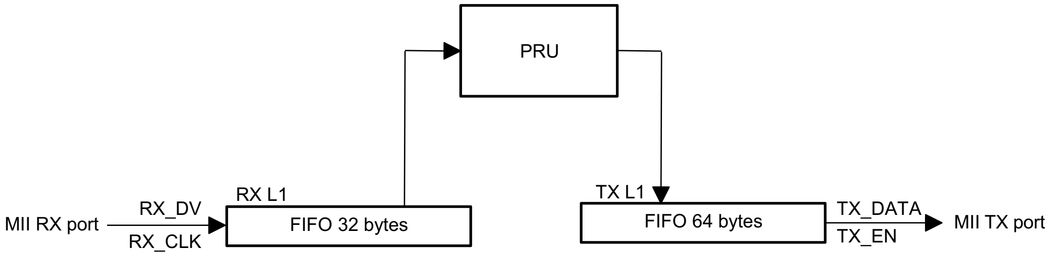
Figure 30-41. Auto-forward with PRU Snoop



30.4.6.2.1.2 8- or 16-bit Processing with On-the-Fly Modifications

This configuration services one byte or word in flight and has low latency. The PRU has the option to manipulate the received word and control popping data from the RX L1 FIFO and pushing it on the TX L1 FIFO.

Figure 30-42. 8- or 16-bit Processing with On-the-Fly Modifications

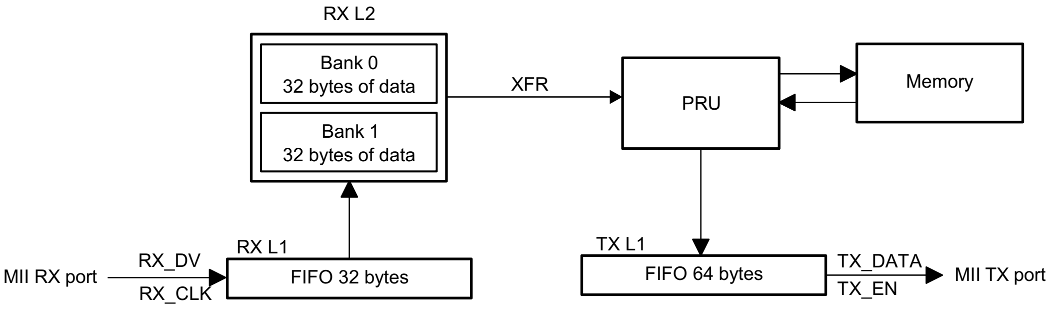


30.4.6.2.1.3 32-byte Double Buffer or Ping-Pong Processing

This configuration supports high bandwidth, high efficiency transactions. Often implementations using this mode permit relaxed servicing requirements allowing the PRU to manipulate the received data before transmitting.

Data received in this configuration is passed into the RX L2 buffer. The PRU reads multiple bytes of data from one of the RX L2 banks through the high bandwidth broadside interface and XFR instructions. The PRU can then store or manipulate data before pushing it to the TX L1 FIFO for transmission on the MII TX port.

Figure 30-43. 32-byte Double Buffer or Ping-Pong Processing



30.4.6.2.2 Definition and Terms

30.4.6.2.2.1 Data Frame Structure

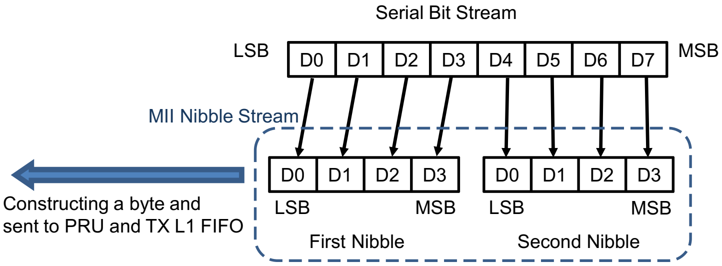
The data received and transmitted over MII conforms with the following frame structure:

Table 30-39. Frame Structure

Inter-frame	Preamble	Start of Frame Delimiter (SFD)	Data	Cyclic Redundancy Check (CRC)

The data following the SFD is formatted in a 4-bit nibble structure. Figure 30-44 illustrates the nibble order. The MSB arriving first is on the LSB side of a nibble. When receiving data, the MII_RT receive logic will wait for the next nibble to arrive before constructing a byte and delivering to the PRU.

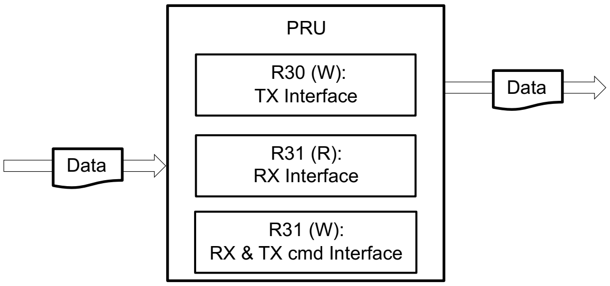
Figure 30-44. Data Nibble Structure



30.4.6.2.2.2 PRU R30 and R31

The PRU registers R30 and R31 are used to receive, transmit, and control the data for the PRU. As shown in Figure 30-45, the R31 is used to access data in the RX L1 FIFO, the R30 is used to transmit data from the PRU, and the R31 output is used the control the flow of receive and transmit. For more details about these registers, see the following sections.

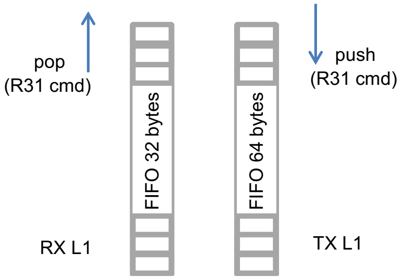
Figure 30-45. PRU R30, R31 Operations



30.4.6.2.2.3 RX and TX L1 FIFO Data Movement

To advance the next data byte seen by R31, the PRU must pop the data from the RX L1 FIFO. Likewise, the PRU can push the data from R30 to the TX L1 FIFO. These operations are illustrated in Figure 30-46.

Figure 30-46. Reading and Writing FIFO Data



30.4.6.2.2.4 CRC Computation

30.4.6.2.2.4.1 Receive CRC Computation

For the incoming data, the MII_RT calculates CRC32 and then compares against the value provided in the incoming frame. If there is a mismatch, the MII RT signals ERROR_CRC to the PRU. If a previous node or Ethernet device appended an error nibble, the CRC calculation of received packet will be wrong because the longer frame and the frame length will end at a 4-bit boundary instead of the usual 8-bit boundary. When RX_DV goes inactive on the 4-bit boundary, the interface will assert DATA_RDY and BYTE_RDY flag with the ERROR NIBBLE. The error event is also mapped into the PRU-ICSS INTC.

30.4.6.2.2.4.2 Transmit CRC Computation

For the outgoing data, the MII_RT calculates the CRC32 value and inserts it into outgoing packets. The CRC value computed on each MII transmit path is also available in memory map registers (MMRs) that can be read by the PRU and used primarily for debug and diagnostic purposes. The CRC is inserted into the outgoing packet based on the commands received through the R31 register of the PRU. The CRC will be inserted into the TX L1 FIFO, and there must be enough room to store the CRC value in the FIFO or else the FIFO will overflow. As Table 30-40 shows, the CRC programming model supports three sequences that provide more flexibility. Note “cmdR31” indicates write to the mentioned bits of the R31 command interface.

Table 30-40. TX CRC Programming Models

Option 1	Step 1: cmdR31 [TX_CRC_HIGH + TX_CRC_LOW + TX_EOF]
Option 2	Step 1: cmdR31 [TX_CRC_HIGH] Step 2: wait > 6 clocks (PRU cycles) Step 3: cmdR31 [TX_CRC_LOW + TX_EOF]
Option 3	Step 1: cmdR31 [TX_CRC_HIGH] Step 2: wait > 6 clocks (PRU cycles) Step 3: read TXCRC0/1 Step 4: modify CRC[15:0] Step 5: cmdR31 [TX_PUSH16 + TX_EOF + TX_ERROR_NIBBLE]

NOTE: If TX_AUTO_PREAMBLE = 1, then the first push of payload into the TX FIFO must be a byte push (TX_PUSH8). If the first push is a word (TX_PUSH16), then the CRC is not calculated correctly.

30.4.6.2.3 RX MII Interface

30.4.6.2.3.1 RX MII Submodule Overview

The RX MII interface is composed of multiple submodules that process the incoming frames and pass receive data and status information into the PRU register R31. These submodules include:

- Latch received data
- Start of frame detection
- Start frame delimiter detection
- CRC calculation and error detection
- Enhanced link detection through RX error detection

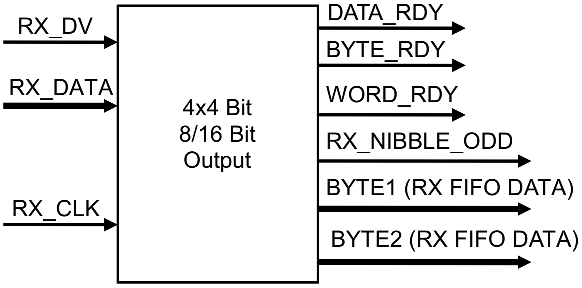
Table 30-41 includes more details about the internal signals and output of these submodules.

30.4.6.2.3.1.1 Receive Data Latch

The receive data from the MII interface is stored in the receive data FIFO which is 32 bytes. The PRU can access this data through the register R31. Depending on the configuration settings, the data can be latched on reception of one or two bytes. In each scheme, the configured number of nibbles is assembled before being copied into the PRU registers. Figure 30-47 shows the inputs and outputs of the data latch logic block.

The receiver logic in MII_RT can be programmed through the RXCFG0 and RXCFG1 registers to remove or retain the preamble + SFD from incoming frames.

Figure 30-47. RX Data Latch



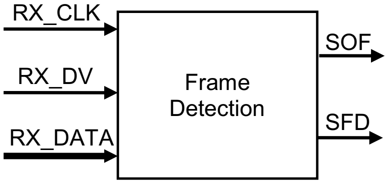
30.4.6.2.3.1.1.1 Start of Frame Detection

The start of frame detection logic tracks the frame boundaries and signals the beginning of a frame to other components of the PRU-ICSS. This logic detects two events:

- Start of Frame (SOF) event that occurs when Receive Data Valid MII signal is sampled high.
- Start of Frame Delimiter (SFD) event is seen on MII Receive Data bus.

These event triggers can be used to add timestamp to the frames. The notification for these events is available through R31 as well as through INTC which is integrated in the PRU-ICSS. Figure 30-48 shows the inputs and outputs of the start of frame detection logic block.

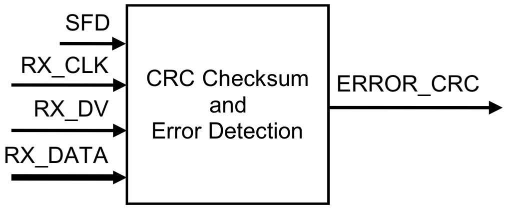
Figure 30-48. Start of Frame Detection



30.4.6.2.3.1.1.2 CRC Error Detection

For each incoming frame, the CRC is calculated by the MII_RT and compared against the CRC included in the frame. When the two values do not match, a CRC error is flagged. The ERROR_CRC indication is available in the register interface (PRU R31 Receive Interface) as well as in the FIFO interface (RX L2 Status Interface). It is also provided to the INTC which is integrated in the PRU-ICSS. Figure 30-49 shows the inputs and outputs of CRC error detection logic block.

Figure 30-49. CRC Error Detection

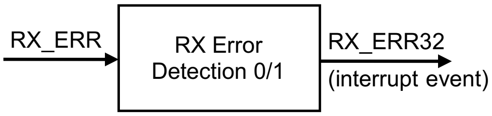


30.4.6.2.3.1.1.3 RX Error Detection and Action

The RX error detection logic tracks the receive error signaled by the physical layer and informs the PRU-ICSS INTC whenever an error is detected. Figure 30-50 shows the inputs and outputs of the RX error detection logic block. Note the following dependencies:

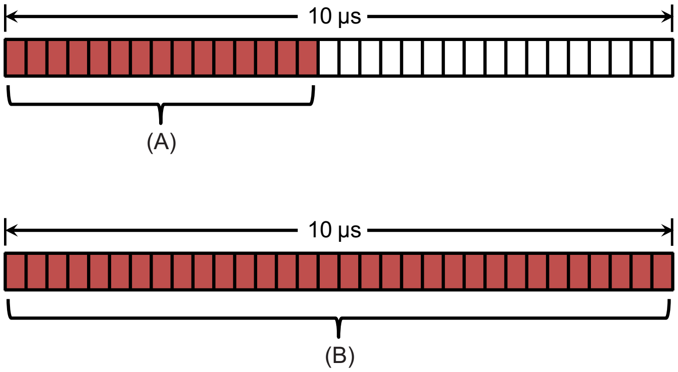
- RX_ERR signal is only sampled when RX_DV is asserted.
- All nibbles are discarded post RX_ERR event, including the nibble which had RX_ERR asserted. This state will remain until EOF occurs.
- Due to this fact, RX L1 FIFO and RX L2 FIFO will never receive any data with RX_ERR or post RX_ERR during that frame.

Figure 30-50. RX Error Detection



This submodule also keeps track of a running count of receive error events within a 10 µs error detection window, as shown in Figure 30-51. The INTC is notified when 32 or more events have occurred in a 10 µs error detection window. The error detection window is not a sliding window but a non-overlapping window with no specific initialization time with respect to incoming traffic. The timer starts its 10 µs counts immediately after de-assertion of reset to the MII_RT module.

Figure 30-51. Error Detection Window with Running Counter



A There are fewer than 32 consecutive error events in the 10 µs window. The detection module will not forward to the interrupt controller (INTC).

B There are more or equal to 32 error events in the 10 µs window. The detection module will notify the interrupt controller (INTC).

30.4.6.2.3.1.2 RX Data Path Options to PRU

There are two data path options for delivering received data to the PRU, described further in the subsequent sections:

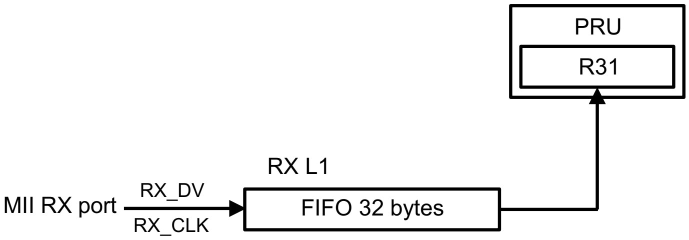
1. RX MII port → RX L1 FIFO → PRU (one word in flight)
2. RX MII port → RX L1 FIFO → RX L2 buffer → PRU (multi-word in flight)

Once the PRU has received RX data, the PRU can both manipulate received data or send data to the TX MII Interface.

30.4.6.2.3.1.2.1 RX MII Port → RX L1 FIFO → PRU

The RX L1 FIFO to PRU interface is depicted in Figure 30-52. In this mode, the data received from the MII interface is fed into the 32-byte RX L1 FIFO. The first data byte into the FIFO is automatically available in R31 of the PRU. Therefore, the PRU firmware can directly operate on this data without having to read it in a separate instruction. This allows the PRU to access receive data with low latency.

Figure 30-52. RX L1 to PRU Interface



When the new data is received, the PRU is provided with up to two bytes at a time in the R31 register, as shown in Figure 30-53. Once the PRU processes the incoming data, it instructs the MII_RT by writing to the R31 command interface bits to pop one or two bytes of data from the 32-byte RX FIFO. The pop operation causes current contents of R31 to be refreshed with new data from the incoming packet. Each time the data is popped, the status bits change to indicate so. If the pop is completed and there is no new data, the status bits immediately change to indicate no new data. Note the current R31 content, including data, will be lost after issuing the pop operation. If this information needs to be accessed later, the PRU should store the existing R31 content before popping new data.

Figure 30-53. MII RX Data to PRU R31 (R) and RX FIFO

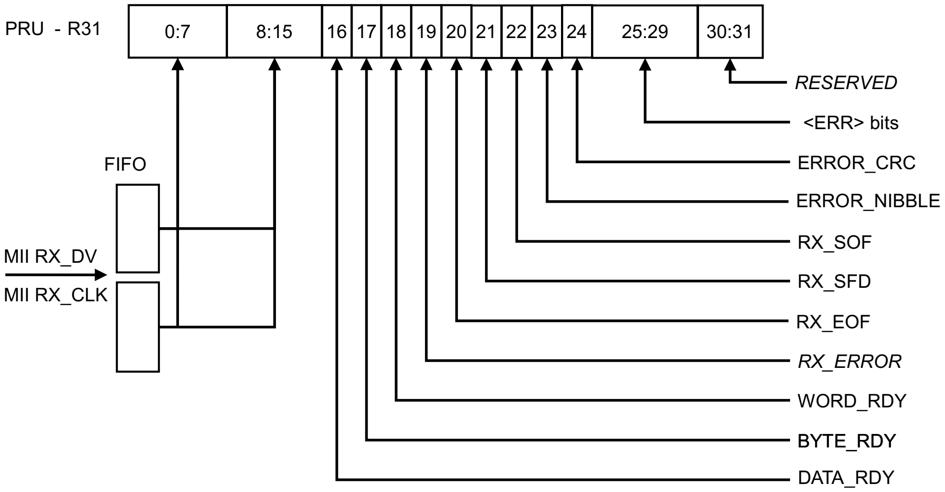


Table 30-41 describes the receive interface data and status contents provided by the R31 register. These contents are available when R31 is read. To configure this register, the PRU GPI mode should be set for MII_RT mode in the CFG register space. Note the following:

1. If the data from receive path is not read in time, it could cause an overflow event because the data is still continuously provided to the 32-byte receive FIFO. Due to the receive FIFO overflow, the data gets automatically discarded to avoid lack of space in the FIFO. At the same time, an interrupt is raised to the INTC through a system event (PRU<n>_RX_OVERFLOW). To detect an overflow condition, the PRU should poll for this system event condition and a RX RESET command through the R31 command interface is required to clear out from this condition. Note that the received Ethernet frame is corrupted and should not be used for further processing as bytes have been dropped due to the overflow condition. A FIFO reset is recommended.
2. The receive data in the R31 register is available following synchronization to the PRU clock domain. So, there is a finite delay (120 ns) when data is available from MII interface and it is accessible to the PRU.
3. The receive FIFO also has the capability to be reset through software. When reset, all contents of receive FIFO are purged and it may result in the current frame not being received as expected. When a frame is being received and the PRU resets the RX FIFO, the remaining frame is not placed into the RX FIFO. However, any new frame arriving on the receive MII port will be stored in the FIFO.

Table 30-41. PRU R31: Receive Interface Data and Status (Read Mode)

Bits	Field Name	Description
31:30	RESERVED	In case of register interface, these bits are provided to PRU by other modules in PRU-ICSS. From the MII_RT module point of view, these bits are always zero.
29	RX_MIN_FRM_CNT_ERR	RX_MIN_FRM_CNT_ERR is set to 1 when the count of total bytes of incoming frame is less than the value defined by RX_MIN_FRM_CNT. RX_MIN_FRM_CNT_ERR is cleared by RX_ERROR_CLR.
28	RX_MAX_FRM_CNT_ERR	RX_MAX_FRM_CNT_ERR is set to 1 when the count of total bytes of incoming frame is more than the value defined by RX_MAX_FRM_CNT_ERR. RX_MAX_FRM_CNT_ERR is cleared by RX_ERROR_CLR.
27	RX_EOF_ERROR	RX_EOF_ERROR is set to 1 when an RX_EOF event or RX_ERROR event occurs. RX_EOF_ERROR is cleared by RX_ EOF_CLR and/or RX_ ERROR_CLR.
26	RX_MAX_PRE_CNT_ERR	RX_MAX_PRE_CNT_ERR is set to 1 when the number of nibbles equaling 0x5 before SFD event (0x5D) is more than the value defined by RXPCNT0/1 [RX_MAX_PRE_CNT]. RX_MAX_PRE_CNT_ERR is cleared by RX_ERROR_CLR.
25	RX_ERR	RX_ERR is set to 1 when pr1_mii0/1_rxer is asserted while pr1_mii0/1_rxdv bit is set. RX_ERR is cleared by RX_ERROR_CLR.
24	ERROR_CRC	ERROR_CRC indicates that the frame has a CRC mismatch. This bit is valid when the RX_EOF bit is set. It should be noted that ERROR_CRC bit is ready in early status, which means it is calculated before data is available in RXL1 FIFO. ERROR_CRC is cleared by RX_ERROR_CLR.
23	ERROR_NIBBLE	ERROR_NIBBLE indicates that the frame ended in odd nibble. It should be considered valid only when the RX_EOF bit and pr1_mii0/1_rxdv are set. Nibble counter is enabled post SFD event. It should be noted that ERROR_NIBBLE bit is ready in early status, which means it is calculated before data is available in RXL1 FIFO. ERROR_NIBBLE is cleared by RX_ERROR_CLR.
22	RX_SOF	RX_SOF transitions from low to high when the frame data starts to arrive and pr1_mii0/1_rxdv is asserted. Note there will be a small sync delay of 0ns – 5ns. The PRU must write one to this bit through the R31 command interface to clear it. The recommended time to clear this bit is at the end of frame (EOF). It should be noted that RX_SOF bit is ready in early status, which means it is calculated before data is available in RXL1 FIFO.
21	RX_SFD	RX_SFD transitions from low to high when the SFD sequence (0x5D) post RX_SOF is observed on the receive MII data. The PRU must write one to this bit through the R31 command interface to clear it. The recommended time to clear this bit is at the end of frame (EOF). It should be noted that RX_SFD bit is ready in early status, which means it is calculated before data is available in RXL1 FIFO.
20	RX_EOF	RX_EOF indicates that the frame has ended and pr1_mii0/1_rxdv is de-asserted. It also validates the CRC match bit. Note there will be a small sync delay of 0ns – 5ns. The PRU must write one to clear this bit in the R31 command interface at the end of the frame. It should be noted that RX_EOF bit is ready in early status, which means it is calculated before data is available in RXL1 FIFO.
19	RX_ERROR	RX_ERROR indicates one or more of the following errors occurred: • RX_MAX/MIN_FRM_CNT_ERR • RX_MAX/MIN_PRE_CNT_ERR • RX_ERR RX_ERROR is cleared by RX_ERROR_CLR.
18	WORD_RDY	WORD_RDY indicates that all four nibbles in R31 have valid data. There is a 2 clock cycle latency from the command RX_POP16 to WORD_RDY update. Therefore, firmware needs to insure it does not read WORD_RDY until 2 clock cycles after RX_POP16.
17	BYTE_RDY	BYTE_RDY indicates that the lower two nibbles in R31 have valid data. There is a 2 clock cycle latency from the command RX_POP8 to BYTE_RDY update. Therefore, PRU firmware needs to insure it does not read BYTE_RDY until 2 clock cycles after RX_POP8.

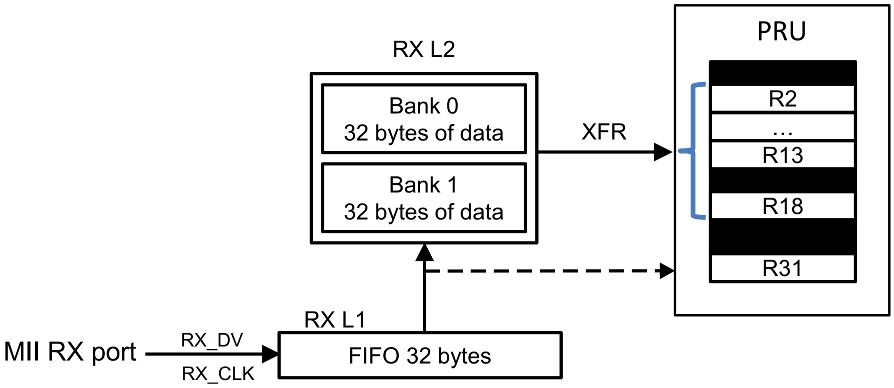
Table 30-41. PRU R31: Receive Interface Data and Status (Read Mode) (continued)

Bits	Field Name	Description
16	DATA_RDY	DATA_RDY indicates there is valid data in R31 ready to be read. This bit goes to zero when the PRU does a POP8/16 and there is no new data left in the receive MII port. This bit is high if there is more receive data for PRU to read. There is a 2 clock cycle latency from the command RX_POP16/8 to WORD_RDY/BYTE_RDY update. Therefore, PRU firmware needs to insure it does not read BYTE_RDY/WORD_RDY until 2 clock cycles after RX_POP16/8.
15:8	BYTE1	Data Byte 1. This data is available such that it is safe to read by the PRU when the DATA_RDY/BYTE_RDY/WORD_RDY bits are asserted.
7:0	BYTE0	Data Byte 0. This data is available such that it is safe to read by the PRU when the DATA_RDY/BYTE_RDY/WORD_RDY bits are asserted.

30.4.6.2.3.1.2.2 RX MII Port → RX L1 FIFO → RX L2 Buffer → PRU

The RX L2 is an optional high performance buffer between the RX L1 FIFO and the PRU. Figure 30-54 illustrates the receive data path using RX L2 buffer. This data path is characterized by multi-word in flight transactions.

Figure 30-54. RX L2 to PRU Interface



The 64-byte RX L2 buffer is divided into two 32 byte banks, or ping/pong buffers. When the RX L2 is enabled, the incoming data from the MII RX port will transmit first to the 32 byte RX L1 FIFO. RX L1 pushes data into RX L2, starting when the first byte is ready until the final EOF marker. The RX L2 buffer does not apply any backpressure to the RX L1 FIFO. Therefore, it is the PRU firmware’s responsibility to fetch the data in RX L2 before it is overwritten by the cyclic buffer. The RX L1 will remain near empty, with only one byte (nibble) stored.

Each RX L2 bank holds up to 32 bytes of data, and every four nibbles (or 16 bits) of data has a corresponding 8-bit status. The data and status information are stored in packed arrays. In each bank, R2 to R9 contains the data packed array and R10 to R13 contains the status packed array. Figure 30-55 shows the relationship of the data registers and status registers. The RX L2 status registers record status information about the received data, such as ERROR_CRC, RX_ERROR, STATUS_RDY, etc. The RX L2 status register details are described in Table 30-42. Note RX_RESET clears all Data and Status elements and resets R18.

Figure 30-55. Data and Status Register Dependency

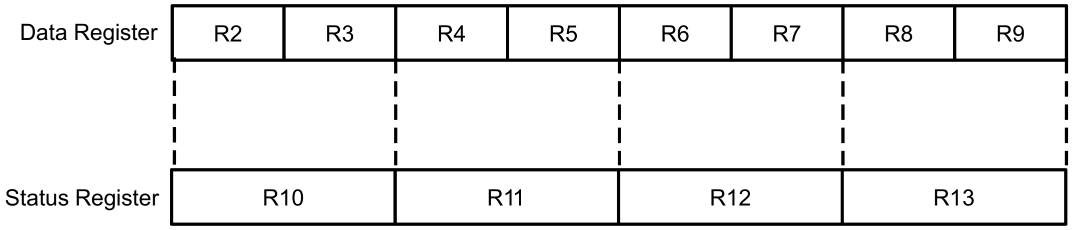


Table 30-42. RX L2 Status

Bit	Field Name	Description
7	ERROR_CRC	ERROR_CRC indicates that the frame has a CRC mismatch. This bit is valid when the RX_EOF bit is set. It should be noted that ERROR_CRC bit is ready in early status, which means it is calculated before data is available in RXL1 FIFO. ERROR_CRC will only be set for one entry, self clear on next entry.
6	ERROR_NIBBLE	ERROR_NIBBLE indicates that the frame ended in odd nibble. It should be considered valid only when the RX_EOF bit and pr1_mii0/1_rxdv are set. Nibble counter is enabled post SFD event. It should be noted that ERROR_NIBBLE bit is ready in early status, which means it is calculated before data is available in RXL1 FIFO. ERROR_NIBBLE will only be set for one entry, self clear on next entry.
5	RX_SOF	RX_SOF transitions from low to high when the frame data starts to arrive and pr1_mii0/1_rxdv is asserted. Note there will be a small sync delay of 0ns – 5ns. The recommended time to clear this bit is at the end of frame (EOF). It should be noted that RX_SOF bit is ready in early status, which means it is calculated before data is available in RXL1 FIFO. RX_SOF will only be set for one entry, self clear on next entry.
4	RX_SFD	RX_SFD transitions from low to high when the SFD sequence (0x5D) post RX_SOF is observed on the receive MII data. The recommended time to clear this bit is at the end of frame (EOF). It should be noted that RX_SFD bit is ready in early status, which means it is calculated before data is available in RXL1 FIFO. RX_SOF will only be set for one entry, self clear on next entry.
3	RX_EOF	RX_EOF indicates that the frame has ended and pr1_mii0/1_rxdv is de-asserted. It also validates the CRC match bit. Note there will be a small sync delay of 0ns – 5ns. It should be noted that RX_EOF bit is ready in early status, which means it is calculated before data is available in RXL1 FIFO. RX_EOF will only be set for one entry, self clear on next entry.
2	RX_ERROR	RX_ERROR indicates one or more of the following errors occurred: • RX_MAX/MIN_FRM_CNT_ERR • RX_MAX/MIN_PRE_CNT_ERR • RX_ERR RX_ERROR is cleared by RX_ERROR_CLR.
1	STATUS_RDY	STATUS_RDY is set when RX_EOF or write pointer advanced by 2. This is a simple method for software to determine if RX_EOF event has occurred or new data is available. If RX_EOF is not set, all status bits are static.
0	RX_ERR	RX_ERR is set to 1 when pr1_mii0/1_rxer is asserted while pr1_mii0/1_rxdv bit is set. It will get set for first pr1_mii0/1_rxer event and self clear on SOF for the next FRAME.

Bank 0 and Bank 1 are used as ping/pong buffers. RX L2 supports the reading of a write pointer in R18 that allows software to determine which bank has active write transactions, as well as the specific write address within packed data arrays.

The PRU interacts with the RX L2 buffer using the high performance XFR read instructions and broadside interface. Table 30-43 shows the device XFR ID numbers for each bank.

Table 30-43. RX L2 XFR ID

Device ID	Function	Description
20	Selects RX L2 Bank0	R2:R9 Data packed array R10:R13 Status packed array
21	Selects RX L2 Bank1	R2:R9 Data packed array R10:R13 Status packed array

Table 30-43. RX L2 XFR ID (continued)

Device ID	Function	Description
20/21	Byte pointer of current write	R18[5:0] Pointer indicating location of current write in data packed array. 0 = Bank0.R2.Byte0 (default and reset value) 1 = Bank0.R2.Byte1 2 = Bank0.R2.Byte2 3 = Bank0.R2.Byte3 4 = Bank0.R3.Byte0 … 63=Bank1.R9.Byte3

XFR read transactions are passive and have no effect on any status or other states in RX L2. The firmware can also read R18 to determine which Bank has active write transactions and the location of the transaction. With this information, the firmware can read multiple times the stable preserved data. Note when RX L1 data is written to RX L2, the next status byte gets cleared at the same time the current status byte gets updated. The rest of the status buffer is persistent. When software is accessing any register of the ping/ pong buffer, software needs to issue an XFER read transaction to fetch the latest/current state of the ping/pong buffer. The PRU registers will not reflect the current snapshot of L2 unless an XFER is issued by software.

30.4.6.2.4 TX MII Interface

Data to be transmitted is loaded into the TX L1 FIFO. The transmit FIFO (TX L1) stores up to 64 bytes of transmit data. From the FIFO, the data is sent to the MII TX port of the PHY by the MII_RT transmit logic.

The transmit FIFO also has the capability to be reset through software (TX_RESET). When reset, all contents of transmit FIFO are purged and this may result in a frame not getting transmitted as expected, if the transmission is already ongoing. Any new data written in the transmit FIFO results in a new frame being composed and transmitted. An overflow event will require a TX_RESET to recover from this condition.

There are four dependencies that must be true for TX_EN to assert.

1. TX L1 FIFO not empty
2. Interpacket gap (IPG) timer expiration
3. RX_DV to TX_EN timer expiration
4. TX_EN compare timer expiration

The transmit interface also provides an underflow error signal in case there was no data loaded when TX_EN triggered. The transmit underflow signal is mapped to the INTC in PRU-ICSS. The PRU firmware must track the FIFO fill level, such as by a timer or the PRU cycle count register (PRU_ICSS_CTRL_CYCLE).The current FIFO fill level cannot be accessed by PRU firmware. The firmware can issue an R31 command via R31 bit 29 (TX_EOF) to indicate that the last byte has been written into the TX FIFO.

30.4.6.2.4.1 TX Data Path Options to TX L1 FIFO

There are two data path options for delivering data to the TX L1 FIFO and transmit port, described further in the subsequent sections:

1. PRU → TX L1 FIFO → TX MII port
2. RX L1 FIFO → TX L1 FIFO → TX MII port

30.4.6.2.4.1.1 PRU → TX L1 FIFO → TX MII Port

The PRU can be used to feed data into the TX L1 FIFO using the R30 and R31 registers, shown in Figure 30-56. The PRU writes up to two bytes of data into R30 and then pushes the data into the TX L1 FIFO by writing to the R31 command interface.

Figure 30-56. PRU to TX L1 Interface

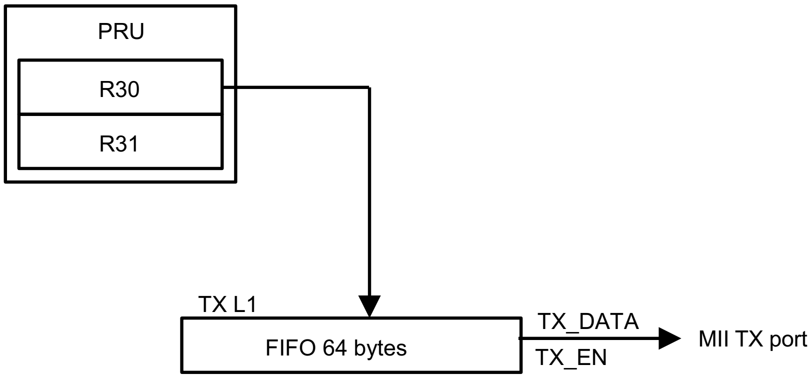


Figure 30-57 shows the R30 transmit interface. The lower 16 bits of the R30 (or FIFO transmit word) contain transmit data nibbles. The upper 16 bits contain mask information. The operation to be performed on the transmit interface is controlled by PRU writes to the R31 command interface. Table 30-44 describes the TXMASK and TXDATA bit fields of the R30 transmit interface.

Figure 30-57. PRU to TX MII Interface

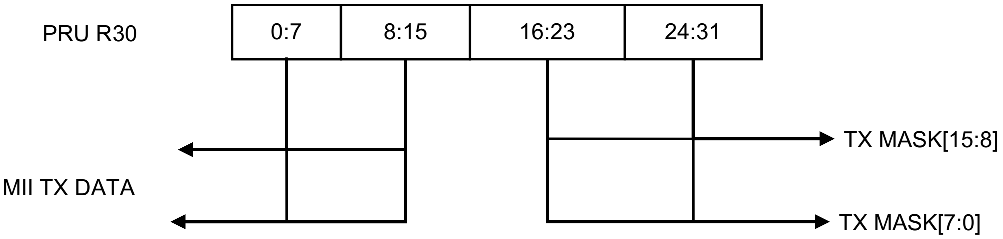


Table 30-44. PRU R30: Transmit Interface

Bits	Field Name	Description
31:16	TXMASK	The TXMASK is used to determine which of RX L1 FIFO received data and R30 data is sent to transmit FIFO. It must to be applied to TXDATA and RXDATA before it is transmitted. To disable TXMASK and transmit only TXDATA via R30, set to 0xFFFF. Note software should not pop the RXDATA from the RX L1 FIFO before pushing the TXDATA. This will cause new data to propagate before the push. Otherwise, software can pop and push on the same command for bytes only or delay the pop after the push for words or bytes.
15:0	TXDATA	Data provided by the PRU to be sent to transmit path after applying the mask. When 16 bits are to be transmitted, all bits of this and TXMASK field are used. When 8 bits are to be transmitted, the bits [7:0] of this and TXMASK field are used.

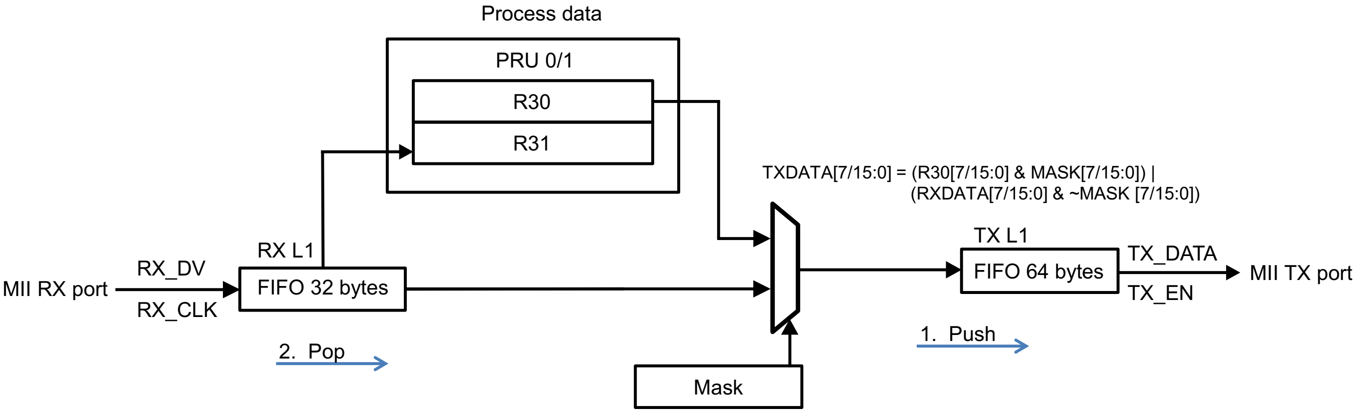
Using the TX mask, the PRU can send a mix of R30 and RX L1 FIFO data to the TX L1 FIFO. Note the TX mask is only available when the PRU is fed one word or byte at a time by the RX L1 FIFO. It is not applicable when the RX L2 buffer is enabled. To disable TX mask, set TXMASK to 0xFFFF.

As shown in Figure 30-58, the PRU drives the MII transmit interface through its R30 register. The contents of R30 and RX data from the receive interface are taken and fed into a 64 byte transmit FIFO.

Before transmission, a mask is applied to the data portion of the R30 register. By using the mask, the PRU firmware can control whether received data from the RX L1 FIFO is sent to transmit, R30 data is sent to transmit, or a mix of the two is sent. The Boolean equation that is used by MII_RT to compose TX data is:

TXDATA[7/15:0] = (R30[7/15:0] & MASK[7/15:0]) | (RXDATA[7/15:0] & ~MASK [7/15:0])

Figure 30-58. TX Mask Mode

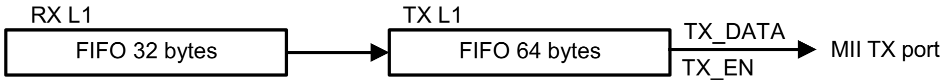


30.4.6.2.4.1.2 RX L1 FIFO → TX L1 FIFO → TX MII Port

When TXCFG0/1[TX_AUTO_SEQUENCE] is set, the data frame is passed from the RX to TX FIFOs without the any interaction of the PRU. This mode of operations is shown in Figure 30-59. The RX L1 will push into TX L1 as long as it is enabled and not full.

There is no PRU dependency in this mode and no option for the PRU to perform any operation to the TX L1 FIFO. RX_RESET clears all data and status elements.

Figure 30-59. RX L1 to TX L1 Interface



30.4.6.2.5 PRU R31 Command Interface

The PRU uses writes to R31[31:16] to control the reception and transmission of packets in register mode. Table 30-45 lists the available commands. Each bit in the table is a single clock pulse output from the PRU. When more than one action is to be performed in the same instant, the PRU firmware must set those command bits in one instruction.

Table 30-45. PRU R31: Command Interface (Write Mode)

Bit	Command	Description
31	TX_CRC_ERR	TX_CRC_ERR command when set will add 0xa5 byte to the TX L1 FIFO if the current FCS is valid. This bit can only be set with the TX_EOF command and optionally with the TX_ERROR_NIBBLE command. It cannot get set with any other commands. Note for proper operations auto-forward preamble must be enabled.
30	TX_RESET	TX_RESET command is used to reset the transmit FIFO and clear all its contents. This is required to recover from a TX FIFO overrun.
29	TX_EOF	TX_EOF command is used to indicate that the data loaded is considered last for the current frame
28	TX_ERROR_NIBBLE	TX_ERROR_NIBBLE command is used to insert an error nibble. This makes the frame invalid. Also, it will add 0x0 after the 32-bit CRC.
27	TX_CRC_HIGH	TX_CRC_HIGH command ends the CRC calculations and pushes CRC[31:16] to append to the outgoing frame in the TX L1 FIFO. Note TXCRC0/1 will become valid after 6 clock cycles.
26	TX_CRC_LOW	TX_CRC_LOW command pushes CRC[15:0] to append to the outgoing frame in the TX L1 FIFO.

Table 30-45. PRU R31: Command Interface (Write Mode) (continued)

Bit	Command	Description
25	TX_PUSH16	TX_PUSH16 command applies mask to two bytes from receive path and transmit. Note TX_PUSH16 needs to occur before TX_POP16 if data is not fully masked. TX CRC requires the data to be valid for 2 clock cycles.
24	TX_PUSH8	TX_PUSH8 command applies mask to one byte from receive path and transmit. Note TX_PUSH8 needs to occur before TX_POP8 if data is not fully masked.
23	RX_ ERROR_CLR	RX_ERROR_CLR command is used to clear RX_ ERROR indicator bit by writing 1.
22	RX_EOF_CLR	RX_EOF_CLR command is used to clear RX_EOF status indicator bit by writing 1.
21	RX_SFD_CLR	RX_SFD_CLR command is used to clear RX_SFD indicator bit by writing 1.
20	RX_SOF_CLR	RX_SOF_CLR command is used to clear RX_SOF indicator bit by writing 1.
19	Reserved	Reserved
18	RX_RESET	RX_RESET is used to reset the receive FIFO and clear all contents. This is required to recover from a RX FIFO overrun, if software does not want to undrain. The typical use case is assertion after RX_EOF. If asserted during an active frame, the following actions will occur: 1. Terminate the current frame 2. Block/terminate all new data 3. Flush/clear all FIFO elements 4. Cause RX state machine into an idle state 5. Cause EOF event 6. Cause minimum frame error, if you abort before minimum size reached
17	RX_POP16	RX_POP16 command advances the receive traffic by two bytes. This is only required when you are using R31 to read the data. After R31[15:0] is ready to read by PRU, it will set 1 to WORD_RDY, and the next new data will be allowed to advance. RX_POP16 to WORD_RDY update has 2 clock cycles latency. Firmware needs to insure it does not read WORD_RDY/BYTE_RDY until 2 clock cycles after RX_POP16.
16	RX_POP8	RX_POP8 command advances the receive traffic by one bytes. This is only required when you are using R31 to read the data. After R31[7:0] is ready to read by PRU, it will set 1 to BYTE_RDY, and the next new data will be allowed to advance. RX_POP8 to BYTE_RDY update has 2 clock cycles latency. Firmware needs to insure it does not read WORD_RDY/BYTE_RDY until 2 clock cycles after RX_POP8.

30.4.6.2.6 Other Configuration Options

30.4.6.2.6.1 Nibble and Byte Order

The PRU core is little endian. To support big endian, the MII_RT supports optional nibble swapping on both the RX and TX side.

On the receive side, the order of the two data bytes in RX R31 and the RX L2 buffer are configurable through the RX_BYTE_SWAP bit in the RXCFG0/1 registers, as shown in Table 30-46. Note the Nibble0 is the first nibble received.

Table 30-46. RX Nibble and Byte Order

Configuration	Order
RXCFG0/1[RX_BYTE_SWAP] = 0 (default)	R31[15:8] / RXL2[15:8] = Byte1{Nibble3,Nibble2} R31[7:0] / RXL2[7:0] = Byte0{Nibble1,Nibble0}

Table 30-46. RX Nibble and Byte Order (continued)

Configuration	Order
RXCFG0/1[RX_BYTE_SWAP] = 1	R31[15:8] / RXL2[15:8] = Byte0{Nibble1,Nibble0} R31[7:0] / RXL2[7:0] = Byte1{Nibble3,Nibble2}

On the transmit side, the order of the two data bytes and mask bytes in TX R30 are configurable through the TX_BYTE_SWAP bit in the TXCFG0/1 registers, as shown in Table 30-47. Note the Nibble0 is the first nibble received.

Table 30-47. TX Nibble and Byte Order

Configuration	Order
TXCFG0/1[TX_BYTE_SWAP] = 0 (default)	R30[15:8] = Byte1{Nibble3,Nibble2} R30[7:0] = Byte0{Nibble1,Nibble0} R30[31:24] = TX_MASK[15:8] R30[23:16] = TX_MASK[7:0]
TXCFG0/1[TX_BYTE_SWAP] = 1	R30[15:8] = Byte0{Nibble1,Nibble0} R30[7:0] = Byte1{Nibble3,Nibble2} R30[31:24] = TX_MASK[7:0] R30[23:16] = TX_MASK[15:8]

30.4.6.2.6.2 Preamble Source

The MII_RT module has the option to preserve and forward a received preamble in the TX data stream, use a preamble provided by the PRU, or auto-generate a preamble. These configurations are highlighted in Table 30-48.

Table 30-48. Preamble Configuration Options

RX_CUT_PREAMBLE	Determines whether RX preamble is passed to the RX L1/L2 FIFO
RX_AUTO_FWD_PRE	Determines whether RX preamble is automatically passed to TX L1 FIFO
TX_AUTO_PREAMBLE	TX interface logic auto-generates and appends preamble to TX data stream with the first push of data into the TX L1 FIFO. Note that enabling this option does fill the TX FIFO with the preamble length, hence software has to consider this to not overrun the TX FIFO.

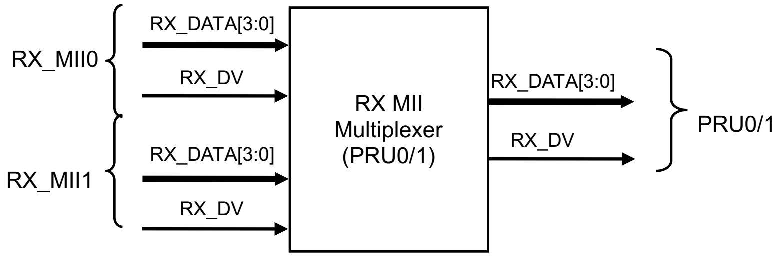
30.4.6.2.6.3 PRU and MII Port Multiplexer

The MII_RT module supports configurable PRU core to MII TXn / RXn port mapping. By default, PRU0 is mapped to TX1 and RX0 and PRU1 is mapped to TX0 and RX1. However, the system supports the flexibility to map any PRU core to any TX and RX port. Note the mapping options are destination fixed. For example, the input to PRU0 can be either RX_MII0 or RX_MII1. Similarly, the input to TX_MII0 can be either PRU0 or PRU1.

30.4.6.2.6.3.1 Receive Multiplexer

A multiplexer is provided to allow selecting either of the two MII interfaces for the receive data that is sent to PRU. Figure 30-60 shows the symbol of receive multiplexer of PRU.

Figure 30-60. MII Receive Multiplexer

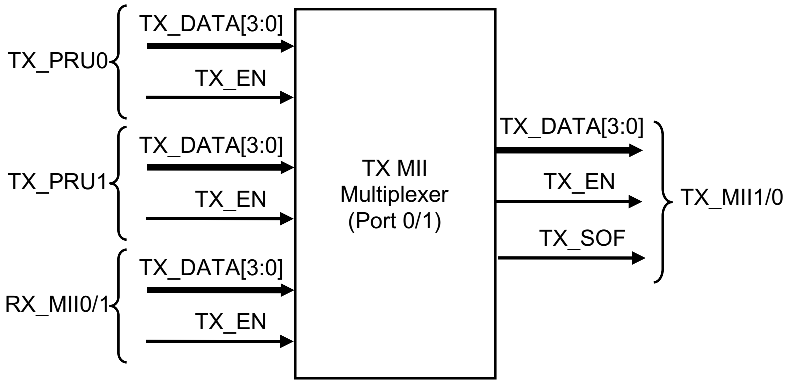


There are two receive multiplexer instances to enable selection of RX MII path for each PRU. The select lines of the RX multiplexers are driven from the PRU-ICSS programmable registers (RXCFG0/1).

30.4.6.2.6.3.2 Transmit Multiplexer

On the MII transmit ports, there is a multiplexer for each MII transmit port that enables selection of either the transmit data from the PRUs or from the RX MII interface of the other MII interface. Figure 30-61 shows the symbol of transmit multiplexer of PRU.

Figure 30-61. MII Transmit Multiplexer

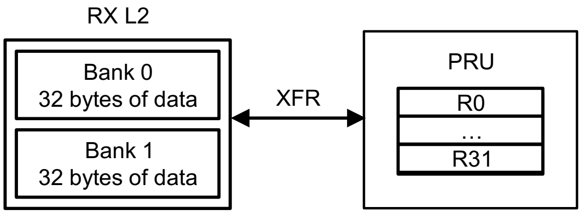


The transmit multiplexers enable the PRU-ICSS to either operate in a bypass mode where the PRU is not involved in processing MII traffic or use of one of the PRU cores for transmitting data into the MII interface. There are two instances of the TX MII multiplexer and the select lines for each TX multiplexer are provided by the PRU-ICSS. The select lines are common between register and FIFO interface. It is expected that the select lines will not change during the course of a frame so that can avoid data exchange error.

30.4.6.2.6.4 RX L2 Scratch Pad

When the RX L2 is disabled (RXCFG0/1[RX_L2_ENABLE] = 0), the RX L2 banks can be used as a generic scratch pad. In scratch pad mode, RX L2 Bank0 and RX L2 Bank1 operate like simple write/read memory mapped registers (MMRs). All XFR size and start operations are supported. RX_RESET has no effect in this mode. This mode is shown in Figure 30-62.

Figure 30-62. Scratch Pad Mode



30.4.6.3 Interrupts

The MII_RT tracks multiple events that could lead to generation of interrupt to the INTC which integrated in PRU-ICSS. These events are classified into receive, transmit and MII link events. Each event can occur on either of the two MII interfaces. Table 30-49 lists all interrupts from MII_RT to PRU-ICSS INTC module. Note All events to INTC are pulse type.

Table 30-49. Interrupt Events in MII_RT

Index	Name	Description
0	PRU0_RX_ERR	Error occurs on receive packet attached to PRU0.
1	PRU0_RX_ERR32	There are over 32 errors in a 10us window on receive packets attached to PRU0.
2	PRU0_RX_SFD	Start of frame delimiter detects 0x5D event attached to PRU0.
3	PRU0_RX_SOF	Start of frame detects event attached to PRU0.
4	PRU0_RX_CRC	CRC mismatch error occurs on receive packet attached to PRU0.
5	PRU0_RX_NIBBLE_ODD	There are odd nibbles in a received frame attached to PRU0.
6	PRU0_RX_OVERFLOW	RX L1 FIFO is overflows attached to PRU0.
7	PORT0_TX_UNDERFLOW	TX L1 FIFO is underflow attached to PORT0.
8	PORT0_TX_OVERFLOW	TX L1 FIFO is overflow attached to PORT0.
9	PORT0_MII_LINK	PHY link Interrupt attached to PORT0.
10	PRU0_RX_EOF	End of frame detects event attached to PRU0.
11	RESERVED	Reserved
12	PRU1_RX_ERR	Error occurs on receive packet attached to PRU1.
13	PRU1_RX_ERR32	There are over 32 errors in a 10us window on receive packets attached to PRU1.
14	PRU1_RX_SFD	Start of frame delimiter detects 0x5D event attached to PRU1.
15	PRU1_RX_SOF	Start of frame detects event attached to PRU1.
16	PRU1_RX_CRC	CRC mismatch error occurs on receive packet attached to PRU1.
17	PRU1_RX_NIBBLE_ODD	There are odd nibbles in a received frame attached to PRU1.
18	PRU1_RX_OVERFLOW	RX L1 FIFO is overflows attached to PRU1.
19	PORT1_TX_UNDERFLOW	TX L1 FIFO is underflow attached to PORT1.
20	PORT1_TX_OVERFLOW	TX L1 FIFO is overflow attached to PORT1.
21	PORT1_MII_LINK	PHY link Interrupt attached to PORT1.
22	PRU1_RX_EOF	End of frame detects event attached to PRU1.
23	RESERVED	Reserved

30.4.7 MDIO

The MDIO module within the PRU-ICSS is identical to the MDIO module in the AM437xAM437x Ethernet Subsystem. For additional details about the MDIO module, see Section 15.3.8, *MDIOAM437x ARM® Cortex™-A9 Processors Technical Reference Manual* (SPRUHL7)..

30.5 Registers

30.5.1 PRU_ICSS_PRU_CTRL Registers

Table 30-50 lists the memory-mapped registers for the PRU_ICSS_PRU_CTRL. All register offset addresses not listed in Table 30-50 should be considered as reserved locations and the register contents should not be modified.

Table 30-50. PRU_ICSS_PRU_CTRL REGISTERS

Offset	Acronym	Register Name	Section
0h	PRU_ICSS_CTRL		Section 30.5.1.1
4h	PRU_ICSS_CTRL_STS		Section 30.5.1.2
8h	PRU_ICSS_CTRL_WAKEUP_EN		Section 30.5.1.3
Ch	PRU_ICSS_CTRL_CYCLE		Section 30.5.1.4
10h	PRU_ICSS_CTRL_STALL		Section 30.5.1.5
20h	PRU_ICSS_CTRL_CTBIR0		Section 30.5.1.6
24h	PRU_ICSS_CTRL_CTBIR1		Section 30.5.1.7
28h	PRU_ICSS_CTRL_CTPPR0		Section 30.5.1.8
2Ch	PRU_ICSS_CTRL_CTPPR1		Section 30.5.1.9

30.5.1.1 PRU_ICSS_CTRL Register (offset = 0h) [reset = 1h]

PRU_ICSS_CTRL is shown in Figure 30-63 and described in Table 30-51.

CONTROL REGISTER

Figure 30-63. PRU_ICSS_CTRL Register

31	30	29	28	27	26	25	24
PCTR_RST_VAL							
R/W-0h							

23	22	21	20	19	18	17	16
PCTR_RST_VAL							
R/W-0h							

15	14	13	12	11	10	9	8
RUNSTATE	RESERVED	RESERVED					SINGLE_STEP
R-0h	R-0h	R/W-0h					R/W-0h

7	6	5	4	3	2	1	0
RESERVED				CTR_EN	SLEEPING	EN	SOFT_RST_N
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-1h

Table 30-51. PRU_ICSS_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCTR_RST_VAL	R/W	0h	Program Counter Reset Value: This field controls the address where the PRU will start executing code from after it is taken out of reset.
15	RUNSTATE	R	0h	Run State: This bit indicates whether the PRU is currently executing an instruction or is halted. This bit is used by an external debug agent to know when the PRU has actually halted when waiting for a HALT instruction to execute, a single step to finish, or any other time when the pru_enable has been cleared. 0h (R/W) = PRU is halted and host has access to the instruction RAM and debug registers regions. 1h (R/W) = PRU is currently running and the host is locked out of the instruction RAM and debug registers regions.
14	RESERVED	R	0h	Reserved.
13-9	RESERVED	R/W	0h	
8	SINGLE_STEP	R/W	0h	Single Step Enable: This bit controls whether or not the PRU will only execute a single instruction when enabled. Note that this bit does not actually enable the PRU, it only sets the policy for how much code will be run after the PRU is enabled. The pru_enable bit must be explicitly asserted. It is legal to initialize both the SINGLE_STEP and EN bits simultaneously. (Two independent writes are not required to cause the stated functionality.) 0h (R/W) = PRU will free run when enabled. 1h (R/W) = PRU will execute a single instruction and then the pru_enable bit will be cleared.
7-4	RESERVED	R/W	0h	
3	CTR_EN	R/W	0h	PRU Cycle Counter Enable: Enables PRU cycle counters. 0h (R/W) = Counters not enabled 1h (R/W) = Counters enabled
2	SLEEPING	R/W	0h	PRU Sleep Indicator: This bit indicates whether or not the PRU is currently asleep. 0h (R/W) = PRU is not asleep 1h (R/W) = PRU is asleep; If this bit is written to a 0, the PRU will be forced to power up from sleep mode.

Table 30-51. PRU_ICSS_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	EN	R/W	0h	Processor Enable: This bit controls whether or not the PRU is allowed to fetch new instructions. If this bit is de-asserted while the PRU is currently running and has completed the initial cycle of a multi-cycle instruction (LBxO, SBxO, etc.), the current instruction will be allowed to complete before the PRU pauses execution. Otherwise, the PRU will halt immediately. Because of the unpredictability/timing sensitivity of the instruction execution loop, this bit is not a reliable indication of whether or not the PRU is currently running. The pru_state bit should be consulted for an absolute indication of the run state of the core. When the PRU is halted, its internal state remains coherent therefore this bit can be reasserted without issuing a software reset and the PRU will resume processing exactly where it left off in the instruction stream. 0h (R/W) = PRU is disabled. 1h (R/W) = PRU is enabled.
0	SOFT_RST_N	R/W	1h	Soft Reset: When this bit is cleared, the PRU will be reset. This bit is set back to 1 on the next cycle after it has been cleared.

30.5.1.2 PRU_ICSS_CTRL_STS Register (offset = 4h) [reset = 0h]

PRU_ICSS_CTRL_STS is shown in Figure 30-64 and described in Table 30-52.

STATUS REGISTER

Figure 30-64. PRU_ICSS_CTRL_STS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PCTR															
R-0h																R-0h															

Table 30-52. PRU_ICSS_CTRL_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	PCTR	R	0h	Program Counter: This field is a registered (1 cycle delayed) reflection of the PRU program counter. Note that the PC is an instruction address where each instruction is a 32 bit word. This is not a byte address and to compute the byte address just multiply the PC by 4 (PC of 2 = byte address of 0x8, or PC of 8 = byte address of 0x20).

30.5.1.3 PRU_ICSS_CTRL_WAKEUP_EN Register (offset = 8h) [reset = 0h]

PRU_ICSS_CTRL_WAKEUP_EN is shown in Figure 30-65 and described in Table 30-53.

WAKEUP ENABLE REGISTER

Figure 30-65. PRU_ICSS_CTRL_WAKEUP_EN Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">BITWISE_ENS</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 30-53. PRU_ICSS_CTRL_WAKEUP_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BITWISE_ENS	R/W	0h	Wakeup Enables: This field is ANDed with the incoming R31 status inputs (whose bit positions were specified in the stmap parameter) to produce a vector which is unary ORed to produce the status_wakeup source for the core. Setting any bit in this vector will allow the corresponding status input to wake up the core when it is asserted high. The PRU should set this enable vector prior to executing a SLP (sleep) instruction to ensure that the desired sources can wake up the core.

30.5.1.4 PRU_ICSS_CTRL_CYCLE Register (offset = Ch) [reset = 0h]

PRU_ICSS_CTRL_CYCLE is shown in Figure 30-66 and described in Table 30-54.

CYCLE COUNT. This register counts the number of cycles for which the PRU has been enabled.

Figure 30-66. PRU_ICSS_CTRL_CYCLE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CYCLECOUNT																															
0h																															

Table 30-54. PRU_ICSS_CTRL_CYCLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CYCLECOUNT		0h	This value is incremented by 1 for every cycle during which the PRU is enabled and the counter is enabled (both bits ENABLE and COUNTENABLE set in the PRU control register). Counting halts while the PRU is disabled or counter is disabled, and resumes when re-eneabled. Counter clears the COUNTENABLE bit in the PRU control register when the count reaches 0xFFFFFFFF. (Count does does not wrap). The register can be read at any time. The register can be cleared when the counter or PRU is disabled. Clearing this register also clears the PRU Stall Count Register.

30.5.1.5 PRU_ICSS_CTRL_STALL Register (offset = 10h) [reset = 0h]

PRU_ICSS_CTRL_STALL is shown in Figure 30-67 and described in Table 30-55.

STALL COUNT. This register counts the number of cycles for which the PRU has been enabled, but unable to fetch a new instruction. It is linked to the Cycle Count Register (0x0C) such that this register reflects the stall cycles measured over the same cycles as counted by the cycle count register. Thus the value of this register is always less than or equal to cycle count.

Figure 30-67. PRU_ICSS_CTRL_STALL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">STALLCOUNT</td></tr>
<tr><td colspan="32">0h</td></tr>
</table>

Table 30-55. PRU_ICSS_CTRL_STALL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STALLCOUNT		0h	This value is incremented by 1 for every cycle during which the PRU is enabled and the counter is enabled (both bits ENABLE and COUNTENABLE set in the PRU control register), and the PRU was unable to fetch a new instruction for any reason.

30.5.1.6 PRU_ICSS_CTRL_CTBIR0 Register (offset = 20h) [reset = 0h]

PRU_ICSS_CTRL_CTBIR0 is shown in Figure 30-68 and described in Table 30-56.

CONSTANT TABLE BLOCK INDEX REGISTER 0. This register is used to set the block indices which are used to modify entries 24 and 25 in the PRU Constant Table. This register can be written by the PRU whenever it needs to change to a new base pointer for a block in the State / Scratchpad RAM. This function is useful since the PRU is often processing multiple processing threads which require it to change contexts. The PRU can use this register to avoid requiring excessive amounts of code for repetitive context switching. The format of this register is as follows:

Figure 30-68. PRU_ICSS_CTRL_CTBIR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								C25_BLK_IDX							
R/W-0h								R/W-0h							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								C24_BLK_IDX							
R/W-0h								R/W-0h							

Table 30-56. PRU_ICSS_CTRL_CTBIR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-16	C25_BLK_IDX	R/W	0h	PRU Constant Entry 25 Block Index: This field sets the value that will appear in bits 11 to 8 of entry 25 in the PRU Constant Table.
15-8	RESERVED	R/W	0h	
7-0	C24_BLK_IDX	R/W	0h	PRU Constant Entry 24 Block Index: This field sets the value that will appear in bits 11 to 8 of entry 24 in the PRU Constant Table.

30.5.1.7 PRU_ICSS_CTRL_CTBIR1 Register (offset = 24h) [reset = 0h]

PRU_ICSS_CTRL_CTBIR1 is shown in Figure 30-69 and described in Table 30-57.

CONSTANT TABLE BLOCK INDEX REGISTER 1. This register is used to set the block indices which are used to modify entries 24 and 25 in the PRU Constant Table. This register can be written by the PRU whenever it needs to change to a new base pointer for a block in the State / Scratchpad RAM. This function is useful since the PRU is often processing multiple processing threads which require it to change contexts. The PRU can use this register to avoid requiring excessive amounts of code for repetitive context switching. The format of this register is as follows:

Figure 30-69. PRU_ICSS_CTRL_CTBIR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								C27_BLK_IDX							
R/W-0h								R/W-0h							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								C26_BLK_IDX							
R/W-0h								R/W-0h							

Table 30-57. PRU_ICSS_CTRL_CTBIR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-16	C27_BLK_IDX	R/W	0h	PRU Constant Entry 27 Block Index: This field sets the value that will appear in bits 11 to 8 of entry 27 in the PRU Constant Table.
15-8	RESERVED	R/W	0h	
7-0	C26_BLK_IDX	R/W	0h	PRU Constant Entry 26 Block Index: This field sets the value that will appear in bits 11 to 8 of entry 26 in the PRU Constant Table.

30.5.1.8 PRU_ICSS_CTRL_CTPPR0 Register (offset = 28h) [reset = 0h]

PRU_ICSS_CTRL_CTPPR0 is shown in Figure 30-70 and described in Table 30-58.

CONSTANT TABLE PROGRAMMABLE POINTER REGISTER 0. This register allows the PRU to set up the 256-byte page index for entries 28 and 29 in the PRU Constant Table which serve as general purpose pointers which can be configured to point to any locations inside the session router address map. This register is useful when the PRU needs to frequently access certain structures inside the session router address space whose locations are not hard coded such as tables in scratchpad memory. This register is formatted as follows:

Figure 30-70. PRU_ICSS_CTRL_CTPPR0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">C29_POINTER</td><td colspan="16">C28_POINTER</td></tr>
<tr><td colspan="16">R/W-0h</td><td colspan="16">R/W-0h</td></tr>
</table>

Table 30-58. PRU_ICSS_CTRL_CTPPR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	C29_POINTER	R/W	0h	PRU Constant Entry 29 Pointer: This field sets the value that will appear in bits 23 to 8 of entry 29 in the PRU Constant Table.
15-0	C28_POINTER	R/W	0h	PRU Constant Entry 28 Pointer: This field sets the value that will appear in bits 23 to 8 of entry 28 in the PRU Constant Table.

30.5.1.9 PRU_ICSS_CTRL_CTPPR1 Register (offset = 2Ch) [reset = 0h]

PRU_ICSS_CTRL_CTPPR1 is shown in Figure 30-71 and described in Table 30-59.

CONSTANT TABLE PROGRAMMABLE POINTER REGISTER 1. This register functions the same as the PRU Constant Table Programmable Pointer Register 0 but allows the PRU to control entries 30 and 31 in the PRU Constant Table. This register is formatted as follows:

Figure 30-71. PRU_ICSS_CTRL_CTPPR1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">C31_POINTER</td><td colspan="16">C30_POINTER</td></tr>
<tr><td colspan="16">R/W-0h</td><td colspan="16">R/W-0h</td></tr>
</table>

Table 30-59. PRU_ICSS_CTRL_CTPPR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	C31_POINTER	R/W	0h	PRU Constant Entry 31 Pointer: This field sets the value that will appear in bits 23 to 8 of entry 31 in the PRU Constant Table.
15-0	C30_POINTER	R/W	0h	PRU Constant Entry 30 Pointer: This field sets the value that will appear in bits 23 to 8 of entry 30 in the PRU Constant Table.

30.5.2 PRU_ICSS_PRU_DEBUG Registers

Table 30-60 lists the memory-mapped registers for the PRU_ICSS_PRU_DEBUG. All register offset addresses not listed in Table 30-60 should be considered as reserved locations and the register contents should not be modified.

Table 30-60. PRU_ICSS_PRU_DEBUG Registers

Offset	Acronym	Register Name	Section
0h	PRU_ICSS_DBG_GPREG0		Section 30.5.2.1
4h	PRU_ICSS_DBG_GPREG1		Section 30.5.2.2
8h	PRU_ICSS_DBG_GPREG2		Section 30.5.2.3
Ch	PRU_ICSS_DBG_GPREG3		Section 30.5.2.4
10h	PRU_ICSS_DBG_GPREG4		Section 30.5.2.5
14h	PRU_ICSS_DBG_GPREG5		Section 30.5.2.6
18h	PRU_ICSS_DBG_GPREG6		Section 30.5.2.7
1Ch	PRU_ICSS_DBG_GPREG7		Section 30.5.2.8
20h	PRU_ICSS_DBG_GPREG8		Section 30.5.2.9
24h	PRU_ICSS_DBG_GPREG9		Section 30.5.2.10
28h	PRU_ICSS_DBG_GPREG10		Section 30.5.2.11
2Ch	PRU_ICSS_DBG_GPREG11		Section 30.5.2.12
30h	PRU_ICSS_DBG_GPREG12		Section 30.5.2.13
34h	PRU_ICSS_DBG_GPREG13		Section 30.5.2.14
38h	PRU_ICSS_DBG_GPREG14		Section 30.5.2.15
3Ch	PRU_ICSS_DBG_GPREG15		Section 30.5.2.16
40h	PRU_ICSS_DBG_GPREG16		Section 30.5.2.17
44h	PRU_ICSS_DBG_GPREG17		Section 30.5.2.18
48h	PRU_ICSS_DBG_GPREG18		Section 30.5.2.19
4Ch	PRU_ICSS_DBG_GPREG19		Section 30.5.2.20
50h	PRU_ICSS_DBG_GPREG20		Section 30.5.2.21
54h	PRU_ICSS_DBG_GPREG21		Section 30.5.2.22
58h	PRU_ICSS_DBG_GPREG22		Section 30.5.2.23
5Ch	PRU_ICSS_DBG_GPREG23		Section 30.5.2.24
60h	PRU_ICSS_DBG_GPREG24		Section 30.5.2.25
64h	PRU_ICSS_DBG_GPREG25		Section 30.5.2.26
68h	PRU_ICSS_DBG_GPREG26		Section 30.5.2.27
6Ch	PRU_ICSS_DBG_GPREG27		Section 30.5.2.28
70h	PRU_ICSS_DBG_GPREG28		Section 30.5.2.29
74h	PRU_ICSS_DBG_GPREG29		Section 30.5.2.30
78h	PRU_ICSS_DBG_GPREG30		Section 30.5.2.31
7Ch	PRU_ICSS_DBG_GPREG31		Section 30.5.2.32
80h	PRU_ICSS_DBG_CT_REG0		Section 30.5.2.33
84h	PRU_ICSS_DBG_CT_REG1		Section 30.5.2.34
88h	PRU_ICSS_DBG_CT_REG2		Section 30.5.2.35
8Ch	PRU_ICSS_DBG_CT_REG3		Section 30.5.2.36
90h	PRU_ICSS_DBG_CT_REG4		Section 30.5.2.37
94h	PRU_ICSS_DBG_CT_REG5		Section 30.5.2.38
98h	PRU_ICSS_DBG_CT_REG6		Section 30.5.2.39
9Ch	PRU_ICSS_DBG_CT_REG7		Section 30.5.2.40
A0h	PRU_ICSS_DBG_CT_REG8		Section 30.5.2.41
A4h	PRU_ICSS_DBG_CT_REG9		Section 30.5.2.42
A8h	PRU_ICSS_DBG_CT_REG10		Section 30.5.2.43

Table 30-60. PRU_ICSS_PRU_DEBUG Registers (continued)

Offset	Acronym	Register Name	Section
ACh	PRU_ICSS_DBG_CT_REG11		Section 30.5.2.44
B0h	PRU_ICSS_DBG_CT_REG12		Section 30.5.2.45
B4h	PRU_ICSS_DBG_CT_REG13		Section 30.5.2.46
B8h	PRU_ICSS_DBG_CT_REG14		Section 30.5.2.47
BCh	PRU_ICSS_DBG_CT_REG15		Section 30.5.2.48
C0h	PRU_ICSS_DBG_CT_REG16		Section 30.5.2.49
C4h	PRU_ICSS_DBG_CT_REG17		Section 30.5.2.50
C8h	PRU_ICSS_DBG_CT_REG18		Section 30.5.2.51
CCh	PRU_ICSS_DBG_CT_REG19		Section 30.5.2.52
D0h	PRU_ICSS_DBG_CT_REG20		Section 30.5.2.53
D4h	PRU_ICSS_DBG_CT_REG21		Section 30.5.2.54
D8h	PRU_ICSS_DBG_CT_REG22		Section 30.5.2.55
DCh	PRU_ICSS_DBG_CT_REG23		Section 30.5.2.56
E0h	PRU_ICSS_DBG_CT_REG24		Section 30.5.2.57
E4h	PRU_ICSS_DBG_CT_REG25		Section 30.5.2.58
E8h	PRU_ICSS_DBG_CT_REG26		Section 30.5.2.59
ECh	PRU_ICSS_DBG_CT_REG27		Section 30.5.2.60
F0h	PRU_ICSS_DBG_CT_REG28		Section 30.5.2.61
F4h	PRU_ICSS_DBG_CT_REG29		Section 30.5.2.62
F8h	PRU_ICSS_DBG_CT_REG30		Section 30.5.2.63
FCh	PRU_ICSS_DBG_CT_REG31		Section 30.5.2.64

30.5.2.1 PRU_ICSS_DBG_GPREG0 Register (offset = 0h) [reset = 0h]

PRU_ICSS_DBG_GPREG0 is shown in Figure 30-72 and described in Table 30-61.

DEBUG PRU GENERAL PURPOSE REGISTER 0. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-72. PRU_ICSS_DBG_GPREG0 Register

31–0
GP_REG0
R/W-0h

Table 30-61. PRU_ICSS_DBG_GPREG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG0	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.2 PRU_ICSS_DBG_GPREG1 Register (offset = 4h) [reset = 0h]

PRU_ICSS_DBG_GPREG1 is shown in Figure 30-73 and described in Table 30-62.

DEBUG PRU GENERAL PURPOSE REGISTER 1. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-73. PRU_ICSS_DBG_GPREG1 Register

31-0
GP_REG1
R/W-0h

Table 30-62. PRU_ICSS_DBG_GPREG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG1	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.3 PRU_ICSS_DBG_GPREG2 Register (offset = 8h) [reset = 0h]

PRU_ICSS_DBG_GPREG2 is shown in Figure 30-74 and described in Table 30-63.

DEBUG PRU GENERAL PURPOSE REGISTER 2. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-74. PRU_ICSS_DBG_GPREG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG2																															
R/W-0h																															

Table 30-63. PRU_ICSS_DBG_GPREG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG2	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.4 PRU_ICSS_DBG_GPREG3 Register (offset = Ch) [reset = 0h]

PRU_ICSS_DBG_GPREG3 is shown in Figure 30-75 and described in Table 30-64.

DEBUG PRU GENERAL PURPOSE REGISTER 3. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-75. PRU_ICSS_DBG_GPREG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG3																															
R/W-0h																															

Table 30-64. PRU_ICSS_DBG_GPREG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG3	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.5 PRU_ICSS_DBG_GPREG4 Register (offset = 10h) [reset = 0h]

PRU_ICSS_DBG_GPREG4 is shown in Figure 30-76 and described in Table 30-65.

DEBUG PRU GENERAL PURPOSE REGISTER 4. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-76. PRU_ICSS_DBG_GPREG4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG4																															
R/W-0h																															

Table 30-65. PRU_ICSS_DBG_GPREG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG4	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.6 PRU_ICSS_DBG_GPREG5 Register (offset = 14h) [reset = 0h]

PRU_ICSS_DBG_GPREG5 is shown in Figure 30-77 and described in Table 30-66.

DEBUG PRU GENERAL PURPOSE REGISTER 5. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-77. PRU_ICSS_DBG_GPREG5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG5																															
R/W-0h																															

Table 30-66. PRU_ICSS_DBG_GPREG5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG5	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.7 PRU_ICSS_DBG_GPREG6 Register (offset = 18h) [reset = 0h]

PRU_ICSS_DBG_GPREG6 is shown in Figure 30-78 and described in Table 30-67.

DEBUG PRU GENERAL PURPOSE REGISTER 6. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-78. PRU_ICSS_DBG_GPREG6 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">GP_REG6</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 30-67. PRU_ICSS_DBG_GPREG6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG6	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.8 PRU_ICSS_DBG_GPREG7 Register (offset = 1Ch) [reset = 0h]

PRU_ICSS_DBG_GPREG7 is shown in Figure 30-79 and described in Table 30-68.

DEBUG PRU GENERAL PURPOSE REGISTER 7. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-79. PRU_ICSS_DBG_GPREG7 Register

31–0
GP_REG7
R/W-0h

Table 30-68. PRU_ICSS_DBG_GPREG7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG7	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.9 PRU_ICSS_DBG_GPREG8 Register (offset = 20h) [reset = 0h]

PRU_ICSS_DBG_GPREG8 is shown in Figure 30-80 and described in Table 30-69.

DEBUG PRU GENERAL PURPOSE REGISTER 8. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-80. PRU_ICSS_DBG_GPREG8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG8																															
R/W-0h																															

Table 30-69. PRU_ICSS_DBG_GPREG8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG8	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.10 PRU_ICSS_DBG_GPREG9 Register (offset = 24h) [reset = 0h]

PRU_ICSS_DBG_GPREG9 is shown in Figure 30-81 and described in Table 30-70.

DEBUG PRU GENERAL PURPOSE REGISTER 9. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-81. PRU_ICSS_DBG_GPREG9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG9																															
R/W-0h																															

Table 30-70. PRU_ICSS_DBG_GPREG9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG9	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.11 PRU_ICSS_DBG_GPREG10 Register (offset = 28h) [reset = 0h]

PRU_ICSS_DBG_GPREG10 is shown in Figure 30-82 and described in Table 30-71.

DEBUG PRU GENERAL PURPOSE REGISTER 10. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-82. PRU_ICSS_DBG_GPREG10 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG10																															
R/W-0h																															

Table 30-71. PRU_ICSS_DBG_GPREG10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG10	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.12 PRU_ICSS_DBG_GPREG11 Register (offset = 2Ch) [reset = 0h]

PRU_ICSS_DBG_GPREG11 is shown in Figure 30-83 and described in Table 30-72.

DEBUG PRU GENERAL PURPOSE REGISTER 11. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-83. PRU_ICSS_DBG_GPREG11 Register

31–0
GP_REG11
R/W-0h

Table 30-72. PRU_ICSS_DBG_GPREG11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG11	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.13 PRU_ICSS_DBG_GPREG12 Register (offset = 30h) [reset = 0h]

PRU_ICSS_DBG_GPREG12 is shown in Figure 30-84 and described in Table 30-73.

DEBUG PRU GENERAL PURPOSE REGISTER 12. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-84. PRU_ICSS_DBG_GPREG12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG12																															
R/W-0h																															

Table 30-73. PRU_ICSS_DBG_GPREG12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG12	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.14 PRU_ICSS_DBG_GPREG13 Register (offset = 34h) [reset = 0h]

PRU_ICSS_DBG_GPREG13 is shown in Figure 30-85 and described in Table 30-74.

DEBUG PRU GENERAL PURPOSE REGISTER 13. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-85. PRU_ICSS_DBG_GPREG13 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">GP_REG13</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 30-74. PRU_ICSS_DBG_GPREG13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG13	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.15 PRU_ICSS_DBG_GPREG14 Register (offset = 38h) [reset = 0h]

PRU_ICSS_DBG_GPREG14 is shown in Figure 30-86 and described in Table 30-75.

DEBUG PRU GENERAL PURPOSE REGISTER 14. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-86. PRU_ICSS_DBG_GPREG14 Register

31-0
GP_REG14
R/W-0h

Table 30-75. PRU_ICSS_DBG_GPREG14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG14	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.16 PRU_ICSS_DBG_GPREG15 Register (offset = 3Ch) [reset = 0h]

PRU_ICSS_DBG_GPREG15 is shown in Figure 30-87 and described in Table 30-76.

DEBUG PRU GENERAL PURPOSE REGISTER 15. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-87. PRU_ICSS_DBG_GPREG15 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG15																															
R/W-0h																															

Table 30-76. PRU_ICSS_DBG_GPREG15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG15	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.17 PRU_ICSS_DBG_GPREG16 Register (offset = 40h) [reset = 0h]

PRU_ICSS_DBG_GPREG16 is shown in Figure 30-88 and described in Table 30-77.

DEBUG PRU GENERAL PURPOSE REGISTER 16. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-88. PRU_ICSS_DBG_GPREG16 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG16																															
R/W-0h																															

Table 30-77. PRU_ICSS_DBG_GPREG16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG16	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.18 PRU_ICSS_DBG_GPREG17 Register (offset = 44h) [reset = 0h]

PRU_ICSS_DBG_GPREG17 is shown in Figure 30-89 and described in Table 30-78.

DEBUG PRU GENERAL PURPOSE REGISTER 17. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-89. PRU_ICSS_DBG_GPREG17 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG17																															
R/W-0h																															

Table 30-78. PRU_ICSS_DBG_GPREG17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG17	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.19 PRU_ICSS_DBG_GPREG18 Register (offset = 48h) [reset = 0h]

PRU_ICSS_DBG_GPREG18 is shown in Figure 30-90 and described in Table 30-79.

DEBUG PRU GENERAL PURPOSE REGISTER 18. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-90. PRU_ICSS_DBG_GPREG18 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">GP_REG18</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 30-79. PRU_ICSS_DBG_GPREG18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG18	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.20 PRU_ICSS_DBG_GPREG19 Register (offset = 4Ch) [reset = 0h]

PRU_ICSS_DBG_GPREG19 is shown in Figure 30-91 and described in Table 30-80.

DEBUG PRU GENERAL PURPOSE REGISTER 19. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-91. PRU_ICSS_DBG_GPREG19 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG19																															
R/W-0h																															

Table 30-80. PRU_ICSS_DBG_GPREG19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG19	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.21 PRU_ICSS_DBG_GPREG20 Register (offset = 50h) [reset = 0h]

PRU_ICSS_DBG_GPREG20 is shown in Figure 30-92 and described in Table 30-81.

DEBUG PRU GENERAL PURPOSE REGISTER 20. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-92. PRU_ICSS_DBG_GPREG20 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG20																															
R/W-0h																															

Table 30-81. PRU_ICSS_DBG_GPREG20 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG20	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.22 PRU_ICSS_DBG_GPREG21 Register (offset = 54h) [reset = 0h]

PRU_ICSS_DBG_GPREG21 is shown in Figure 30-93 and described in Table 30-82.

DEBUG PRU GENERAL PURPOSE REGISTER 21. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-93. PRU_ICSS_DBG_GPREG21 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG21																															
R/W-0h																															

Table 30-82. PRU_ICSS_DBG_GPREG21 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG21	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.23 PRU_ICSS_DBG_GPREG22 Register (offset = 58h) [reset = 0h]

PRU_ICSS_DBG_GPREG22 is shown in Figure 30-94 and described in Table 30-83.

DEBUG PRU GENERAL PURPOSE REGISTER 22. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-94. PRU_ICSS_DBG_GPREG22 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">GP_REG22</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 30-83. PRU_ICSS_DBG_GPREG22 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG22	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.24 PRU_ICSS_DBG_GPREG23 Register (offset = 5Ch) [reset = 0h]

PRU_ICSS_DBG_GPREG23 is shown in Figure 30-95 and described in Table 30-84.

DEBUG PRU GENERAL PURPOSE REGISTER 23. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-95. PRU_ICSS_DBG_GPREG23 Register

31–0
GP_REG23
R/W-0h

Table 30-84. PRU_ICSS_DBG_GPREG23 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG23	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.25 PRU_ICSS_DBG_GPREG24 Register (offset = 60h) [reset = 0h]

PRU_ICSS_DBG_GPREG24 is shown in Figure 30-96 and described in Table 30-85.

DEBUG PRU GENERAL PURPOSE REGISTER 24. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-96. PRU_ICSS_DBG_GPREG24 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG24																															
R/W-0h																															

Table 30-85. PRU_ICSS_DBG_GPREG24 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG24	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.26 PRU_ICSS_DBG_GPREG25 Register (offset = 64h) [reset = 0h]

PRU_ICSS_DBG_GPREG25 is shown in Figure 30-97 and described in Table 30-86.

DEBUG PRU GENERAL PURPOSE REGISTER 25. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-97. PRU_ICSS_DBG_GPREG25 Register

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="32">GP_REG25</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 30-86. PRU_ICSS_DBG_GPREG25 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG25	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.27 PRU_ICSS_DBG_GPREG26 Register (offset = 68h) [reset = 0h]

PRU_ICSS_DBG_GPREG26 is shown in Figure 30-98 and described in Table 30-87.

DEBUG PRU GENERAL PURPOSE REGISTER 26. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-98. PRU_ICSS_DBG_GPREG26 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG26																															
R/W-0h																															

Table 30-87. PRU_ICSS_DBG_GPREG26 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG26	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.28 PRU_ICSS_DBG_GPREG27 Register (offset = 6Ch) [reset = 0h]

PRU_ICSS_DBG_GPREG27 is shown in Figure 30-99 and described in Table 30-88.

DEBUG PRU GENERAL PURPOSE REGISTER 27. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-99. PRU_ICSS_DBG_GPREG27 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG27																															
R/W-0h																															

Table 30-88. PRU_ICSS_DBG_GPREG27 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG27	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.29 PRU_ICSS_DBG_GPREG28 Register (offset = 70h) [reset = 0h]

PRU_ICSS_DBG_GPREG28 is shown in Figure 30-100 and described in Table 30-89.

DEBUG PRU GENERAL PURPOSE REGISTER 28. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-100. PRU_ICSS_DBG_GPREG28 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG28																															
R/W-0h																															

Table 30-89. PRU_ICSS_DBG_GPREG28 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG28	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.30 PRU_ICSS_DBG_GPREG29 Register (offset = 74h) [reset = 0h]

PRU_ICSS_DBG_GPREG29 is shown in Figure 30-101 and described in Table 30-90.

DEBUG PRU GENERAL PURPOSE REGISTER 29. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-101. PRU_ICSS_DBG_GPREG29 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG29																															
R/W-0h																															

Table 30-90. PRU_ICSS_DBG_GPREG29 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG29	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.31 PRU_ICSS_DBG_GPREG30 Register (offset = 78h) [reset = 0h]

PRU_ICSS_DBG_GPREG30 is shown in Figure 30-102 and described in Table 30-91.

DEBUG PRU GENERAL PURPOSE REGISTER 30. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-102. PRU_ICSS_DBG_GPREG30 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP_REG30																															
R/W-0h																															

Table 30-91. PRU_ICSS_DBG_GPREG30 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP_REG30	R/W	0h	PRU Internal GP Register n: Reading / writing this field directly inspects/modifies the corresponding internal register in the PRU internal regfile.

30.5.2.32 PRU_ICSS_DBG_GPREG31 Register (offset = 7Ch) [reset = 0h]

PRU_ICSS_DBG_GPREG31 is shown in Figure 30-103 and described in Table 30-92.

DEBUG PRU GENERAL PURPOSE REGISTER 31. This register allows an external agent to debug the PRU while it is disabled. Reading or writing to these registers will have the same effect as a read or write to these registers from an internal instruction in the PRU. For R30, this includes generation of the pulse outputs whenever the register is written.

Figure 30-103. PRU_ICSS_DBG_GPREG31 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">GPREG31</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 30-92. PRU_ICSS_DBG_GPREG31 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPREG31	R/W	0h	

30.5.2.33 PRU_ICSS_DBG_CT_REG0 Register (offset = 80h) [reset = 20000h]

PRU_ICSS_DBG_CT_REG0 is shown in Figure 30-104 and described in Table 30-93.

DEBUG PRU CONSTANTS TABLE ENTRY 0. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-104. PRU_ICSS_DBG_CT_REG0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">CT_REG0</td></tr>
<tr><td colspan="32">R-20000h</td></tr>
</table>

Table 30-93. PRU_ICSS_DBG_CT_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG0	R	20000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.34 PRU_ICSS_DBG_CT_REG1 Register (offset = 84h) [reset = 48040000h]

PRU_ICSS_DBG_CT_REG1 is shown in Figure 30-105 and described in Table 30-94.

DEBUG PRU CONSTANTS TABLE ENTRY 1. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-105. PRU_ICSS_DBG_CT_REG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT_REG1																															
R-48040000h																															

Table 30-94. PRU_ICSS_DBG_CT_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG1	R	48040000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.35 PRU_ICSS_DBG_CT_REG2 Register (offset = 88h) [reset = 4802A000h]

PRU_ICSS_DBG_CT_REG2 is shown in Figure 30-106 and described in Table 30-95.

DEBUG PRU CONSTANTS TABLE ENTRY 2. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-106. PRU_ICSS_DBG_CT_REG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT_REG2																															
R-4802A000h																															

Table 30-95. PRU_ICSS_DBG_CT_REG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG2	R	4802A000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.36 PRU_ICSS_DBG_CT_REG3 Register (offset = 8Ch) [reset = 30000h]

PRU_ICSS_DBG_CT_REG3 is shown in Figure 30-107 and described in Table 30-96.

DEBUG PRU CONSTANTS TABLE ENTRY 3. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-107. PRU_ICSS_DBG_CT_REG3 Register

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="32">CT_REG3</td></tr>
<tr><td colspan="32">R-30000h</td></tr>
</table>

Table 30-96. PRU_ICSS_DBG_CT_REG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG3	R	30000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.37 PRU_ICSS_DBG_CT_REG4 Register (offset = 90h) [reset = 26000h]

PRU_ICSS_DBG_CT_REG4 is shown in Figure 30-108 and described in Table 30-97.

DEBUG PRU CONSTANTS TABLE ENTRY 4. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-108. PRU_ICSS_DBG_CT_REG4 Register

31–0
CT_REG4
R-26000h

Table 30-97. PRU_ICSS_DBG_CT_REG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG4	R	26000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.38 PRU_ICSS_DBG_CT_REG5 Register (offset = 94h) [reset = 48060000h]

PRU_ICSS_DBG_CT_REG5 is shown in Figure 30-109 and described in Table 30-98.

DEBUG PRU CONSTANTS TABLE ENTRY 5. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-109. PRU_ICSS_DBG_CT_REG5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT_REG5																															
R-48060000h																															

Table 30-98. PRU_ICSS_DBG_CT_REG5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG5	R	48060000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.39 PRU_ICSS_DBG_CT_REG6 Register (offset = 98h) [reset = 48030000h]

PRU_ICSS_DBG_CT_REG6 is shown in Figure 30-110 and described in Table 30-99.

DEBUG PRU CONSTANTS TABLE ENTRY 6. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-110. PRU_ICSS_DBG_CT_REG6 Register

31-0
CT_REG6
R-48030000h

Table 30-99. PRU_ICSS_DBG_CT_REG6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG6	R	48030000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.40 PRU_ICSS_DBG_CT_REG7 Register (offset = 9Ch) [reset = 28000h]

PRU_ICSS_DBG_CT_REG7 is shown in Figure 30-111 and described in Table 30-100.

DEBUG PRU CONSTANTS TABLE ENTRY 7. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-111. PRU_ICSS_DBG_CT_REG7 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">CT_REG7</td></tr>
<tr><td colspan="32">R-28000h</td></tr>
</table>

Table 30-100. PRU_ICSS_DBG_CT_REG7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG7	R	28000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.41 PRU_ICSS_DBG_CT_REG8 Register (offset = A0h) [reset = 46000000h]

PRU_ICSS_DBG_CT_REG8 is shown in Figure 30-112 and described in Table 30-101.

DEBUG PRU CONSTANTS TABLE ENTRY 8. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-112. PRU_ICSS_DBG_CT_REG8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT_REG8																															
R-46000000h																															

Table 30-101. PRU_ICSS_DBG_CT_REG8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG8	R	46000000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.42 PRU_ICSS_DBG_CT_REG9 Register (offset = A4h) [reset = 4A100000h]

PRU_ICSS_DBG_CT_REG9 is shown in Figure 30-113 and described in Table 30-102.

DEBUG PRU CONSTANTS TABLE ENTRY 9. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-113. PRU_ICSS_DBG_CT_REG9 Register

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="32">CT_REG9</td></tr>
<tr><td colspan="32">R-4A100000h</td></tr>
</table>

Table 30-102. PRU_ICSS_DBG_CT_REG9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG9	R	4A100000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.43 PRU_ICSS_DBG_CT_REG10 Register (offset = A8h) [reset = 48318000h]

PRU_ICSS_DBG_CT_REG10 is shown in Figure 30-114 and described in Table 30-103.

DEBUG PRU CONSTANTS TABLE ENTRY 10. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-114. PRU_ICSS_DBG_CT_REG10 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">CT_REG10</td></tr>
<tr><td colspan="32">R-48318000h</td></tr>
</table>

Table 30-103. PRU_ICSS_DBG_CT_REG10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG10	R	48318000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.44 PRU_ICSS_DBG_CT_REG11 Register (offset = ACh) [reset = 48022000h]

PRU_ICSS_DBG_CT_REG11 is shown in Figure 30-115 and described in Table 30-104.

DEBUG PRU CONSTANTS TABLE ENTRY 11. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-115. PRU_ICSS_DBG_CT_REG11 Register

31–0
CT_REG11
R-48022000h

Table 30-104. PRU_ICSS_DBG_CT_REG11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG11	R	48022000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.45 PRU_ICSS_DBG_CT_REG12 Register (offset = B0h) [reset = 48024000h]

PRU_ICSS_DBG_CT_REG12 is shown in Figure 30-116 and described in Table 30-105.

DEBUG PRU CONSTANTS TABLE ENTRY 12. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-116. PRU_ICSS_DBG_CT_REG12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT_REG12																															
R-48024000h																															

Table 30-105. PRU_ICSS_DBG_CT_REG12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG12	R	48024000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.46 PRU_ICSS_DBG_CT_REG13 Register (offset = B4h) [reset = 48310000h]

PRU_ICSS_DBG_CT_REG13 is shown in Figure 30-117 and described in Table 30-106.

DEBUG PRU CONSTANTS TABLE ENTRY 13. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-117. PRU_ICSS_DBG_CT_REG13 Register

31–0
CT_REG13
R-48310000h

Table 30-106. PRU_ICSS_DBG_CT_REG13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG13	R	48310000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.47 PRU_ICSS_DBG_CT_REG14 Register (offset = B8h) [reset = 481CC000h]

PRU_ICSS_DBG_CT_REG14 is shown in Figure 30-118 and described in Table 30-107.

DEBUG PRU CONSTANTS TABLE ENTRY 14. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-118. PRU_ICSS_DBG_CT_REG14 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">CT_REG14</td></tr>
<tr><td colspan="32">R-481CC000h</td></tr>
</table>

Table 30-107. PRU_ICSS_DBG_CT_REG14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG14	R	481CC000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.48 PRU_ICSS_DBG_CT_REG15 Register (offset = BCh) [reset = 481D0000h]

PRU_ICSS_DBG_CT_REG15 is shown in Figure 30-119 and described in Table 30-108.

DEBUG PRU CONSTANTS TABLE ENTRY 15. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-119. PRU_ICSS_DBG_CT_REG15 Register

31-0
CT_REG15
R-481D0000h

Table 30-108. PRU_ICSS_DBG_CT_REG15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG15	R	481D0000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.49 PRU_ICSS_DBG_CT_REG16 Register (offset = C0h) [reset = 481A0000h]

PRU_ICSS_DBG_CT_REG16 is shown in Figure 30-120 and described in Table 30-109.

DEBUG PRU CONSTANTS TABLE ENTRY 16. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-120. PRU_ICSS_DBG_CT_REG16 Register

31-0
CT_REG16
R-481A0000h

Table 30-109. PRU_ICSS_DBG_CT_REG16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG16	R	481A0000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.50 PRU_ICSS_DBG_CT_REG17 Register (offset = C4h) [reset = 4819C000h]

PRU_ICSS_DBG_CT_REG17 is shown in Figure 30-121 and described in Table 30-110.

DEBUG PRU CONSTANTS TABLE ENTRY 17. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-121. PRU_ICSS_DBG_CT_REG17 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT_REG17																															
R-4819C000h																															

Table 30-110. PRU_ICSS_DBG_CT_REG17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG17	R	4819C000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.51 PRU_ICSS_DBG_CT_REG18 Register (offset = C8h) [reset = 48300000h]

PRU_ICSS_DBG_CT_REG18 is shown in Figure 30-122 and described in Table 30-111.

DEBUG PRU CONSTANTS TABLE ENTRY 18. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-122. PRU_ICSS_DBG_CT_REG18 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">CT_REG18</td></tr>
<tr><td colspan="32">R-48300000h</td></tr>
</table>

Table 30-111. PRU_ICSS_DBG_CT_REG18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG18	R	48300000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.52 PRU_ICSS_DBG_CT_REG19 Register (offset = CCh) [reset = 48302000h]

PRU_ICSS_DBG_CT_REG19 is shown in Figure 30-123 and described in Table 30-112.

DEBUG PRU CONSTANTS TABLE ENTRY 19. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-123. PRU_ICSS_DBG_CT_REG19 Register

31-0
CT_REG19
R-48302000h

Table 30-112. PRU_ICSS_DBG_CT_REG19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG19	R	48302000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.53 PRU_ICSS_DBG_CT_REG20 Register (offset = D0h) [reset = 48304000h]

PRU_ICSS_DBG_CT_REG20 is shown in Figure 30-124 and described in Table 30-113.

DEBUG PRU CONSTANTS TABLE ENTRY 20. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-124. PRU_ICSS_DBG_CT_REG20 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT_REG20																															
R-48304000h																															

Table 30-113. PRU_ICSS_DBG_CT_REG20 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG20	R	48304000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.54 PRU_ICSS_DBG_CT_REG21 Register (offset = D4h) [reset = 32400h]

PRU_ICSS_DBG_CT_REG21 is shown in Figure 30-125 and described in Table 30-114.

DEBUG PRU CONSTANTS TABLE ENTRY 21. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-125. PRU_ICSS_DBG_CT_REG21 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">CT_REG21</td></tr>
<tr><td colspan="32">R-32400h</td></tr>
</table>

Table 30-114. PRU_ICSS_DBG_CT_REG21 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG21	R	32400h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.55 PRU_ICSS_DBG_CT_REG22 Register (offset = D8h) [reset = 480C8000h]

PRU_ICSS_DBG_CT_REG22 is shown in Figure 30-126 and described in Table 30-115.

DEBUG PRU CONSTANTS TABLE ENTRY 22. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-126. PRU_ICSS_DBG_CT_REG22 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">CT_REG22</td></tr>
<tr><td colspan="32">R-480C8000h</td></tr>
</table>

Table 30-115. PRU_ICSS_DBG_CT_REG22 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG22	R	480C8000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.56 PRU_ICSS_DBG_CT_REG23 Register (offset = DCh) [reset = 480CA000h]

PRU_ICSS_DBG_CT_REG23 is shown in Figure 30-127 and described in Table 30-116.

DEBUG PRU CONSTANTS TABLE ENTRY 23. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-127. PRU_ICSS_DBG_CT_REG23 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">CT_REG23</td></tr>
<tr><td colspan="32">R-480CA000h</td></tr>
</table>

Table 30-116. PRU_ICSS_DBG_CT_REG23 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG23	R	480CA000h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table.

30.5.2.57 PRU_ICSS_DBG_CT_REG24 Register (offset = E0h) [reset = 0h]

PRU_ICSS_DBG_CT_REG24 is shown in Figure 30-128 and described in Table 30-117.

DEBUG PRU CONSTANTS TABLE ENTRY 24. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-128. PRU_ICSS_DBG_CT_REG24 Register

31–0
CT_REG24
R-0h

Table 30-117. PRU_ICSS_DBG_CT_REG24 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG24	R	0h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table. This entry is partially programmable through the C24_BLK_INDEX in the PRU Control register. The reset value for this Constant Table Entry is 0x00000n00, n=C24_BLK_INDEX[3:0].

30.5.2.58 PRU_ICSS_DBG_CT_REG25 Register (offset = E4h) [reset = 0h]

PRU_ICSS_DBG_CT_REG25 is shown in Figure 30-129 and described in Table 30-118.

DEBUG PRU CONSTANTS TABLE ENTRY 25. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-129. PRU_ICSS_DBG_CT_REG25 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">CT_REG25</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 30-118. PRU_ICSS_DBG_CT_REG25 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG25	R	0h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table. This entry is partially programmable through the C25_BLK_INDEX in the PRU Control register. The reset value for this Constant Table Entry is 0x00002n00, n=C25_BLK_INDEX[3:0].

30.5.2.59 PRU_ICSS_DBG_CT_REG26 Register (offset = E8h) [reset = 0h]

PRU_ICSS_DBG_CT_REG26 is shown in Figure 30-130 and described in Table 30-119.

DEBUG PRU CONSTANTS TABLE ENTRY 26. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-130. PRU_ICSS_DBG_CT_REG26 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT_REG26																															
R-0h																															

Table 30-119. PRU_ICSS_DBG_CT_REG26 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG26	R	0h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table. This entry is partially programmable through the C26_BLK_INDEX in the PRU Control register. The reset value for this Constant Table Entry is 0x0002En00, n=C26_BLK_INDEX[3:0].

30.5.2.60 PRU_ICSS_DBG_CT_REG27 Register (offset = ECh) [reset = 0h]

PRU_ICSS_DBG_CT_REG27 is shown in Figure 30-131 and described in Table 30-120.

DEBUG PRU CONSTANTS TABLE ENTRY 27. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-131. PRU_ICSS_DBG_CT_REG27 Register

31-0
CT_REG27
R-0h

Table 30-120. PRU_ICSS_DBG_CT_REG27 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG27	R	0h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table. This entry is partially programmable through the C27_BLK_INDEX in the PRU Control register. The reset value for this Constant Table Entry is 0x00032n00, n=C27_BLK_INDEX[3:0].

30.5.2.61 PRU_ICSS_DBG_CT_REG28 Register (offset = F0h) [reset = 0h]

PRU_ICSS_DBG_CT_REG28 is shown in Figure 30-132 and described in Table 30-121.

DEBUG PRU CONSTANTS TABLE ENTRY 28. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-132. PRU_ICSS_DBG_CT_REG28 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">CT_REG28</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 30-121. PRU_ICSS_DBG_CT_REG28 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG28	R	0h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table. This entry is partially programmable through the C28_POINTER in the PRU Control register. The reset value for this Constant Table Entry is 0x00nnnn00, nnnn=C28_POINTER[15:0].

30.5.2.62 PRU_ICSS_DBG_CT_REG29 Register (offset = F4h) [reset = 0h]

PRU_ICSS_DBG_CT_REG29 is shown in Figure 30-133 and described in Table 30-122.

DEBUG PRU CONSTANTS TABLE ENTRY 29. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-133. PRU_ICSS_DBG_CT_REG29 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT_REG29																															
R-0h																															

Table 30-122. PRU_ICSS_DBG_CT_REG29 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG29	R	0h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table. This entry is partially programmable through the C29_POINTER in the PRU Control register. The reset value for this Constant Table Entry is 0x49nnnn00, nnnn=C29_POINTER[15:0].

30.5.2.63 PRU_ICSS_DBG_CT_REG30 Register (offset = F8h) [reset = 0h]

PRU_ICSS_DBG_CT_REG30 is shown in Figure 30-134 and described in Table 30-123.

DEBUG PRU CONSTANTS TABLE ENTRY 30. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-134. PRU_ICSS_DBG_CT_REG30 Register

31-0
CT_REG30
R-0h

Table 30-123. PRU_ICSS_DBG_CT_REG30 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG30	R	0h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table. This entry is partially programmable through the C30_POINTER in the PRU Control register. The reset value for this Constant Table Entry is 0x40nnnn00, nnnn=C30_POINTER[15:0].

30.5.2.64 PRU_ICSS_DBG_CT_REG31 Register (offset = FCh) [reset = 0h]

PRU_ICSS_DBG_CT_REG31 is shown in Figure 30-135 and described in Table 30-124.

DEBUG PRU CONSTANTS TABLE ENTRY 31. This register allows an external agent to debug the PRU while it is disabled. Since some of the constants table entries may actually depend on system inputs / and or the internal state of the PRU, these registers are provided to allow an external agent to easily determine the state of the constants table.

Figure 30-135. PRU_ICSS_DBG_CT_REG31 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT_REG31																															
R-0h																															

Table 30-124. PRU_ICSS_DBG_CT_REG31 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT_REG31	R	0h	PRU Internal Constants Table Entry n: Reading this field directly inspects the corresponding entry in the PRU internal constants table. This entry is partially programmable through the C31_POINTER in the PRU Control register. The reset value for this Constant Table Entry is 0x80nnnn00, nnnn=C31_POINTER[15:0].

30.5.3 PRU_ICSS_INTC Registers

Table 30-125 lists the memory-mapped registers for the PRU_ICSS_INTC. All register offset addresses not listed in Table 30-125 should be considered as reserved locations and the register contents should not be modified.

Table 30-125. PRU_ICSS_INTC Registers

Offset	Acronym	Register Name	Section
0h	PRU_ICSS_INTC_REVID		Section 30.5.3.1
4h	PRU_ICSS_INTC_CR		Section 30.5.3.2
10h	PRU_ICSS_INTC_GER		Section 30.5.3.3
1Ch	PRU_ICSS_INTC_GNLR		Section 30.5.3.4
20h	PRU_ICSS_INTC_SISR		Section 30.5.3.5
24h	PRU_ICSS_INTC_SICR		Section 30.5.3.6
28h	PRU_ICSS_INTC_EISR		Section 30.5.3.7
2Ch	PRU_ICSS_INTC_EICR		Section 30.5.3.8
34h	PRU_ICSS_INTC_HIEISR		Section 30.5.3.9
38h	PRU_ICSS_INTC_HIDISR		Section 30.5.3.10
80h	PRU_ICSS_INTC_GPIR		Section 30.5.3.11
200h	PRU_ICSS_INTC_SRSR0		Section 30.5.3.12
204h	PRU_ICSS_INTC_SRSR1		Section 30.5.3.13
280h	PRU_ICSS_INTC_SECR0		Section 30.5.3.14
284h	PRU_ICSS_INTC_SECR1		Section 30.5.3.15
300h	PRU_ICSS_INTC_ESR0		Section 30.5.3.16
304h	PRU_ICSS_INTC_ERS1		Section 30.5.3.17
380h	PRU_ICSS_INTC_ECR0		Section 30.5.3.18
384h	PRU_ICSS_INTC_ECR1		Section 30.5.3.19
400h	PRU_ICSS_INTC_CMR0		Section 30.5.3.20
404h	PRU_ICSS_INTC_CMR1		Section 30.5.3.21
408h	PRU_ICSS_INTC_CMR2		Section 30.5.3.22
40Ch	PRU_ICSS_INTC_CMR3		Section 30.5.3.23

Table 30-125. PRU_ICSS_INTC Registers (continued)

Offset	Acronym	Register Name	Section
410h	PRU_ICSS_INTC_CMR4		Section 30.5.3.24
414h	PRU_ICSS_INTC_CMR5		Section 30.5.3.25
418h	PRU_ICSS_INTC_CMR6		Section 30.5.3.26
41Ch	PRU_ICSS_INTC_CMR7		Section 30.5.3.27
420h	PRU_ICSS_INTC_CMR8		Section 30.5.3.28
424h	PRU_ICSS_INTC_CMR9		Section 30.5.3.29
428h	PRU_ICSS_INTC_CMR10		Section 30.5.3.30
42Ch	PRU_ICSS_INTC_CMR11		Section 30.5.3.31
430h	PRU_ICSS_INTC_CMR12		Section 30.5.3.32
434h	PRU_ICSS_INTC_CMR13		Section 30.5.3.33
438h	PRU_ICSS_INTC_CMR14		Section 30.5.3.34
43Ch	PRU_ICSS_INTC_CMR15		Section 30.5.3.35
800h	PRU_ICSS_INTC_HMR0		Section 30.5.3.36
804h	PRU_ICSS_INTC_HMR1		Section 30.5.3.37
808h	PRU_ICSS_INTC_HMR2		Section 30.5.3.38
900h	PRU_ICSS_INTC_HIPIR0		Section 30.5.3.39
904h	PRU_ICSS_INTC_HIPIR1		Section 30.5.3.40
908h	PRU_ICSS_INTC_HIPIR2		Section 30.5.3.41
90Ch	PRU_ICSS_INTC_HIPIR3		Section 30.5.3.42
910h	PRU_ICSS_INTC_HIPIR4		Section 30.5.3.43
914h	PRU_ICSS_INTC_HIPIR5		Section 30.5.3.44
918h	PRU_ICSS_INTC_HIPIR6		Section 30.5.3.45
91Ch	PRU_ICSS_INTC_HIPIR7		Section 30.5.3.46
920h	PRU_ICSS_INTC_HIPIR8		Section 30.5.3.47
924h	PRU_ICSS_INTC_HIPIR9		Section 30.5.3.48
D00h	PRU_ICSS_INTC_SIPR0		Section 30.5.3.49
D04h	PRU_ICSS_INTC_SIPR1		Section 30.5.3.50
D80h	PRU_ICSS_INTC_SITR0		Section 30.5.3.51
D84h	PRU_ICSS_INTC_SITR1		Section 30.5.3.52
1100h	PRU_ICSS_INTC_HINLR0		Section 30.5.3.53
1104h	PRU_ICSS_INTC_HINLR1		Section 30.5.3.54
1108h	PRU_ICSS_INTC_HINLR2		Section 30.5.3.55
110Ch	PRU_ICSS_INTC_HINLR3		Section 30.5.3.56
1110h	PRU_ICSS_INTC_HINLR4		Section 30.5.3.57
1114h	PRU_ICSS_INTC_HINLR5		Section 30.5.3.58
1118h	PRU_ICSS_INTC_HINLR6		Section 30.5.3.59
111Ch	PRU_ICSS_INTC_HINLR7		Section 30.5.3.60
1120h	PRU_ICSS_INTC_HINLR8		Section 30.5.3.61
1124h	PRU_ICSS_INTC_HINLR9		Section 30.5.3.62
1500h	PRU_ICSS_INTC_HIER		Section 30.5.3.63

30.5.3.1 PRU_ICSS_INTC_REVID Register (offset = 0h) [reset = 4E82A900h]

PRU_ICSS_INTC_REVID is shown in Figure 30-136 and described in Table 30-126.

Revision ID Register

Figure 30-136. PRU_ICSS_INTC_REVID Register

31	30	29	28	27	26	25	24
REV_SCHEME		RESERVED		REV_MODULE			
R-1h		R-0h		R-E82h			

23	22	21	20	19	18	17	16
REV_MODULE							
R-E82h							

15	14	13	12	11	10	9	8
REV_RTL					REV_MAJOR		
R-15h					R-1h		

7	6	5	4	3	2	1	0
REV_CUSTOM		REV_MINOR					
R-0h		R-0h					

Table 30-126. PRU_ICSS_INTC_REVID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	REV_SCHEME	R	1h	SCHEME
29-28	RESERVED	R	0h	
27-16	REV_MODULE	R	E82h	MODULE ID
15-11	REV_RTL	R	15h	RTL REVISIONS
10-8	REV_MAJOR	R	1h	MAJOR REVISION
7-6	REV_CUSTOM	R	0h	CUSTOM REVISION
5-0	REV_MINOR	R	0h	MINOR REVISION

30.5.3.2 PRU_ICSS_INTC_CR Register (offset = 4h) [reset = 0h]

PRU_ICSS_INTC_CR is shown in Figure 30-137 and described in Table 30-127.

The Control Register holds global control parameters and can forces a soft reset on the module.

Figure 30-137. PRU_ICSS_INTC_CR Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED			RESERVED	NEST_MODE		RESERVED	RESERVED
R/W-0h			R-0h	R/W-0h		R/W-0h	R/W-0h

Table 30-127. PRU_ICSS_INTC_CR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4	RESERVED	R	0h	Reserved.
3-2	NEST_MODE	R/W	0h	The nesting mode. 0 = no nesting 1 = automatic individual nesting (per host interrupt) 2 = automatic global nesting (over all host interrupts) 3 = manual nesting
1	RESERVED	R/W	0h	Reserved.
0	RESERVED	R/W	0h	

30.5.3.3 PRU_ICSS_INTC_GER Register (offset = 10h) [reset = 0h]

PRU_ICSS_INTC_GER is shown in Figure 30-138 and described in Table 30-128.

The Global Host Interrupt Enable Register enables all the host interrupts. Individual host interrupts are still enabled or disabled from their individual enables and are not overridden by the global enable.

Figure 30-138. PRU_ICSS_INTC_GER Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED							EN_HINT_ANY
R/W-0h							R/W-0h

Table 30-128. PRU_ICSS_INTC_GER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	EN_HINT_ANY	R/W	0h	The current global enable value when read. Writes set the global enable.

30.5.3.4 PRU_ICSS_INTC_GNLR Register (offset = 1Ch) [reset = 100h]

PRU_ICSS_INTC_GNLR is shown in Figure 30-139 and described in Table 30-129.

The Global Nesting Level Register allows the checking and setting of the global nesting level across all host interrupts when automatic global nesting mode is set. The nesting level is the channel (and all of lower priority) that are nested out because of a current interrupt. This register is only available when nesting is configured.

Figure 30-139. PRU_ICSS_INTC_GNLR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>AUTO_OVERRIDE</td><td colspan="7">RESERVED</td></tr>
<tr><td>W-0h</td><td colspan="7">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>GLB_NEST_LEVEL</td></tr>
<tr><td colspan="7">R/W-0h</td><td>R/W-100h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">GLB_NEST_LEVEL</td></tr>
<tr><td colspan="8">R/W-100h</td></tr>
</table>

Table 30-129. PRU_ICSS_INTC_GNLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	Always read as 0. Writes of 1 override the automatic nesting and set the nesting_level to the written data.
30-9	RESERVED	R/W	0h	
8-0	GLB_NEST_LEVEL	R/W	100h	The current global nesting level (highest channel that is nested). Writes set the nesting level. In auto nesting mode this value is updated internally unless the auto_override bit is set.

30.5.3.5 PRU_ICSS_INTC_SISR Register (offset = 20h) [reset = 0h]

PRU_ICSS_INTC_SISR is shown in Figure 30-140 and described in Table 30-130.

The System Event Status Indexed Set Register allows setting the status of an event. The event to set is the index value written (0-63). This sets the Raw Status Register bit of the given index.

Figure 30-140. PRU_ICSS_INTC_SISR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="10">STS_SET_IDX</td></tr>
<tr><td colspan="6">W-0h</td><td colspan="10">W-0h</td></tr>
</table>

Table 30-130. PRU_ICSS_INTC_SISR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	W	0h	
9-0	STS_SET_IDX	W	0h	Writes set the status of the event given in the index value (0-63). Reads return 0.

30.5.3.6 PRU_ICSS_INTC_SICR Register (offset = 24h) [reset = 0h]

PRU_ICSS_INTC_SICR is shown in Figure 30-141 and described in Table 30-131.

The System Event Status Indexed Clear Register allows clearing the status of an event. The event to clear is the index value written. This clears the Raw Status Register bit of the given index.

Figure 30-141. PRU_ICSS_INTC_SICR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
W-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						STS_CLR_IDX									
W-0h						W-0h									

Table 30-131. PRU_ICSS_INTC_SICR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	W	0h	
9-0	STS_CLR_IDX	W	0h	Writes clear the status of the event given in the index value (0-63). Reads return 0.

30.5.3.7 PRU_ICSS_INTC_EISR Register (offset = 28h) [reset = 0h]

PRU_ICSS_INTC_EISR is shown in Figure 30-142 and described in Table 30-132.

The System Event Enable Indexed Set Register allows enabling an event. The event to enable is the index value written (0-63). This sets the Enable Register bit of the given index.

Figure 30-142. PRU_ICSS_INTC_EISR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="22">RESERVED</td><td colspan="10">EN_SET_IDX</td></tr>
<tr><td colspan="22">W-0h</td><td colspan="10">W-0h</td></tr>
</table>

Table 30-132. PRU_ICSS_INTC_EISR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	W	0h	
9-0	EN_SET_IDX	W	0h	Writes set the enable of the event given in the index value (0-63). Reads return 0.

30.5.3.8 PRU_ICSS_INTC_EICR Register (offset = 2Ch) [reset = 0h]

PRU_ICSS_INTC_EICR is shown in Figure 30-143 and described in Table 30-133.

The System Event Enable Indexed Clear Register allows disabling an event. The event to disable is the index value written (0-63). This clears the Enable Register bit of the given index.

Figure 30-143. PRU_ICSS_INTC_EICR Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="22">RESERVED</td><td colspan="10">EN_CLR_IDX</td></tr>
<tr><td colspan="22">W-0h</td><td colspan="10">W-0h</td></tr>
</table>

Table 30-133. PRU_ICSS_INTC_EICR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	W	0h	
9-0	EN_CLR_IDX	W	0h	Writes clear the enable of the event given in the index value (0-63). Reads return 0.

30.5.3.9 PRU_ICSS_INTC_HIEISR Register (offset = 34h) [reset = 0h]

PRU_ICSS_INTC_HIEISR is shown in Figure 30-144 and described in Table 30-134.

The Host Interrupt Enable Indexed Set Register allows enabling a host interrupt output. The host interrupt to enable is the index value written (0-9). This enables the host interrupt output or triggers the output again if already enabled.

Figure 30-144. PRU_ICSS_INTC_HIEISR Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				HINT_EN_SET_IDX			
R/W-0h				R/W-0h			

Table 30-134. PRU_ICSS_INTC_HIEISR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	HINT_EN_SET_IDX	R/W	0h	Writes set the enable of the host interrupt given in the index value (0-9). Reads return 0.

30.5.3.10 PRU_ICSS_INTC_HIDISR Register (offset = 38h) [reset = 0h]

PRU_ICSS_INTC_HIDISR is shown in Figure 30-145 and described in Table 30-135.

The Host Interrupt Enable Indexed Clear Register allows disabling a host interrupt output. The host interrupt to disable is the index value written (0-9). This disables the host interrupt output.

Figure 30-145. PRU_ICSS_INTC_HIDISR Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				HINT_EN_CLR_IDX			
R/W-0h				R/W-0h			

Table 30-135. PRU_ICSS_INTC_HIDISR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	HINT_EN_CLR_IDX	R/W	0h	Writes clear the enable of the host interrupt given in the index value (0-9). Reads return 0.

30.5.3.11 PRU_ICSS_INTC_GPIR Register (offset = 80h) [reset = 80000000h]

PRU_ICSS_INTC_GPIR is shown in Figure 30-146 and described in Table 30-136.

The Global Prioritized Index Register shows the event number of the highest priority event pending across all the host interrupts.

Figure 30-146. PRU_ICSS_INTC_GPIR Register

31	30	29	28	27	26	25	24
GLB_NONE	RESERVED						
R-1h	R-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED						GLB_PRI_INTR	
R-0h						R-0h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GLB_PRI_INTR							
R-0h							

Table 30-136. PRU_ICSS_INTC_GPIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GLB_NONE	R	1h	No Interrupt is pending. Can be used by host to test for a negative value to see if no interrupts are pending.
30-10	RESERVED	R	0h	
9-0	GLB_PRI_INTR	R	0h	The currently highest priority event index (0-63) pending across all the host interrupts.

30.5.3.12 PRU_ICSS_INTC_SRSR0 Register (offset = 200h) [reset = 0h]

PRU_ICSS_INTC_SRSR0 is shown in Figure 30-147 and described in Table 30-137.

The System Event Status Raw/Set Register0 show the pending enabled status of the system event 0 to 31. Software can write to the Status Set Registers to set a system event without a hardware trigger. There is one bit per system event.

Figure 30-147. PRU_ICSS_INTC_SRSR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW_STS_31_0																															
R/W-0h																															

Table 30-137. PRU_ICSS_INTC_SRSR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RAW_STS_31_0	R/W	0h	System event raw status and setting of the system events 0 to 31. Reads return the raw status. Write a 1 in a bit position to set the status of the system event. Writing a 0 has no effect.

30.5.3.13 PRU_ICSS_INTC_SRSR1 Register (offset = 204h) [reset = 0h]

PRU_ICSS_INTC_SRSR1 is shown in Figure 30-148 and described in Table 30-138.

The System Event Status Raw/Set Register1 show the pending enabled status of the system events 32 to 63. Software can write to the Status Set Registers to set a system event without a hardware trigger. There is one bit per system event.

Figure 30-148. PRU_ICSS_INTC_SRSR1 Register

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="32">RAW_STS_63_32</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 30-138. PRU_ICSS_INTC_SRSR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RAW_STS_63_32	R/W	0h	System event raw status and setting of the system events 32 to 63. Reads return the raw status. Write a 1 in a bit position to set the status of the system event. Writing a 0 has no effect.

30.5.3.14 PRU_ICSS_INTC_SECR0 Register (offset = 280h) [reset = 0h]

PRU_ICSS_INTC_SECR0 is shown in Figure 30-149 and described in Table 30-139.

The System Event Status Enabled/Clear Register0 show the pending enabled status of the system events 0 to 31. Software can write to the Status Clear Registers to clear a system event after it has been serviced. If a system event status is not cleared then another host interrupt may not be triggered or another host interrupt may be triggered incorrectly. There is one bit per system event.

Figure 30-149. PRU_ICSS_INTC_SECR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENA_STS_31_0																															
R/W-0h																															

Table 30-139. PRU_ICSS_INTC_SECR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ENA_STS_31_0	R/W	0h	System event enabled status and clearing of the system events 0 to 31. Reads return the enabled status (before enabling with the Enable Registers). Write a 1 in a bit position to clear the status of the system event. Writing a 0 has no effect.

30.5.3.15 PRU_ICSS_INTC_SECR1 Register (offset = 284h) [reset = 0h]

PRU_ICSS_INTC_SECR1 is shown in Figure 30-150 and described in Table 30-140.

The System Event Status Enabled/Clear Register1 show the pending enabled status of the system events 32 to 63. Software can write to the Status Clear Registers to clear a system event after it has been serviced. If a system event status is not cleared then another host interrupt may not be triggered or another host interrupt may be triggered incorrectly. There is one bit per system event.

Figure 30-150. PRU_ICSS_INTC_SECR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENA_STS_63_32																															
R/W-0h																															

Table 30-140. PRU_ICSS_INTC_SECR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ENA_STS_63_32	R/W	0h	System event enabled status and clearing of the system event 32 to 63. Reads return the enabled status (before enabling with the Enable Registers). Write a 1 in a bit position to clear the status of the system event. Writing a 0 has no effect.

30.5.3.16 PRU_ICSS_INTC_ESR0 Register (offset = 300h) [reset = 0h]

PRU_ICSS_INTC_ESR0 is shown in Figure 30-151 and described in Table 30-141.

The System Event Enable Set Register0 enables system events 0 to 31 to trigger outputs. System events that are not enabled do not interrupt the host. There is a bit per system event.

Figure 30-151. PRU_ICSS_INTC_ESR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN_SET_31_0																															
0h																															

Table 30-141. PRU_ICSS_INTC_ESR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	EN_SET_31_0		0h	System event enables system events 0 to 31. Read returns the enable value (0 = disabled, 1 = enabled). Write a 1 in a bit position to set that enable. Writing a 0 has no effect.

30.5.3.17 PRU_ICSS_INTC_ERS1 Register (offset = 304h) [reset = 0h]

PRU_ICSS_INTC_ERS1 is shown in Figure 30-152 and described in Table 30-142.

The System Event Enable Set Register1 enables system events 32 to 63 to trigger outputs. System events that are not enabled do not interrupt the host. There is a bit per system event.

Figure 30-152. PRU_ICSS_INTC_ERS1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN_SET_63_32																															
0h																															

Table 30-142. PRU_ICSS_INTC_ERS1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	EN_SET_63_32		0h	System event enables system events 32 to 63. Read returns the enable value (0 = disabled, 1 = enabled). Write a 1 in a bit position to set that enable. Writing a 0 has no effect.

30.5.3.18 PRU_ICSS_INTC_ECR0 Register (offset = 380h) [reset = 0h]

PRU_ICSS_INTC_ECR0 is shown in Figure 30-153 and described in Table 30-143.

The System Event Enable Clear Register0 disables system events 0 to 31 to map to channels. System events that are not enabled do not interrupt the host. There is a bit per system event.

Figure 30-153. PRU_ICSS_INTC_ECR0 Register

31–0
EN_CLR_31_0
0h

Table 30-143. PRU_ICSS_INTC_ECR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	EN_CLR_31_0		0h	System event enables system events 0 to 31. Read returns the enable value (0 = disabled, 1 = enabled). Write a 1 in a bit position to clear that enable. Writing a 0 has no effect.

30.5.3.19 PRU_ICSS_INTC_ECR1 Register (offset = 384h) [reset = 0h]

PRU_ICSS_INTC_ECR1 is shown in Figure 30-154 and described in Table 30-144.

The System Event Enable Clear Register1 disables system events 32 to 63 to map to channels. System events that are not enabled do not interrupt the host. There is a bit per system event.

Figure 30-154. PRU_ICSS_INTC_ECR1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">EN_CLR_63_32</td></tr>
<tr><td colspan="32">0h</td></tr>
</table>

Table 30-144. PRU_ICSS_INTC_ECR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	EN_CLR_63_32		0h	System event enables system events 32 to 63. Read returns the enable value (0 = disabled, 1 = enabled). Write a 1 in a bit position to clear that enable. Writing a 0 has no effect.

30.5.3.20 PRU_ICSS_INTC_CMR0 Register (offset = 400h) [reset = 0h]

PRU_ICSS_INTC_CMR0 is shown in Figure 30-155 and described in Table 30-145.

The Channel Map Register0 specify the channel (0-9)for the system events 0 to 3. There is one register per 4 system events.

Figure 30-155. PRU_ICSS_INTC_CMR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				CH_MAP_3				RESERVED				CH_MAP_2			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CH_MAP_1				RESERVED				CH_MAP_0			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 30-145. PRU_ICSS_INTC_CMR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	CH_MAP_3	R/W	0h	Sets the channel (0-9)for the system event 3
23-20	RESERVED	R/W	0h	
19-16	CH_MAP_2	R/W	0h	Sets the channel (0-9)for the system event 2
15-12	RESERVED	R/W	0h	
11-8	CH_MAP_1	R/W	0h	Sets the channel (0-9)for the system event 1
7-4	RESERVED	R/W	0h	
3-0	CH_MAP_0	R/W	0h	Sets the channel (0-9)for the system event 0

30.5.3.21 PRU_ICSS_INTC_CMR1 Register (offset = 404h) [reset = 0h]

PRU_ICSS_INTC_CMR1 is shown in Figure 30-156 and described in Table 30-146.

The Channel Map Register1 specify the channel for the system events 4 to 7. There is one register per 4 system events.

Figure 30-156. PRU_ICSS_INTC_CMR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				CH_MAP_7				RESERVED				CH_MAP_6			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CH_MAP_5				RESERVED				CH_MAP_4			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 30-146. PRU_ICSS_INTC_CMR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	CH_MAP_7	R/W	0h	Sets the channel for the system event 7
23-20	RESERVED	R/W	0h	
19-16	CH_MAP_6	R/W	0h	Sets the channel for the system event 6
15-12	RESERVED	R/W	0h	
11-8	CH_MAP_5	R/W	0h	Sets the channel for the system event 5
7-4	RESERVED	R/W	0h	
3-0	CH_MAP_4	R/W	0h	Sets the channel for the system event 4

30.5.3.22 PRU_ICSS_INTC_CMR2 Register (offset = 408h) [reset = 0h]

PRU_ICSS_INTC_CMR2 is shown in Figure 30-157 and described in Table 30-147.

The Channel Map Register2 specify the channel for the system events 8 to 11. There is one register per 4 system events.

Figure 30-157. PRU_ICSS_INTC_CMR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				CH_MAP_11				RESERVED				CH_MAP_10			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CH_MAP_9				RESERVED				CH_MAP_8			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 30-147. PRU_ICSS_INTC_CMR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	CH_MAP_11	R/W	0h	Sets the channel for the system event 11
23-20	RESERVED	R/W	0h	
19-16	CH_MAP_10	R/W	0h	Sets the channel for the system event 10
15-12	RESERVED	R/W	0h	
11-8	CH_MAP_9	R/W	0h	Sets the channel for the system event 9
7-4	RESERVED	R/W	0h	
3-0	CH_MAP_8	R/W	0h	Sets the channel for the system event 8

30.5.3.23 PRU_ICSS_INTC_CMR3 Register (offset = 40Ch) [reset = 0h]

PRU_ICSS_INTC_CMR3 is shown in Figure 30-158 and described in Table 30-148.

The Channel Map Register3 specify the channel for the system events 12 to 15. There is one register per 4 system events.

Figure 30-158. PRU_ICSS_INTC_CMR3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				CH_MAP_15				RESERVED				CH_MAP_14			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CH_MAP_13				RESERVED				CH_MAP_12			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 30-148. PRU_ICSS_INTC_CMR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	CH_MAP_15	R/W	0h	Sets the channel for the system event 15
23-20	RESERVED	R/W	0h	
19-16	CH_MAP_14	R/W	0h	Sets the channel for the system event 14
15-12	RESERVED	R/W	0h	
11-8	CH_MAP_13	R/W	0h	Sets the channel for the system event 13
7-4	RESERVED	R/W	0h	
3-0	CH_MAP_12	R/W	0h	Sets the channel for the system event 12

30.5.3.24 PRU_ICSS_INTC_CMR4 Register (offset = 410h) [reset = 0h]

PRU_ICSS_INTC_CMR4 is shown in Figure 30-159 and described in Table 30-149.

The Channel Map Register4 specify the channel for the system events 16 to 19. There is one register per 4 system events.

Figure 30-159. PRU_ICSS_INTC_CMR4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				CH_MAP_19				RESERVED				CH_MAP_18			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CH_MAP_17				RESERVED				CH_MAP_16			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 30-149. PRU_ICSS_INTC_CMR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	CH_MAP_19	R/W	0h	Sets the channel for the system event 19
23-20	RESERVED	R/W	0h	
19-16	CH_MAP_18	R/W	0h	Sets the channel for the system event 18
15-12	RESERVED	R/W	0h	
11-8	CH_MAP_17	R/W	0h	Sets the channel for the system event 17
7-4	RESERVED	R/W	0h	
3-0	CH_MAP_16	R/W	0h	Sets the channel for the system event 16

30.5.3.25 PRU_ICSS_INTC_CMR5 Register (offset = 414h) [reset = 0h]

PRU_ICSS_INTC_CMR5 is shown in Figure 30-160 and described in Table 30-150.

The Channel Map Register5 specify the channel for the system events 20 to 23. There is one register per 4 system events.

Figure 30-160. PRU_ICSS_INTC_CMR5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				CH_MAP_23				RESERVED				CH_MAP_22			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CH_MAP_21				RESERVED				CH_MAP_20			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 30-150. PRU_ICSS_INTC_CMR5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	CH_MAP_23	R/W	0h	Sets the channel for the system event 23
23-20	RESERVED	R/W	0h	
19-16	CH_MAP_22	R/W	0h	Sets the channel for the system event 22
15-12	RESERVED	R/W	0h	
11-8	CH_MAP_21	R/W	0h	Sets the channel for the system event 21
7-4	RESERVED	R/W	0h	
3-0	CH_MAP_20	R/W	0h	Sets the channel for the system event 20

30.5.3.26 PRU_ICSS_INTC_CMR6 Register (offset = 418h) [reset = 0h]

PRU_ICSS_INTC_CMR6 is shown in Figure 30-161 and described in Table 30-151.

The Channel Map Register6 specify the channel for the system events 24 to 27. There is one register per 4 system events.

Figure 30-161. PRU_ICSS_INTC_CMR6 Register

31-28	27-24	23-20	19-16
RESERVED	CH_MAP_27	RESERVED	CH_MAP_26
R/W-0h	R/W-0h	R/W-0h	R/W-0h

15-12	11-8	7-4	3-0
RESERVED	CH_MAP_25	RESERVED	CH_MAP_24
R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 30-151. PRU_ICSS_INTC_CMR6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	CH_MAP_27	R/W	0h	Sets the channel for the system event 27
23-20	RESERVED	R/W	0h	
19-16	CH_MAP_26	R/W	0h	Sets the channel for the system event 26
15-12	RESERVED	R/W	0h	
11-8	CH_MAP_25	R/W	0h	Sets the channel for the system event 25
7-4	RESERVED	R/W	0h	
3-0	CH_MAP_24	R/W	0h	Sets the channel for the system event 24

30.5.3.27 PRU_ICSS_INTC_CMR7 Register (offset = 41Ch) [reset = 0h]

PRU_ICSS_INTC_CMR7 is shown in Figure 30-162 and described in Table 30-152.

The Channel Map Register7 specify the channel for the system events 28 to 31. There is one register per 4 system events.

Figure 30-162. PRU_ICSS_INTC_CMR7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				CH_MAP_31				RESERVED				CH_MAP_30			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CH_MAP_29				RESERVED				CH_MAP_28			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 30-152. PRU_ICSS_INTC_CMR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	CH_MAP_31	R/W	0h	Sets the channel for the system event 31
23-20	RESERVED	R/W	0h	
19-16	CH_MAP_30	R/W	0h	Sets the channel for the system event 30
15-12	RESERVED	R/W	0h	
11-8	CH_MAP_29	R/W	0h	Sets the channel for the system event 29
7-4	RESERVED	R/W	0h	
3-0	CH_MAP_28	R/W	0h	Sets the channel for the system event 28

30.5.3.28 PRU_ICSS_INTC_CMR8 Register (offset = 420h) [reset = 0h]

PRU_ICSS_INTC_CMR8 is shown in Figure 30-163 and described in Table 30-153.

The Channel Map Register8 specify the channel for the system events 32 to 35. There is one register per 4 system events.

Figure 30-163. PRU_ICSS_INTC_CMR8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				CH_MAP_35				RESERVED				CH_MAP_34			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CH_MAP_33				RESERVED				CH_MAP_32			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 30-153. PRU_ICSS_INTC_CMR8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	CH_MAP_35	R/W	0h	Sets the channel for the system event 35
23-20	RESERVED	R/W	0h	
19-16	CH_MAP_34	R/W	0h	Sets the channel for the system event 34
15-12	RESERVED	R/W	0h	
11-8	CH_MAP_33	R/W	0h	Sets the channel for the system event 33
7-4	RESERVED	R/W	0h	
3-0	CH_MAP_32	R/W	0h	Sets the channel for the system event 32

30.5.3.29 PRU_ICSS_INTC_CMR9 Register (offset = 424h) [reset = 0h]

PRU_ICSS_INTC_CMR9 is shown in Figure 30-164 and described in Table 30-154.

The Channel Map Register9 specify the channel for the system events 36 to 39. There is one register per 4 system events.

Figure 30-164. PRU_ICSS_INTC_CMR9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				CH_MAP_39				RESERVED				CH_MAP_38			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CH_MAP_37				RESERVED				CH_MAP_36			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 30-154. PRU_ICSS_INTC_CMR9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	CH_MAP_39	R/W	0h	Sets the channel for the system event 39
23-20	RESERVED	R/W	0h	
19-16	CH_MAP_38	R/W	0h	Sets the channel for the system event 38
15-12	RESERVED	R/W	0h	
11-8	CH_MAP_37	R/W	0h	Sets the channel for the system event 37
7-4	RESERVED	R/W	0h	
3-0	CH_MAP_36	R/W	0h	Sets the channel for the system event 36

30.5.3.30 PRU_ICSS_INTC_CMR10 Register (offset = 428h) [reset = 0h]

PRU_ICSS_INTC_CMR10 is shown in Figure 30-165 and described in Table 30-155.

The Channel Map Register10 specify the channel for the system events 40 to 43. There is one register per 4 system events.

Figure 30-165. PRU_ICSS_INTC_CMR10 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">CH_MAP_43</td><td colspan="4">RESERVED</td><td colspan="4">CH_MAP_42</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">CH_MAP_41</td><td colspan="4">RESERVED</td><td colspan="4">CH_MAP_40</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 30-155. PRU_ICSS_INTC_CMR10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	CH_MAP_43	R/W	0h	Sets the channel for the system event 43
23-20	RESERVED	R/W	0h	
19-16	CH_MAP_42	R/W	0h	Sets the channel for the system event 42
15-12	RESERVED	R/W	0h	
11-8	CH_MAP_41	R/W	0h	Sets the channel for the system event 41
7-4	RESERVED	R/W	0h	
3-0	CH_MAP_40	R/W	0h	Sets the channel for the system event 40

30.5.3.31 PRU_ICSS_INTC_CMR11 Register (offset = 42Ch) [reset = 0h]

PRU_ICSS_INTC_CMR11 is shown in Figure 30-166 and described in Table 30-156.

The Channel Map Register11 specify the channel for the system events 44 to 47. There is one register per 4 system events.

Figure 30-166. PRU_ICSS_INTC_CMR11 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				CH_MAP_47				RESERVED				CH_MAP_46			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CH_MAP_45				RESERVED				CH_MAP_44			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 30-156. PRU_ICSS_INTC_CMR11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	CH_MAP_47	R/W	0h	Sets the channel for the system event 47
23-20	RESERVED	R/W	0h	
19-16	CH_MAP_46	R/W	0h	Sets the channel for the system event 46
15-12	RESERVED	R/W	0h	
11-8	CH_MAP_45	R/W	0h	Sets the channel for the system event 45
7-4	RESERVED	R/W	0h	
3-0	CH_MAP_44	R/W	0h	Sets the channel for the system event 44

30.5.3.32 PRU_ICSS_INTC_CMR12 Register (offset = 430h) [reset = 0h]

PRU_ICSS_INTC_CMR12 is shown in Figure 30-167 and described in Table 30-157.

The Channel Map Register12 specify the channel for the system events 48 to 51. There is one register per 4 system events.

Figure 30-167. PRU_ICSS_INTC_CMR12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				CH_MAP_51				RESERVED				CH_MAP_50			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CH_MAP_49				RESERVED				CH_MAP_48			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 30-157. PRU_ICSS_INTC_CMR12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	CH_MAP_51	R/W	0h	Sets the channel for the system event 51
23-20	RESERVED	R/W	0h	
19-16	CH_MAP_50	R/W	0h	Sets the channel for the system event 50
15-12	RESERVED	R/W	0h	
11-8	CH_MAP_49	R/W	0h	Sets the channel for the system event 49
7-4	RESERVED	R/W	0h	
3-0	CH_MAP_48	R/W	0h	Sets the channel for the system event 48

30.5.3.33 PRU_ICSS_INTC_CMR13 Register (offset = 434h) [reset = 0h]

PRU_ICSS_INTC_CMR13 is shown in Figure 30-168 and described in Table 30-158.

The Channel Map Register13 specify the channel for the system events 52 to 55. There is one register per 4 system events.

Figure 30-168. PRU_ICSS_INTC_CMR13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				CH_MAP_55				RESERVED				CH_MAP_54			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CH_MAP_53				RESERVED				CH_MAP_52			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 30-158. PRU_ICSS_INTC_CMR13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	CH_MAP_55	R/W	0h	Sets the channel for the system event 55
23-20	RESERVED	R/W	0h	
19-16	CH_MAP_54	R/W	0h	Sets the channel for the system event 54
15-12	RESERVED	R/W	0h	
11-8	CH_MAP_53	R/W	0h	Sets the channel for the system event 53
7-4	RESERVED	R/W	0h	
3-0	CH_MAP_52	R/W	0h	Sets the channel for the system event 52

30.5.3.34 PRU_ICSS_INTC_CMR14 Register (offset = 438h) [reset = 0h]

PRU_ICSS_INTC_CMR14 is shown in Figure 30-169 and described in Table 30-159.

The Channel Map Register14 specify the channel for the system events 56 to 59. There is one register per 4 system events.

Figure 30-169. PRU_ICSS_INTC_CMR14 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">CH_MAP_59</td><td colspan="4">RESERVED</td><td colspan="4">CH_MAP_58</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">CH_MAP_57</td><td colspan="4">RESERVED</td><td colspan="4">CH_MAP_56</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 30-159. PRU_ICSS_INTC_CMR14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	CH_MAP_59	R/W	0h	Sets the channel for the system event 59
23-20	RESERVED	R/W	0h	
19-16	CH_MAP_58	R/W	0h	Sets the channel for the system event 58
15-12	RESERVED	R/W	0h	
11-8	CH_MAP_57	R/W	0h	Sets the channel for the system event 57
7-4	RESERVED	R/W	0h	
3-0	CH_MAP_56	R/W	0h	Sets the channel for the system event 56

30.5.3.35 PRU_ICSS_INTC_CMR15 Register (offset = 43Ch) [reset = 0h]

PRU_ICSS_INTC_CMR15 is shown in Figure 30-170 and described in Table 30-160.

The Channel Map Register15 specify the channel for the system events 60 to 63. There is one register per 4 system events.

Figure 30-170. PRU_ICSS_INTC_CMR15 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				CH_MAP_63				RESERVED				CH_MAP_62			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				CH_MAP_61				RESERVED				CH_MAP_60			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 30-160. PRU_ICSS_INTC_CMR15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	CH_MAP_63	R/W	0h	Sets the channel for the system event 63
23-20	RESERVED	R/W	0h	
19-16	CH_MAP_62	R/W	0h	Sets the channel for the system event 62
15-12	RESERVED	R/W	0h	
11-8	CH_MAP_61	R/W	0h	Sets the channel for the system event 61
7-4	RESERVED	R/W	0h	
3-0	CH_MAP_60	R/W	0h	Sets the channel for the system event 60

30.5.3.36 PRU_ICSS_INTC_HMR0 Register (offset = 800h) [reset = 0h]

PRU_ICSS_INTC_HMR0 is shown in Figure 30-171 and described in Table 30-161.

The Host Interrupt Map Register0 define the host interrupt for channels 0 to 3. There is one register per 4 channels. Channels with forced host interrupt mappings will have their fields read-only.

Figure 30-171. PRU_ICSS_INTC_HMR0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">HINT_MAP_3</td><td colspan="4">RESERVED</td><td colspan="4">HINT_MAP_2</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="4">HINT_MAP_1</td><td colspan="4">RESERVED</td><td colspan="4">HINT_MAP_0</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
</table>

Table 30-161. PRU_ICSS_INTC_HMR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	HINT_MAP_3	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 3
23-20	RESERVED	R/W	0h	
19-16	HINT_MAP_2	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 2
15-12	RESERVED	R/W	0h	
11-8	HINT_MAP_1	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 1
7-4	RESERVED	R/W	0h	
3-0	HINT_MAP_0	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 0

30.5.3.37 PRU_ICSS_INTC_HMR1 Register (offset = 804h) [reset = 0h]

PRU_ICSS_INTC_HMR1 is shown in Figure 30-172 and described in Table 30-162.

The Host Interrupt Map Register1 define the host interrupt for channels 4 to 7. There is one register per 4 channels. Channels with forced host interrupt mappings will have their fields read-only.

Figure 30-172. PRU_ICSS_INTC_HMR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				HINT_MAP_7				RESERVED				HINT_MAP_6			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				HINT_MAP_5				RESERVED				HINT_MAP_4			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 30-162. PRU_ICSS_INTC_HMR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	HINT_MAP_7	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 7
23-20	RESERVED	R/W	0h	
19-16	HINT_MAP_6	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 6
15-12	RESERVED	R/W	0h	
11-8	HINT_MAP_5	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 5
7-4	RESERVED	R/W	0h	
3-0	HINT_MAP_4	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 4

30.5.3.38 PRU_ICSS_INTC_HMR2 Register (offset = 808h) [reset = 0h]

PRU_ICSS_INTC_HMR2 is shown in Figure 30-173 and described in Table 30-163.

The Host Interrupt Map Register2 define the host interrupt for channels 8 to 9. There is one register per 4 channels. Channels with forced host interrupt mappings will have their fields read-only.

Figure 30-173. PRU_ICSS_INTC_HMR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				HINT_MAP_9				RESERVED				HINT_MAP_8			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 30-163. PRU_ICSS_INTC_HMR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-8	HINT_MAP_9	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 9
7-4	RESERVED	R/W	0h	
3-0	HINT_MAP_8	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 8

30.5.3.39 PRU_ICSS_INTC_HIPIR0 Register (offset = 900h) [reset = 80000000h]

PRU_ICSS_INTC_HIPIR0 is shown in Figure 30-174 and described in Table 30-164.

The Host Interrupt Prioritized Index Register0 shows the highest priority current pending interrupt for the host interrupt 0. There is one register per host interrupt.

Figure 30-174. PRU_ICSS_INTC_HIPIR0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>NONE_HINT_0</td><td colspan="7">RESERVED</td></tr>
<tr><td>R-1h</td><td colspan="7">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">PRI_HINT_0</td></tr>
<tr><td colspan="6">R-0h</td><td colspan="2">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">PRI_HINT_0</td></tr>
<tr><td colspan="8">R-0h</td></tr>
</table>

Table 30-164. PRU_ICSS_INTC_HIPIR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_0	R	1h	No pending interrupt.
30-10	RESERVED	R	0h	
9-0	PRI_HINT_0	R	0h	HOST INT 0 PRIORITIZED INTERRUPT. Interrupt number of the highest priority pending interrupt for this host interrupt.

30.5.3.40 PRU_ICSS_INTC_HIPIR1 Register (offset = 904h) [reset = 80000000h]

PRU_ICSS_INTC_HIPIR1 is shown in Figure 30-175 and described in Table 30-165.

The Host Interrupt Prioritized Index Register1 shows the highest priority current pending interrupt for the host interrupt 1. There is one register per host interrupt.

Figure 30-175. PRU_ICSS_INTC_HIPIR1 Register

31	30	29	28	27	26	25	24
NONE_HINT_1	RESERVED						
R-1h	R-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED						PRI_HINT_1	
R-0h						R-0h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRI_HINT_1							
R-0h							

Table 30-165. PRU_ICSS_INTC_HIPIR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_1	R	1h	No pending interrupt.
30-10	RESERVED	R	0h	
9-0	PRI_HINT_1	R	0h	HOST INT 1 PRIORITIZED INTERRUPT. Interrupt number of the highest priority pending interrupt for this host interrupt.

30.5.3.41 PRU_ICSS_INTC_HIPIR2 Register (offset = 908h) [reset = 80000000h]

PRU_ICSS_INTC_HIPIR2 is shown in Figure 30-176 and described in Table 30-166.

The Host Interrupt Prioritized Index Register2 shows the highest priority current pending interrupt for the host interrupt 2. There is one register per host interrupt.

Figure 30-176. PRU_ICSS_INTC_HIPIR2 Register

31	30	29	28	27	26	25	24
NONE_HINT_2	RESERVED						
R-1h	R-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED						PRI_HINT_2	
R-0h						R-0h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRI_HINT_2							
R-0h							

Table 30-166. PRU_ICSS_INTC_HIPIR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_2	R	1h	No pending interrupt.
30-10	RESERVED	R	0h	
9-0	PRI_HINT_2	R	0h	HOST INT 2 PRIORITIZED INTERRUPT. Interrupt number of the highest priority pending interrupt for this host interrupt.

30.5.3.42 PRU_ICSS_INTC_HIPIR3 Register (offset = 90Ch) [reset = 80000000h]

PRU_ICSS_INTC_HIPIR3 is shown in Figure 30-177 and described in Table 30-167.

The Host Interrupt Prioritized Index Register3 shows the highest priority current pending interrupt for the host interrupt 3. There is one register per host interrupt.

Figure 30-177. PRU_ICSS_INTC_HIPIR3 Register

31	30	29	28	27	26	25	24
NONE_HINT_3	RESERVED						
R-1h	R-0h						

23	22	21	20	19	18	17	16
RESERVED							
R-0h							

15	14	13	12	11	10	9	8
RESERVED						PRI_HINT_3	
R-0h						R-0h	

7	6	5	4	3	2	1	0
PRI_HINT_3							
R-0h							

Table 30-167. PRU_ICSS_INTC_HIPIR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_3	R	1h	No pending interrupt.
30-10	RESERVED	R	0h	
9-0	PRI_HINT_3	R	0h	HOST INT 3 PRIORITIZED INTERRUPT. Interrupt number of the highest priority pending interrupt for this host interrupt.

30.5.3.43 PRU_ICSS_INTC_HIPIR4 Register (offset = 910h) [reset = 80000000h]

PRU_ICSS_INTC_HIPIR4 is shown in Figure 30-178 and described in Table 30-168.

The Host Interrupt Prioritized Index Register4 shows the highest priority current pending interrupt for the host interrupt 4. There is one register per host interrupt.

Figure 30-178. PRU_ICSS_INTC_HIPIR4 Register

31	30	29	28	27	26	25	24
NONE_HINT_4	RESERVED						
R-1h	R-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED						PRI_HINT_4	
R-0h						R-0h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRI_HINT_4							
R-0h							

Table 30-168. PRU_ICSS_INTC_HIPIR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_4	R	1h	No pending interrupt.
30-10	RESERVED	R	0h	
9-0	PRI_HINT_4	R	0h	HOST INT 4 PRIORITIZED INTERRUPT. Interrupt number of the highest priority pending interrupt for this host interrupt.

30.5.3.44 PRU_ICSS_INTC_HIPIR5 Register (offset = 914h) [reset = 80000000h]

PRU_ICSS_INTC_HIPIR5 is shown in Figure 30-179 and described in Table 30-169.

The Host Interrupt Prioritized Index Register5 shows the highest priority current pending interrupt for the host interrupt 5. There is one register per host interrupt.

Figure 30-179. PRU_ICSS_INTC_HIPIR5 Register

31	30	29	28	27	26	25	24
NONE_HINT_5	RESERVED						
R-1h	R-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED						PRI_HINT_5	
R-0h						R-0h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRI_HINT_5							
R-0h							

Table 30-169. PRU_ICSS_INTC_HIPIR5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_5	R	1h	No pending interrupt.
30-10	RESERVED	R	0h	
9-0	PRI_HINT_5	R	0h	HOST INT 5 PRIORITIZED INTERRUPT. Interrupt number of the highest priority pending interrupt for this host interrupt.

30.5.3.45 PRU_ICSS_INTC_HIPIR6 Register (offset = 918h) [reset = 80000000h]

PRU_ICSS_INTC_HIPIR6 is shown in Figure 30-180 and described in Table 30-170.

The Host Interrupt Prioritized Index Register6 shows the highest priority current pending interrupt for the host interrupt 6. There is one register per host interrupt.

Figure 30-180. PRU_ICSS_INTC_HIPIR6 Register

31	30	29	28	27	26	25	24
NONE_HINT_6	RESERVED						
R-1h	R-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED						PRI_HINT_6	
R-0h						R-0h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRI_HINT_6							
R-0h							

Table 30-170. PRU_ICSS_INTC_HIPIR6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_6	R	1h	No pending interrupt.
30-10	RESERVED	R	0h	
9-0	PRI_HINT_6	R	0h	HOST INT 6 PRIORITIZED INTERRUPT. Interrupt number of the highest priority pending interrupt for this host interrupt.

30.5.3.46 PRU_ICSS_INTC_HIPIR7 Register (offset = 91Ch) [reset = 80000000h]

PRU_ICSS_INTC_HIPIR7 is shown in Figure 30-181 and described in Table 30-171.

The Host Interrupt Prioritized Index Register7 shows the highest priority current pending interrupt for the host interrupt 7. There is one register per host interrupt.

Figure 30-181. PRU_ICSS_INTC_HIPIR7 Register

31	30	29	28	27	26	25	24
NONE_HINT_7	RESERVED						
R-1h	R-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED						PRI_HINT_7	
R-0h						R-0h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRI_HINT_7							
R-0h							

Table 30-171. PRU_ICSS_INTC_HIPIR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_7	R	1h	No pending interrupt.
30-10	RESERVED	R	0h	
9-0	PRI_HINT_7	R	0h	HOST INT 7 PRIORITIZED INTERRUPT. Interrupt number of the highest priority pending interrupt for this host interrupt.

30.5.3.47 PRU_ICSS_INTC_HIPIR8 Register (offset = 920h) [reset = 80000000h]

PRU_ICSS_INTC_HIPIR8 is shown in Figure 30-182 and described in Table 30-172.

The Host Interrupt Prioritized Index Register8 shows the highest priority current pending interrupt for the host interrupt 8. There is one register per host interrupt.

Figure 30-182. PRU_ICSS_INTC_HIPIR8 Register

31	30	29	28	27	26	25	24
NONE_HINT_8	RESERVED						
R-1h	R-0h						
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED						PRI_HINT_8	
R-0h						R-0h	
7	6	5	4	3	2	1	0
PRI_HINT_8							
R-0h							

Table 30-172. PRU_ICSS_INTC_HIPIR8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_8	R	1h	No pending interrupt.
30-10	RESERVED	R	0h	
9-0	PRI_HINT_8	R	0h	HOST INT 8 PRIORITIZED INTERRUPT. Interrupt number of the highest priority pending interrupt for this host interrupt.

30.5.3.48 PRU_ICSS_INTC_HIPIR9 Register (offset = 924h) [reset = 80000000h]

PRU_ICSS_INTC_HIPIR9 is shown in Figure 30-183 and described in Table 30-173.

The Host Interrupt Prioritized Index Register9 shows the highest priority current pending interrupt for the host interrupt 9. There is one register per host interrupt.

Figure 30-183. PRU_ICSS_INTC_HIPIR9 Register

31	30	29	28	27	26	25	24
NONE_HINT_9	RESERVED						
R-1h	R-0h						
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED						PRI_HINT_9	
R-0h						R-0h	
7	6	5	4	3	2	1	0
PRI_HINT_9							
R-0h							

Table 30-173. PRU_ICSS_INTC_HIPIR9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_9	R	1h	No pending interrupt.
30-10	RESERVED	R	0h	
9-0	PRI_HINT_9	R	0h	HOST INT 9 PRIORITIZED INTERRUPT. Interrupt number of the highest priority pending interrupt for this host interrupt.

30.5.3.49 PRU_ICSS_INTC_SIPR0 Register (offset = D00h) [reset = 1h]

PRU_ICSS_INTC_SIPR0 is shown in Figure 30-184 and described in Table 30-174.

The System Event Polarity Register0 define the polarity of the system events 0 to 31. There is a polarity for each system event. The polarity of all system events is active high; always write 1 to the bits of this register.

Figure 30-184. PRU_ICSS_INTC_SIPR0 Register

31-0
POLARITY_31_0
R/W-1h

Table 30-174. PRU_ICSS_INTC_SIPR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	POLARITY_31_0	R/W	1h	Interrupt polarity of the system events 0 to 31. 0 = active low. 1 = active high.

30.5.3.50 PRU_ICSS_INTC_SIPR1 Register (offset = D04h) [reset = 1h]

PRU_ICSS_INTC_SIPR1 is shown in Figure 30-185 and described in Table 30-175.

The System Event Polarity Register1 define the polarity of the system events 32 to 63. There is a polarity for each system event. The polarity of all system events is active high; always write 1 to the bits of this register.

Figure 30-185. PRU_ICSS_INTC_SIPR1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">POLARITY_63_32</td></tr>
<tr><td colspan="32">R/W-1h</td></tr>
</table>

Table 30-175. PRU_ICSS_INTC_SIPR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	POLARITY_63_32	R/W	1h	Interrupt polarity of the system events 32 to 63. 0 = active low. 1 = active high.

30.5.3.51 PRU_ICSS_INTC_SITR0 Register (offset = D80h) [reset = 0h]

PRU_ICSS_INTC_SITR0 is shown in Figure 30-186 and described in Table 30-176.

The System Event Type Register0 define the type of the system events 0 to 31. There is a type for each system event. The type of all system events is pulse; always write 0 to the bits of this register.

Figure 30-186. PRU_ICSS_INTC_SITR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE_31_0																															
R/W-0h																															

Table 30-176. PRU_ICSS_INTC_SITR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TYPE_31_0	R/W	0h	Interrupt type of the system events 0 to 31. 0 = level or pulse interrupt. 1 = edge interrupt (required edge detect).

30.5.3.52 PRU_ICSS_INTC_SITR1 Register (offset = D84h) [reset = 0h]

PRU_ICSS_INTC_SITR1 is shown in Figure 30-187 and described in Table 30-177.

The System Event Type Register1 define the type of the system events 32 to 63. There is a type for each system event. The type of all system events is pulse; always write 0 to the bits of this register.

Figure 30-187. PRU_ICSS_INTC_SITR1 Register

31-0
TYPE_63_32
R/W-0h

Table 30-177. PRU_ICSS_INTC_SITR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TYPE_63_32	R/W	0h	Interrupt type of the system events 32 to 63. 0 = level or pulse interrupt. 1 = edge interrupt (required edge detect).

30.5.3.53 PRU_ICSS_INTC_HINLR0 Register (offset = 1100h) [reset = 100h]

PRU_ICSS_INTC_HINLR0 is shown in Figure 30-188 and described in Table 30-178.

The Host Interrupt Nesting Level Register0 display and control the nesting level for host interrupt 0. The nesting level controls which channel and lower priority channels are nested. There is one register per host interrupt.

Figure 30-188. PRU_ICSS_INTC_HINLR0 Register

31	30	29	28	27	26	25	24
AUTO_OVERRIDE	RESERVED						
W-0h	R/W-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							NEST_HINT_0
R/W-0h							R/W-100h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NEST_HINT_0							
R/W-100h							

Table 30-178. PRU_ICSS_INTC_HINLR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	Reads return 0. Writes of a 1 override the auto updating of the nesting_level and use the write data.
30-9	RESERVED	R/W	0h	
8-0	NEST_HINT_0	R/W	100h	Reads return the current nesting level for the host interrupt. Writes set the nesting level for the host interrupt. In auto mode the value is updated internally unless the auto_override is set and then the write data is used.

30.5.3.54 PRU_ICSS_INTC_HINLR1 Register (offset = 1104h) [reset = 100h]

PRU_ICSS_INTC_HINLR1 is shown in Figure 30-189 and described in Table 30-179.

The Host Interrupt Nesting Level Register1 display and control the nesting level for host interrupt 1. The nesting level controls which channel and lower priority channels are nested. There is one register per host interrupt.

Figure 30-189. PRU_ICSS_INTC_HINLR1 Register

31	30	29	28	27	26	25	24
AUTO_OVERRIDE	RESERVED						
W-0h	R/W-0h						
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							NEST_HINT_1
R/W-0h							R/W-100h
7	6	5	4	3	2	1	0
NEST_HINT_1							
R/W-100h							

Table 30-179. PRU_ICSS_INTC_HINLR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	Reads return 0. Writes of a 1 override the auto updating of the nesting_level and use the write data.
30-9	RESERVED	R/W	0h	
8-0	NEST_HINT_1	R/W	100h	Reads return the current nesting level for the host interrupt. Writes set the nesting level for the host interrupt. In auto mode the value is updated internally unless the auto_override is set and then the write data is used.

30.5.3.55 PRU_ICSS_INTC_HINLR2 Register (offset = 1108h) [reset = 100h]

PRU_ICSS_INTC_HINLR2 is shown in Figure 30-190 and described in Table 30-180.

The Host Interrupt Nesting Level Register2 display and control the nesting level for host interrupt 2. The nesting level controls which channel and lower priority channels are nested. There is one register per host interrupt.

Figure 30-190. PRU_ICSS_INTC_HINLR2 Register

31	30	29	28	27	26	25	24
AUTO_OVERRIDE	RESERVED						
W-0h	R/W-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							NEST_HINT_2
R/W-0h							R/W-100h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NEST_HINT_2							
R/W-100h							

Table 30-180. PRU_ICSS_INTC_HINLR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	Reads return 0. Writes of a 1 override the auto updating of the nesting_level and use the write data.
30-9	RESERVED	R/W	0h	
8-0	NEST_HINT_2	R/W	100h	Reads return the current nesting level for the host interrupt. Writes set the nesting level for the host interrupt. In auto mode the value is updated internally unless the auto_override is set and then the write data is used.

30.5.3.56 PRU_ICSS_INTC_HINLR3 Register (offset = 110Ch) [reset = 100h]

PRU_ICSS_INTC_HINLR3 is shown in Figure 30-191 and described in Table 30-181.

The Host Interrupt Nesting Level Register3 display and control the nesting level for host interrupt 3. The nesting level controls which channel and lower priority channels are nested. There is one register per host interrupt.

Figure 30-191. PRU_ICSS_INTC_HINLR3 Register

31	30	29	28	27	26	25	24
AUTO_OVERRIDE	RESERVED						
W-0h	R/W-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							NEST_HINT_3
R/W-0h							R/W-100h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NEST_HINT_3							
R/W-100h							

Table 30-181. PRU_ICSS_INTC_HINLR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	Reads return 0. Writes of a 1 override the auto updating of the nesting_level and use the write data.
30-9	RESERVED	R/W	0h	
8-0	NEST_HINT_3	R/W	100h	Reads return the current nesting level for the host interrupt. Writes set the nesting level for the host interrupt. In auto mode the value is updated internally unless the auto_override is set and then the write data is used.

30.5.3.57 PRU_ICSS_INTC_HINLR4 Register (offset = 1110h) [reset = 100h]

PRU_ICSS_INTC_HINLR4 is shown in Figure 30-192 and described in Table 30-182.

The Host Interrupt Nesting Level Register4 display and control the nesting level for host interrupt 4. The nesting level controls which channel and lower priority channels are nested. There is one register per host interrupt.

Figure 30-192. PRU_ICSS_INTC_HINLR4 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>AUTO_OVERRIDE</td><td colspan="7">RESERVED</td></tr>
<tr><td>W-0h</td><td colspan="7">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>NEST_HINT_4</td></tr>
<tr><td colspan="7">R/W-0h</td><td>R/W-100h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">NEST_HINT_4</td></tr>
<tr><td colspan="8">R/W-100h</td></tr>
</table>

Table 30-182. PRU_ICSS_INTC_HINLR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	Reads return 0. Writes of a 1 override the auto updating of the nesting_level and use the write data.
30-9	RESERVED	R/W	0h	
8-0	NEST_HINT_4	R/W	100h	Reads return the current nesting level for the host interrupt. Writes set the nesting level for the host interrupt. In auto mode the value is updated internally unless the auto_override is set and then the write data is used.

30.5.3.58 PRU_ICSS_INTC_HINLR5 Register (offset = 1114h) [reset = 100h]

PRU_ICSS_INTC_HINLR5 is shown in Figure 30-193 and described in Table 30-183.

The Host Interrupt Nesting Level Register5 display and control the nesting level for host interrupt 5. The nesting level controls which channel and lower priority channels are nested. There is one register per host interrupt.

Figure 30-193. PRU_ICSS_INTC_HINLR5 Register

31	30	29	28	27	26	25	24
AUTO_OVERRIDE	RESERVED						
W-0h	R/W-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							NEST_HINT_5
R/W-0h							R/W-100h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NEST_HINT_5							
R/W-100h							

Table 30-183. PRU_ICSS_INTC_HINLR5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	Reads return 0. Writes of a 1 override the auto updating of the nesting_level and use the write data.
30-9	RESERVED	R/W	0h	
8-0	NEST_HINT_5	R/W	100h	Reads return the current nesting level for the host interrupt. Writes set the nesting level for the host interrupt. In auto mode the value is updated internally unless the auto_override is set and then the write data is used.

30.5.3.59 PRU_ICSS_INTC_HINLR6 Register (offset = 1118h) [reset = 100h]

PRU_ICSS_INTC_HINLR6 is shown in Figure 30-194 and described in Table 30-184.

The Host Interrupt Nesting Level Register6 display and control the nesting level for host interrupt 6. The nesting level controls which channel and lower priority channels are nested. There is one register per host interrupt.

Figure 30-194. PRU_ICSS_INTC_HINLR6 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>AUTO_OVERRIDE</td><td colspan="7">RESERVED</td></tr>
<tr><td>W-0h</td><td colspan="7">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>NEST_HINT_6</td></tr>
<tr><td colspan="7">R/W-0h</td><td>R/W-100h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">NEST_HINT_6</td></tr>
<tr><td colspan="8">R/W-100h</td></tr>
</table>

Table 30-184. PRU_ICSS_INTC_HINLR6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	Reads return 0. Writes of a 1 override the auto updating of the nesting_level and use the write data.
30-9	RESERVED	R/W	0h	
8-0	NEST_HINT_6	R/W	100h	Reads return the current nesting level for the host interrupt. Writes set the nesting level for the host interrupt. In auto mode the value is updated internally unless the auto_override is set and then the write data is used.

30.5.3.60 PRU_ICSS_INTC_HINLR7 Register (offset = 111Ch) [reset = 100h]

PRU_ICSS_INTC_HINLR7 is shown in Figure 30-195 and described in Table 30-185.

The Host Interrupt Nesting Level Register7 display and control the nesting level for host interrupt 7. The nesting level controls which channel and lower priority channels are nested. There is one register per host interrupt.

Figure 30-195. PRU_ICSS_INTC_HINLR7 Register

31	30	29	28	27	26	25	24
AUTO_OVERRIDE	RESERVED						
W-0h	R/W-0h						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							NEST_HINT_7
R/W-0h							R/W-100h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NEST_HINT_7							
R/W-100h							

Table 30-185. PRU_ICSS_INTC_HINLR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	Reads return 0. Writes of a 1 override the auto updating of the nesting_level and use the write data.
30-9	RESERVED	R/W	0h	
8-0	NEST_HINT_7	R/W	100h	Reads return the current nesting level for the host interrupt. Writes set the nesting level for the host interrupt. In auto mode the value is updated internally unless the auto_override is set and then the write data is used.

30.5.3.61 PRU_ICSS_INTC_HINLR8 Register (offset = 1120h) [reset = 100h]

PRU_ICSS_INTC_HINLR8 is shown in Figure 30-196 and described in Table 30-186.

The Host Interrupt Nesting Level Register8 display and control the nesting level for host interrupt 8. The nesting level controls which channel and lower priority channels are nested. There is one register per host interrupt.

Figure 30-196. PRU_ICSS_INTC_HINLR8 Register

31	30	29	28	27	26	25	24
AUTO_OVERRIDE	RESERVED						
W-0h	R/W-0h						
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							NEST_HINT_8
R/W-0h							R/W-100h
7	6	5	4	3	2	1	0
NEST_HINT_8							
R/W-100h							

Table 30-186. PRU_ICSS_INTC_HINLR8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	Reads return 0. Writes of a 1 override the auto updating of the nesting_level and use the write data.
30-9	RESERVED	R/W	0h	
8-0	NEST_HINT_8	R/W	100h	Reads return the current nesting level for the host interrupt. Writes set the nesting level for the host interrupt. In auto mode the value is updated internally unless the auto_override is set and then the write data is used.

30.5.3.62 PRU_ICSS_INTC_HINLR9 Register (offset = 1124h) [reset = 100h]

PRU_ICSS_INTC_HINLR9 is shown in Figure 30-197 and described in Table 30-187.

The Host Interrupt Nesting Level Register9 display and control the nesting level for host interrupt 9. The nesting level controls which channel and lower priority channels are nested. There is one register per host interrupt.

Figure 30-197. PRU_ICSS_INTC_HINLR9 Register

31	30	29	28	27	26	25	24
AUTO_OVERRIDE	RESERVED						
W-0h	R/W-0h						
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							NEST_HINT_9
R/W-0h							R/W-100h
7	6	5	4	3	2	1	0
NEST_HINT_9							
R/W-100h							

Table 30-187. PRU_ICSS_INTC_HINLR9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	Reads return 0. Writes of a 1 override the auto updating of the nesting_level and use the write data.
30-9	RESERVED	R/W	0h	
8-0	NEST_HINT_9	R/W	100h	Reads return the current nesting level for the host interrupt. Writes set the nesting level for the host interrupt. In auto mode the value is updated internally unless the auto_override is set and then the write data is used.

30.5.3.63 PRU_ICSS_INTC_HIER Register (offset = 1500h) [reset = 0h]

PRU_ICSS_INTC_HIER is shown in Figure 30-198 and described in Table 30-188.

The Host Interrupt Enable Registers enable or disable individual host interrupts. These work separately from the global enables. There is one bit per host interrupt. These bits are updated when writing to the Host Interrupt Enable Index Set and Host Interrupt Enable Index Clear registers.

Figure 30-198. PRU_ICSS_INTC_HIER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						EN_HINT									
R/W-0h																						R/W-0h									

Table 30-188. PRU_ICSS_INTC_HIER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	0h	
9-0	EN_HINT	R/W	0h	The enable of the host interrupts (one per bit). 0 = disabled 1 = enabled

30.5.4 PRU_ICSS_IEP Registers

Table 30-189 lists the memory-mapped registers for the PRU_ICSS_IEP. All register offset addresses not listed in Table 30-189 should be considered as reserved locations and the register contents should not be modified.

Table 30-189. PRU_ICSS_IEP Registers

Offset	Acronym	Register Name	Section
0h	PRU_ICSS_IEP_TMR_GLB_ CFG		Section 30.5.4.1
4h	PRU_ICSS_IEP_TMR_GLB_STS		Section 30.5.4.2
8h	PRU_ICSS_IEP_TMR_COMPEN		Section 30.5.4.3
Ch	PRU_ICSS_IEP_TMR_CNT		Section 30.5.4.4
10h	PRU_ICSS_IEP_TMR_CAP_CFG		Section 30.5.4.5
14h	PRU_ICSS_IEP_TMR_CAP_STS		Section 30.5.4.6
18h	PRU_ICSS_IEP_TMR_CAPR0		Section 30.5.4.7
1Ch	PRU_ICSS_IEP_TMR_CAPR1		Section 30.5.4.8
20h	PRU_ICSS_IEP_TMR_CAPR2		Section 30.5.4.9
24h	PRU_ICSS_IEP_TMR_CAPR3		Section 30.5.4.10
28h	PRU_ICSS_IEP_TMR_CAPR4		Section 30.5.4.11
2Ch	PRU_ICSS_IEP_TMR_CAPR5		Section 30.5.4.12
30h	PRU_ICSS_IEP_TMR_CAPR6		Section 30.5.4.13
34h	PRU_ICSS_IEP_TMR_CAPF6		Section 30.5.4.14
38h	PRU_ICSS_IEP_TMR_CAPR7		Section 30.5.4.15
3Ch	PRU_ICSS_IEP_TMR_CAPF7		Section 30.5.4.16
40h	PRU_ICSS_IEP_TMR_CMP_CFG		Section 30.5.4.17
44h	PRU_ICSS_IEP_TMR_CMP_STS		Section 30.5.4.18
48h	PRU_ICSS_IEP_TMR_CMP0		Section 30.5.4.19
4Ch	PRU_ICSS_IEP_TMR_CMP1		Section 30.5.4.20
50h	PRU_ICSS_IEP_TMR_CMP2		Section 30.5.4.21
54h	PRU_ICSS_IEP_TMR_CMP3		Section 30.5.4.22
58h	PRU_ICSS_IEP_TMR_CMP4		Section 30.5.4.23
5Ch	PRU_ICSS_IEP_TMR_CMP5		Section 30.5.4.24
60h	PRU_ICSS_IEP_TMR_CMP6		Section 30.5.4.25

Table 30-189. PRU_ICSS_IEP Registers (continued)

Offset	Acronym	Register Name	Section
64h	PRU_ICSS_IEP_TMR_CMP7		Section 30.5.4.26
80h	PRU_ICSS_IEP_TMR_RXIPG0		Section 30.5.4.27
84h	PRU_ICSS_IEP_TMR_RXIPG1		Section 30.5.4.28
88h	PRU_ICSS_IEP_TMR_CMP8		Section 30.5.4.29
8Ch	PRU_ICSS_IEP_TMR_CMP9		Section 30.5.4.30
90h	PRU_ICSS_IEP_TMR_CMP10		Section 30.5.4.31
94h	PRU_ICSS_IEP_TMR_CMP11		Section 30.5.4.32
98h	PRU_ICSS_IEP_TMR_CMP12		Section 30.5.4.33
9Ch	PRU_ICSS_IEP_TMR_CMP13		Section 30.5.4.34
A0h	PRU_ICSS_IEP_TMR_CMP14		Section 30.5.4.35
A4h	PRU_ICSS_IEP_TMR_CMP15		Section 30.5.4.36
A8h	PRU_ICSS_IEP_TMR_CNT_RST		Section 30.5.4.37
ACh	PRU_ICSS_IEP_TMR_PWM		Section 30.5.4.38
100h	PRU_ICSS_IEP_SYNC_CTRL		Section 30.5.4.39
104h	PRU_ICSS_IEP_SYNC_FIRST_STAT		Section 30.5.4.40
108h	PRU_ICSS_IEP_SYNC0_STAT		Section 30.5.4.41
10Ch	PRU_ICSS_IEP_SYNC1_STAT		Section 30.5.4.42
110h	PRU_ICSS_IEP_SYNC_PWIDTH		Section 30.5.4.43
114h	PRU_ICSS_IEP_SYNC0_PERIOD		Section 30.5.4.44
118h	PRU_ICSS_IEP_SYNC1_DELAY		Section 30.5.4.45
11Ch	PRU_ICSS_IEP_SYNC_START		Section 30.5.4.46
200h	PRU_ICSS_IEP_WD_PREDIV		Section 30.5.4.47
204h	PRU_ICSS_IEP_PDI_WD_TIM		Section 30.5.4.48
208h	PRU_ICSS_IEP_PD_WD_TIM		Section 30.5.4.49
20Ch	PRU_ICSS_IEP_WD_STS		Section 30.5.4.50
210h	PRU_ICSS_IEP_WD_EXP_CNT		Section 30.5.4.51
214h	PRU_ICSS_IEP_WD_CTRL		Section 30.5.4.52
300h	PRU_ICSS_IEP_DIGIO_CTRL		Section 30.5.4.53
304h	PRU_ICSS_IEP_DIGIO_STATUS		Section 30.5.4.54
308h	PRU_ICSS_IEP_DIGIO_DATA_IN		Section 30.5.4.55
30Ch	PRU_ICSS_IEP_DIGIO_DATA_IN_RAW		Section 30.5.4.56
310h	PRU_ICSS_IEP_DIGIO_DATA_OUT		Section 30.5.4.57
314h	PRU_ICSS_IEP_DIGIO_DATA_OUT_EN		Section 30.5.4.58
318h	PRU_ICSS_IEP_DIGIO_EXP		Section 30.5.4.59

30.5.4.1 PRU_ICSS_IEP_TMR_GLB_CFG Register (Offset = 0h) [reset = 550h]

PRU_ICSS_IEP_TMR_GLB_CFG is shown in Figure 30-199 and described in Table 30-190.

Return to Summary Table.

GLOBAL CONFIGURE

Figure 30-199. PRU_ICSS_IEP_TMR_GLB_CFG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				CMP_INC			
R-0h				R/W-5h			
15	14	13	12	11	10	9	8
CMP_INC							
R/W-5h							
7	6	5	4	3	2	1	0
DEFAULT_INC				RESERVED			CNT_ENABLE
R/W-5h				R-0h			R/W-0h

Table 30-190. PRU_ICSS_IEP_TMR_GLB_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-8	CMP_INC	R/W	5h	Defines the increment value when compensation is active
7-4	DEFAULT_INC	R/W	5h	Defines the default increment value
3-1	RESERVED	R	0h	
0	CNT_ENABLE	R/W	0h	Counter enable 0h (R/W) = Disables the counter. The counter maintains the current count. 1h (R/W) = Enables the counter

30.5.4.2 PRU_ICSS_IEP_TMR_GLB_STS Register (Offset = 4h) [reset = 0h]

PRU_ICSS_IEP_TMR_GLB_STS is shown in Figure 30-200 and described in Table 30-191.

Return to Summary Table.

GLOBAL STATUS

Figure 30-200. PRU_ICSS_IEP_TMR_GLB_STS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CNT_OVF
R-0h							R/W1C-0h

Table 30-191. PRU_ICSS_IEP_TMR_GLB_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	CNT_OVF	R/W1C	0h	Counter overflow status. 0h (R/W) = No overflow 1h (R/W) = Overflow occurred

30.5.4.3 PRU_ICSS_IEP_TMR_COMPEN Register (Offset = 8h) [reset = 0h]

PRU_ICSS_IEP_TMR_COMPEN is shown in Figure 30-201 and described in Table 30-192.

Return to Summary Table.

COMPENSATION

Figure 30-201. PRU_ICSS_IEP_TMR_COMPEN Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="24">COMPEN_CNT</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="24">R/W-0h</td></tr>
</table>

Table 30-192. PRU_ICSS_IEP_TMR_COMPEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	COMPEN_CNT	R/W	0h	Compensation counter. Read returns the current compen_cnt value. 0: Compensation is disabled and counter will increment by DEFAULT_INC. Compensation is enabled until COMPEN_CNT decrements to 0. The COMPEN_CNT value decrements on every iep_clk/ocp_clk cycle. When COMPEN_CNT is greater than 0, then count value increments by CMP_INC.

30.5.4.4 PRU_ICSS_IEP_TMR_CNT Register (Offset = Ch) [reset = 0h]

PRU_ICSS_IEP_TMR_CNT is shown in Figure 30-202 and described in Table 30-193.

Return to Summary Table.

COUNTER

Figure 30-202. PRU_ICSS_IEP_TMR_CNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT																															
R/W1C-0h																															

Table 30-193. PRU_ICSS_IEP_TMR_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W1C	0h	32-bit count value. Increments by (DEFAULT_INC or CMP_INC) on every positive edge of iep_clk (200MHz or ocp_clk).

30.5.4.5 PRU_ICSS_IEP_TMR_CAP_CFG Register (Offset = 10h) [reset = 0003FC00h]

PRU_ICSS_IEP_TMR_CAP_CFG is shown in Figure 30-203 and described in Table 30-194.

Return to Summary Table.

CAPTURE CONFIGURE

Figure 30-203. PRU_ICSS_IEP_TMR_CAP_CFG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="6">RESERVED</td><td colspan="2">CAP_ASYNC_EN</td></tr>
<tr><td colspan="6">R/W-</td><td colspan="2">R/W-FFh</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="6">CAP_ASYNC_EN</td><td>CAP7F_1ST_EVENT_EN</td><td>CAP7R_1ST_EVENT_EN</td></tr>
<tr><td colspan="6">R/W-FFh</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>CAP6F_1ST_EVENT_EN</td><td>CAP6R_1ST_EVENT_EN</td><td colspan="6">CAP_1ST_EVENT_EN</td></tr>
<tr><td>R/W-0h</td><td>R/W-0h</td><td colspan="6">R/W-0h</td></tr>
</table>

Table 30-194. PRU_ICSS_IEP_TMR_CAP_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W		
17-10	CAP_ASYNC_EN	R/W	FFh	Synchronization of the capture inputs to the iep_clk/ocp_clk enable. Note if input capture signal is asynchronous to iep_clk, enabling synchronization will cause the capture contents to be invalid. CAP_ASYNC_EN[n] maps to CAPR[n]. 0h (R/W) = Disable synchronization 1h (R/W) = Enable synchronization
9	CAP7F_1ST_EVENT_EN	R/W	0h	Capture 1st Event Enable for CAPF7 0h (R/W) = Continues mode. The capture status is not set when events occur. 1h (R/W) = First Event mode. The capture status is set when the first event occurs and must be cleared before new data will fill buffer. Time value is captured when first event occurs and held until time is read.
8	CAP7R_1ST_EVENT_EN	R/W	0h	Capture 1st Event Enable for CAPR7 0h (R/W) = Continues mode. The capture status is not set when events occur. 1h (R/W) = First Event mode. The capture status is set when the first event occurs and must be cleared before new data will fill buffer. Time value is captured when first event occurs and held until time is read.
7	CAP6F_1ST_EVENT_EN	R/W	0h	Capture 1st Event Enable for CAPF6 0h (R/W) = Continues mode. The capture status is not set when events occur. 1h (R/W) = First Event mode. The capture status is set when the first event occurs and must be cleared before new data will fill buffer. Time value is captured when first event occurs and held until time is read.

Table 30-194. PRU_ICSS_IEP_TMR_CAP_CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	CAP6R_1ST_EVENT_EN	R/W	0h	Capture 1st Event Enable for CAPR6 0h (R/W) = Continues mode. The capture status is not set when events occur. 1h (R/W) = First Event mode. The capture status is set when the first event occurs and must be cleared before new data will fill buffer. Time value is captured when first event occurs and held until time is read.
5-0	CAP_1ST_EVENT_EN	R/W	0h	Capture 1st Event Enable for registers. CAP_1ST_EVENT_EN[n] maps to CAPR[n]. 0h (R/W) = Continues mode. The capture status is not set when events occur. 1h (R/W) = First Event mode. The capture status is set when the first event occurs and must be cleared before new data will fill buffer. Time value is captured when first event occurs and held until time is read.

30.5.4.6 PRU_ICSS_IEP_TMR_CAP_STS Register (Offset = 14h) [reset = X]

PRU_ICSS_IEP_TMR_CAP_STS is shown in Figure 30-204 and described in Table 30-195.

Return to Summary Table.

CAPTURE STATUS CONFIGURE. Note: Capture will always occur as long as it is enabled, if enabled the user cannot tell if 2 events occurred.

Figure 30-204. PRU_ICSS_IEP_TMR_CAP_STS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">CAP_RAW</td></tr>
<tr><td colspan="8">R-X</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="5">RESERVED</td><td>CAP_VALID</td><td>CAPF7_VALID</td><td>CAPR7_VALID</td></tr>
<tr><td colspan="5">R-0h</td><td>R-0h</td><td>RC-0h</td><td>RC-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>CAPF6_VALID</td><td>CAPR6_VALID</td><td colspan="6">CAPR_VALID</td></tr>
<tr><td>RC-0h</td><td>RC-0h</td><td colspan="6">RC-0h</td></tr>
</table>

Table 30-195. PRU_ICSS_IEP_TMR_CAP_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-16	CAP_RAW	R	X	Raw/Current status bit for each of the capture registers, where CAP_RAW[n] maps to CAPR[n]. 0h (R) = Current state is low 1h (R) = Current state is high
15-11	RESERVED	R	0h	
10	CAP_VALID	R	0h	Valid status for capture function. Reflects the ORed result from CAP_STATUS[9 to 0]. 0h (R) = No Hit for any capture event, i.e., there are all 0 in CAP_STATUS [9 to 0]. 1h (R) = Hit for 1 or more captures events is pending, i.e., there has at least one value equal to 1 in CAP_STATUS[9 to 0].
9	CAPF7_VALID	RC	0h	Valid status for CAPF7 (fall). 0h (R) = No Hit, no capture event occurred 1h (R) = Hit, capture event occured
8	CAPR7_VALID	RC	0h	Valid status for CAPR7 (rise). 0h (R) = No Hit, no capture event occurred 1h (R) = Hit, capture event occured
7	CAPF6_VALID	RC	0h	Valid status for CAPF6 (fall). 0h (R) = No Hit, no capture event occurred 1h (R) = Hit, capture event occured
6	CAPR6_VALID	RC	0h	Valid status for CAPR6 (rise). 0h (R) = No Hit, no capture event occurred 1h (R) = Hit, capture event occured
5-0	CAPR_VALID	RC	0h	Valid status bit for each compare register, where CAPR_VALID[n] maps to CAPR[n] (rise). 0h (R) = No Hit, no capture event occurred 1h (R/W) = Hit, capture event occurred

30.5.4.7 PRU_ICSS_IEP_TMR_CAPR0 Register (Offset = 18h) [reset = 0h]

PRU_ICSS_IEP_TMR_CAPR0 is shown in Figure 30-205 and described in Table 30-196.

Return to Summary Table.

CAPTURE RISE0

Figure 30-205. PRU_ICSS_IEP_TMR_CAPR0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">CAPR0</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 30-196. PRU_ICSS_IEP_TMR_CAPR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPR0	R	0h	Value captured for CAPR0 event

30.5.4.8 PRU_ICSS_IEP_TMR_CAPR1 Register (Offset = 1Ch) [reset = 0h]

PRU_ICSS_IEP_TMR_CAPR1 is shown in Figure 30-206 and described in Table 30-197.

Return to Summary Table.

CAPTURE RISE1

Figure 30-206. PRU_ICSS_IEP_TMR_CAPR1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">CAPR1</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 30-197. PRU_ICSS_IEP_TMR_CAPR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPR1	R	0h	Value captured for CAPR1 event

30.5.4.9 PRU_ICSS_IEP_TMR_CAPR2 Register (Offset = 20h) [reset = 0h]

PRU_ICSS_IEP_TMR_CAPR2 is shown in Figure 30-207 and described in Table 30-198.

Return to Summary Table.

CAPTURE RISE2

Figure 30-207. PRU_ICSS_IEP_TMR_CAPR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPR2																															
R-0h																															

Table 30-198. PRU_ICSS_IEP_TMR_CAPR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPR2	R	0h	Value captured for CAPR2 event

30.5.4.10 PRU_ICSS_IEP_TMR_CAPR3 Register (Offset = 24h) [reset = 0h]

PRU_ICSS_IEP_TMR_CAPR3 is shown in Figure 30-208 and described in Table 30-199.

Return to Summary Table.

CAPTURE RISE3

Figure 30-208. PRU_ICSS_IEP_TMR_CAPR3 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">CAPR3</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 30-199. PRU_ICSS_IEP_TMR_CAPR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPR3	R	0h	Value captured for CAPR3 event

30.5.4.11 PRU_ICSS_IEP_TMR_CAPR4 Register (Offset = 28h) [reset = 0h]

PRU_ICSS_IEP_TMR_CAPR4 is shown in Figure 30-209 and described in Table 30-200.

Return to Summary Table.

CAPTURE RISE4

Figure 30-209. PRU_ICSS_IEP_TMR_CAPR4 Register

31-0
CAPR4
R-0h

Table 30-200. PRU_ICSS_IEP_TMR_CAPR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPR4	R	0h	Value captured for CAPR4 event

30.5.4.12 PRU_ICSS_IEP_TMR_CAPR5 Register (Offset = 2Ch) [reset = 0h]

PRU_ICSS_IEP_TMR_CAPR5 is shown in Figure 30-210 and described in Table 30-201.

Return to Summary Table.

CAPTURE RISE5

Figure 30-210. PRU_ICSS_IEP_TMR_CAPR5 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">CAPR5</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 30-201. PRU_ICSS_IEP_TMR_CAPR5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPR5	R	0h	Value captured for CAPR5 event

30.5.4.13 PRU_ICSS_IEP_TMR_CAPR6 Register (Offset = 30h) [reset = 0h]

PRU_ICSS_IEP_TMR_CAPR6 is shown in Figure 30-211 and described in Table 30-202.

Return to Summary Table.

CAPTURE RISE6

Figure 30-211. PRU_ICSS_IEP_TMR_CAPR6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPR6																															
R-0h																															

Table 30-202. PRU_ICSS_IEP_TMR_CAPR6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPR6	R	0h	Value captured for CAPR6 event

30.5.4.14 PRU_ICSS_IEP_TMR_CAPF6 Register (Offset = 34h) [reset = 0h]

PRU_ICSS_IEP_TMR_CAPF6 is shown in Figure 30-212 and described in Table 30-203.

Return to Summary Table.

CAPTURE FALL6

Figure 30-212. PRU_ICSS_IEP_TMR_CAPF6 Register

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="32">CAPF6</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 30-203. PRU_ICSS_IEP_TMR_CAPF6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPF6	R	0h	Value captured for CAPF6 event

30.5.4.15 PRU_ICSS_IEP_TMR_CAPR7 Register (Offset = 38h) [reset = 0h]

PRU_ICSS_IEP_TMR_CAPR7 is shown in Figure 30-213 and described in Table 30-204.

Return to Summary Table.

CAPTURE RISE7

Figure 30-213. PRU_ICSS_IEP_TMR_CAPR7 Register

31–0
CAPR7
R-0h

Table 30-204. PRU_ICSS_IEP_TMR_CAPR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPR7	R	0h	Value captured for CAPR7 event

30.5.4.16 PRU_ICSS_IEP_TMR_CAPF7 Register (Offset = 3Ch) [reset = 0h]

PRU_ICSS_IEP_TMR_CAPF7 is shown in Figure 30-214 and described in Table 30-205.

Return to Summary Table.

CAPTURE FALL7

Figure 30-214. PRU_ICSS_IEP_TMR_CAPF7 Register

31–0
CAPF7
R-0h

Table 30-205. PRU_ICSS_IEP_TMR_CAPF7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPF7	R	0h	Value captured for CAPF7 event

30.5.4.17 PRU_ICSS_IEP_TMR_CMP_CFG Register (Offset = 40h) [reset = 0h]

PRU_ICSS_IEP_TMR_CMP_CFG is shown in Figure 30-215 and described in Table 30-206.

Return to Summary Table.

COMPARE CONFIGURE

Figure 30-215. PRU_ICSS_IEP_TMR_CMP_CFG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="7">RESERVED</td><td>CMP_EN</td></tr>
<tr><td colspan="7">R-0h</td><td>R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">CMP_EN</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="7">CMP_EN</td><td>CMP0_RST_CNT_EN</td></tr>
<tr><td colspan="7">R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 30-206. PRU_ICSS_IEP_TMR_CMP_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16-1	CMP_EN	R/W	0h	Compare registers enable, where CMP_EN[0] maps to CMP[0]. 0h (R/W) = Disables event 1h (R/W) = Enables event
0	CMP0_RST_CNT_EN	R/W	0h	Counter reset enable. 0h (R/W) = Disable 1h (R/W) = Enable the reset of the counter if a CMP0 event occurs

30.5.4.18 PRU_ICSS_IEP_TMR_CMP_STS Register (Offset = 44h) [reset = 0h]

PRU_ICSS_IEP_TMR_CMP_STS is shown in Figure 30-216 and described in Table 30-207.

Return to Summary Table.

COMPARE STATUS

Figure 30-216. PRU_ICSS_IEP_TMR_CMP_STS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">CMP_HIT</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">R/W1C-0h</td></tr>
</table>

Table 30-207. PRU_ICSS_IEP_TMR_CMP_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	CMP_HIT	R/W1C	0h	Status bit for each of the compare registers, where CMP_HIT[n] maps to CMP[n]. "Match" indicates the current counter is greater than or equal to the compare value. Note it is the firmware's responsibility to handle the IEP overflow. 0h (R/W) = Match has not occurred. 1h (R/W) = Match occurred. The associated hardware event signal will assert and remain high until the status is cleared.

30.5.4.19 PRU_ICSS_IEP_TMR_CMP0 Register (Offset = 48h) [reset = 0h]

PRU_ICSS_IEP_TMR_CMP0 is shown in Figure 30-217 and described in Table 30-208.

Return to Summary Table.

COMPARE0

Figure 30-217. PRU_ICSS_IEP_TMR_CMP0 Register

31-0
CMP0
R/W-0h

Table 30-208. PRU_ICSS_IEP_TMR_CMP0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP0	R/W	0h	Compare 0 value

30.5.4.20 PRU_ICSS_IEP_TMR_CMP1 Register (Offset = 4Ch) [reset = 0h]

PRU_ICSS_IEP_TMR_CMP1 is shown in Figure 30-218 and described in Table 30-209.

Return to Summary Table.

COMPARE1

Figure 30-218. PRU_ICSS_IEP_TMR_CMP1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP1																															
R/W-0h																															

Table 30-209. PRU_ICSS_IEP_TMR_CMP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP1	R/W	0h	Compare 1 value

30.5.4.21 PRU_ICSS_IEP_TMR_CMP2 Register (Offset = 50h) [reset = 0h]

PRU_ICSS_IEP_TMR_CMP2 is shown in Figure 30-219 and described in Table 30-210.

Return to Summary Table.

COMPARE2

Figure 30-219. PRU_ICSS_IEP_TMR_CMP2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP2																															
R/W-0h																															

Table 30-210. PRU_ICSS_IEP_TMR_CMP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP2	R/W	0h	Compare 2 value

30.5.4.22 PRU_ICSS_IEP_TMR_CMP3 Register (Offset = 54h) [reset = 0h]

PRU_ICSS_IEP_TMR_CMP3 is shown in Figure 30-220 and described in Table 30-211.

Return to Summary Table.

COMPARE3

Figure 30-220. PRU_ICSS_IEP_TMR_CMP3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP3																															
R/W-0h																															

Table 30-211. PRU_ICSS_IEP_TMR_CMP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP3	R/W	0h	Compare 3 value

30.5.4.23 PRU_ICSS_IEP_TMR_CMP4 Register (Offset = 58h) [reset = 0h]

PRU_ICSS_IEP_TMR_CMP4 is shown in Figure 30-221 and described in Table 30-212.

Return to Summary Table.

COMPARE4

Figure 30-221. PRU_ICSS_IEP_TMR_CMP4 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">CMP4</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 30-212. PRU_ICSS_IEP_TMR_CMP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP4	R/W	0h	Compare 4 value

30.5.4.24 PRU_ICSS_IEP_TMR_CMP5 Register (Offset = 5Ch) [reset = 0h]

PRU_ICSS_IEP_TMR_CMP5 is shown in Figure 30-222 and described in Table 30-213.

Return to Summary Table.

COMPARE5

Figure 30-222. PRU_ICSS_IEP_TMR_CMP5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP5																															
R/W-0h																															

Table 30-213. PRU_ICSS_IEP_TMR_CMP5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP5	R/W	0h	Compare 5 value

30.5.4.25 PRU_ICSS_IEP_TMR_CMP6 Register (Offset = 60h) [reset = 0h]

PRU_ICSS_IEP_TMR_CMP6 is shown in Figure 30-223 and described in Table 30-214.

Return to Summary Table.

COMPARE6

Figure 30-223. PRU_ICSS_IEP_TMR_CMP6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP6																															
R/W-0h																															

Table 30-214. PRU_ICSS_IEP_TMR_CMP6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP6	R/W	0h	Compare 6 value

30.5.4.26 PRU_ICSS_IEP_TMR_CMP7 Register (Offset = 64h) [reset = 0h]

PRU_ICSS_IEP_TMR_CMP7 is shown in Figure 30-224 and described in Table 30-215.

Return to Summary Table.

COMPARE7

Figure 30-224. PRU_ICSS_IEP_TMR_CMP7 Register

31-0
CMP7
R/W-0h

Table 30-215. PRU_ICSS_IEP_TMR_CMP7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP7	R/W	0h	Compare 7 value

30.5.4.27 PRU_ICSS_IEP_TMR_RXIPG0 Register (Offset = 80h) [reset = FFFF0000h]

PRU_ICSS_IEP_TMR_RXIPG0 is shown in Figure 30-225 and described in Table 30-216.

Return to Summary Table.

RX InterPackage Gap (IPG) 0

Figure 30-225. PRU_ICSS_IEP_TMR_RXIPG0 Register

31-16	15-0
RX_MIN_IPG	RX_IPG
R/WtoReset-FFFFh	R-0h

Table 30-216. PRU_ICSS_IEP_TMR_RXIPG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RX_MIN_IPG	R/WtoReset	FFFFh	Defines the minimum number of iep_clk/ocp_clk cycles that is RX_DV is sampled low. It stores the smallest RX_IPG duration. It can be read at any time and gets updated after RX_IPG is updated, if RX_MIN_IPG is greater than RX_IPG.
15-0	RX_IPG	R	0h	Records the current number of iep_clk/ocp_clk cycles RX_DV is sampled low. Value is updated after RX_DV transitions from low to high. It will saturate at 0xffff.

30.5.4.28 PRU_ICSS_IEP_TMR_RXIPG1 Register (Offset = 84h) [reset = FFFF0000h]

PRU_ICSS_IEP_TMR_RXIPG1 is shown in Figure 30-226 and described in Table 30-217.

Return to Summary Table.

RX InterPackage Gap (IPG) 1

Figure 30-226. PRU_ICSS_IEP_TMR_RXIPG1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RX_MIN_IPG</td><td colspan="16">RX_IPG</td></tr>
<tr><td colspan="16">R/WtoReset-FFFFh</td><td colspan="16">R-0h</td></tr>
</table>

Table 30-217. PRU_ICSS_IEP_TMR_RXIPG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RX_MIN_IPG	R/WtoReset	FFFFh	Defines the minimum number of iep_clk/ocp_clk cycles that is RX_DV is sampled low. It stores the smallest RX_IPG duration. It can be read at any time and gets updated after RX_IPG is updated, if RX_MIN_IPG is greater than RX_IPG.
15-0	RX_IPG	R	0h	Records the current number of iep_clk/ocp_clk cycles RX_DV is sampled low. Value is updated after RX_DV transitions from low to high. It will saturate at 0xffff.

30.5.4.29 PRU_ICSS_IEP_TMR_CMP8 Register (Offset = 88h) [reset = 0h]

PRU_ICSS_IEP_TMR_CMP8 is shown in Figure 30-227 and described in Table 30-218.

Return to Summary Table.

COMPARE8

Figure 30-227. PRU_ICSS_IEP_TMR_CMP8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP8																															
R/W-0h																															

Table 30-218. PRU_ICSS_IEP_TMR_CMP8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP8	R/W	0h	Compare 8 value

30.5.4.30 PRU_ICSS_IEP_TMR_CMP9 Register (Offset = 8Ch) [reset = 0h]

PRU_ICSS_IEP_TMR_CMP9 is shown in Figure 30-228 and described in Table 30-219.

Return to Summary Table.

COMPARE9

Figure 30-228. PRU_ICSS_IEP_TMR_CMP9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP9																															
R/W-0h																															

Table 30-219. PRU_ICSS_IEP_TMR_CMP9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP9	R/W	0h	Compare 9 value

30.5.4.31 PRU_ICSS_IEP_TMR_CMP10 Register (Offset = 90h) [reset = 0h]

PRU_ICSS_IEP_TMR_CMP10 is shown in Figure 30-229 and described in Table 30-220.

Return to Summary Table.

COMPARE10

Figure 30-229. PRU_ICSS_IEP_TMR_CMP10 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP10																															
R/W-0h																															

Table 30-220. PRU_ICSS_IEP_TMR_CMP10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP10	R/W	0h	Compare 10 value

30.5.4.32 PRU_ICSS_IEP_TMR_CMP11 Register (Offset = 94h) [reset = 0h]

PRU_ICSS_IEP_TMR_CMP11 is shown in Figure 30-230 and described in Table 30-221.

Return to Summary Table.

COMPARE11

Figure 30-230. PRU_ICSS_IEP_TMR_CMP11 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP11																															
R/W-0h																															

Table 30-221. PRU_ICSS_IEP_TMR_CMP11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP11	R/W	0h	Compare 11 value

30.5.4.33 PRU_ICSS_IEP_TMR_CMP12 Register (Offset = 98h) [reset = 0h]

PRU_ICSS_IEP_TMR_CMP12 is shown in Figure 30-231 and described in Table 30-222.

Return to Summary Table.

COMPARE12

Figure 30-231. PRU_ICSS_IEP_TMR_CMP12 Register

31-0
CMP12
R/W-0h

Table 30-222. PRU_ICSS_IEP_TMR_CMP12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP12	R/W	0h	Compare 12 value

30.5.4.34 PRU_ICSS_IEP_TMR_CMP13 Register (Offset = 9Ch) [reset = 0h]

PRU_ICSS_IEP_TMR_CMP13 is shown in Figure 30-232 and described in Table 30-223.

Return to Summary Table.

COMPARE13

Figure 30-232. PRU_ICSS_IEP_TMR_CMP13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP13																															
R/W-0h																															

Table 30-223. PRU_ICSS_IEP_TMR_CMP13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP13	R/W	0h	Compare 13 value

30.5.4.35 PRU_ICSS_IEP_TMR_CMP14 Register (Offset = A0h) [reset = 0h]

PRU_ICSS_IEP_TMR_CMP14 is shown in Figure 30-233 and described in Table 30-224.

Return to Summary Table.

COMPARE14

Figure 30-233. PRU_ICSS_IEP_TMR_CMP14 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP14																															
R/W-0h																															

Table 30-224. PRU_ICSS_IEP_TMR_CMP14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP14	R/W	0h	Compare 14 value

30.5.4.36 PRU_ICSS_IEP_TMR_CMP15 Register (Offset = A4h) [reset = 0h]

PRU_ICSS_IEP_TMR_CMP15 is shown in Figure 30-234 and described in Table 30-225.

Return to Summary Table.

COMPARE15

Figure 30-234. PRU_ICSS_IEP_TMR_CMP15 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP15																															
R/W-0h																															

Table 30-225. PRU_ICSS_IEP_TMR_CMP15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP15	R/W	0h	Compare 15 value

30.5.4.37 PRU_ICSS_IEP_TMR_CNT_RST Register (Offset = A8h) [reset = 0h]

PRU_ICSS_IEP_TMR_CNT_RST is shown in Figure 30-235 and described in Table 30-226.

Return to Summary Table.

COUNT RESET

Figure 30-235. PRU_ICSS_IEP_TMR_CNT_RST Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
RESET_VAL
R/W-0h

Table 30-226. PRU_ICSS_IEP_TMR_CNT_RST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESET_VAL	R/W	0h	This enables SW to define the reset state of the Master counter when it gets reset due to the following 3 possible events (if enabled): CMP0 event PWM0_SYNC_OUT event PWM3_SYNC_OUT event Note it should be in increments of the default_inc (default state is 5, or 0xA). It should be in incremeants of the default_inc, default state is 5 For example, 0x0000_000A

30.5.4.38 PRU_ICSS_IEP_TMR_PWM Register (Offset = ACh) [reset = 0h]

PRU_ICSS_IEP_TMR_PWM is shown in Figure 30-236 and described in Table 30-227.

Return to Summary Table.

PWM

Figure 30-236. PRU_ICSS_IEP_TMR_PWM Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				PWM3_HIT	PWM3_RST_CNT_EN	PWM0_HIT	PWM0_RST_CNT_EN
R-0h				R/W1C-0h	R/W-0h	R/W1C-0h	R/W-0h

Table 30-227. PRU_ICSS_IEP_TMR_PWM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	PWM3_HIT	R/W1C	0h	Raw Status bit of pwm3_sync_out event. 1h (W) = Clear pwm0_sync_out event 1h (R) = pwm3_sync_out event occurred
2	PWM3_RST_CNT_EN	R/W	0h	Enable the reset of the counter by a pwm3_sync_out event. 0h (R/W) = Disable 1h (R/W) = Enable the reset of the counter if a pwm3_sync_out event occurs
1	PWM0_HIT	R/W1C	0h	Raw Status bit of pwm0_sync_out event. 1h (W) = Clear pwm0_sync_out event 1h (R) = pwm0_sync_out event occurred
0	PWM0_RST_CNT_EN	R/W	0h	Enable the reset of the counter by a pwm0_sync_out event. 0h (R/W) = Disable 1h (R/W) = Enable the reset of the counter if a pwm0_sync_out event occurs

30.5.4.39 PRU_ICSS_IEP_SYNC_CTRL Register (Offset = 100h) [reset = 0h]

PRU_ICSS_IEP_SYNC_CTRL is shown in Figure 30-237 and described in Table 30-228.

Return to Summary Table.

SYNC GENERATION CONTROL

Figure 30-237. PRU_ICSS_IEP_SYNC_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							SYNC1_IND_EN
R-0h							R/W-0h
7	6	5	4	3	2	1	0
SYNC1_CYCLIC_EN	SYNC1_ACK_EN	SYNC0_CYCLIC_EN	SYNC0_ACK_EN	RESERVED	SYNC1_EN	SYNC0_EN	SYNC_EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h

Table 30-228. PRU_ICSS_IEP_SYNC_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	SYNC1_IND_EN	R/W	0h	SYNC1 independent mode enable. Independent mode means the SYNC1 signal can be different from SYNC0. 0: Dependent mode 1: Independent mode
7	SYNC1_CYCLIC_EN	R/W	0h	SYNC1 single shot or cyclic/auto generation mode enable 0: Disable, single shot mode 1: Enable, cyclic generation mode
6	SYNC1_ACK_EN	R/W	0h	SYNC1 acknowledgement mode enable 0: Disable, SYNC1 will go low after pulse width is met. 1: Enable, SYNC1 will remain asserted until receiving software acknowledges by reading SYNC1_STATUS which clears on read.
5	SYNC0_CYCLIC_EN	R/W	0h	SYNC0 single shot or cyclic/auto generation mode enable 0: Disable, single shot mode 1: Enable, cyclic generation mode
4	SYNC0_ACK_EN	R/W	0h	SYNC0 acknowledgement mode enable 0: Disable, SYNC0 will go low after pulse width is met. 1: Enable, SYNC0 will remain asserted until receiving software acknowledges by reading SYNC1_STATUS which clears on read.
3	RESERVED	R	0h	
2	SYNC1_EN	R/W	0h	SYNC1 generation enable 0: Disable 1: Enable
1	SYNC0_EN	R/W	0h	SYNC0 generation enable 0: Disable 1: Enable
0	SYNC_EN	R/W	0h	SYNC generation enable 0: Disable the generation and clocking of SYNC0 and SYNC1 logic 1: Enables SYNC0 and SYNC1 generation

30.5.4.40 PRU_ICSS_IEP_SYNC_FIRST_STAT Register (Offset = 104h) [reset = 0h]

PRU_ICSS_IEP_SYNC_FIRST_STAT is shown in Figure 30-238 and described in Table 30-229.

Return to Summary Table.

SYNC GENERATION FIRST EVENT STATUS

Figure 30-238. PRU_ICSS_IEP_SYNC_FIRST_STAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						FIRST_SYNC1	FIRST_SYNC0
R-0h						R-0h	R-0h

Table 30-229. PRU_ICSS_IEP_SYNC_FIRST_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	FIRST_SYNC1	R	0h	SYNC1 First Event status. This bit is cleared when SYNC1_EN = 0. 0: Not occurred 1: Occurred
0	FIRST_SYNC0	R	0h	SYNC0 First Event status. This bit is cleared when SYNC0_EN = 0. 0: Not occurred 1: Occurred

30.5.4.41 PRU_ICSS_IEP_SYNC0_STAT Register (Offset = 108h) [reset = 0h]

PRU_ICSS_IEP_SYNC0_STAT is shown in Figure 30-239 and described in Table 30-230.

Return to Summary Table.

SYNC0 STATUS

Figure 30-239. PRU_ICSS_IEP_SYNC0_STAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							SYNC0_PEND
R-0h							R/W1C-0h

Table 30-230. PRU_ICSS_IEP_SYNC0_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	SYNC0_PEND	R/W1C	0h	SYNC0 pending state. 0: Not pending 1: Pending or the SYNC0_PEND has occurred when SYNC0_ACK_EN = 0 (Disable).

30.5.4.42 PRU_ICSS_IEP_SYNC1_STAT Register (Offset = 10Ch) [reset = 0h]

PRU_ICSS_IEP_SYNC1_STAT is shown in Figure 30-240 and described in Table 30-231.

Return to Summary Table.

SYNC1 STATUS

Figure 30-240. PRU_ICSS_IEP_SYNC1_STAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							SYNC1_PEND
R-0h							R/W1C-0h

Table 30-231. PRU_ICSS_IEP_SYNC1_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	SYNC1_PEND	R/W1C	0h	SYNC1 pending state. 0: Not pending 1: Pending or the SYNC1_PEND has occurred when SYNC1_ACK_EN = 0 (Disable).

30.5.4.43 PRU_ICSS_IEP_SYNC_PWIDTH Register (Offset = 110h) [reset = 0h]

PRU_ICSS_IEP_SYNC_PWIDTH is shown in Figure 30-241 and described in Table 30-232.

Return to Summary Table.

SYNC PULSE WIDTH CONFIGURE

Figure 30-241. PRU_ICSS_IEP_SYNC_PWIDTH Register

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="32">SYNC_HPW</td></tr>
<tr><td colspan="32">R/W-0h</td></tr>
</table>

Table 30-232. PRU_ICSS_IEP_SYNC_PWIDTH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SYNC_HPW	R/W	0h	Defines the number of clock cycles SYNC0/1 will be high. Note if SYNC0/1 is disabled during pulse width time (that is, SYNC_CTRL[SYNC0_EN | SYNC1_EN | SYNC_EN] = 0), the ongoing pulse will be terminated. 0: 1 clock cycle. 1: 2 clock cycles. N: N+1 clock cycles.

30.5.4.44 PRU_ICSS_IEP_SYNC0_PERIOD Register (Offset = 114h) [reset = 1h]

PRU_ICSS_IEP_SYNC0_PERIOD is shown in Figure 30-242 and described in Table 30-233.

Return to Summary Table.

SYNC PERIOD CONFIGURE

Figure 30-242. PRU_ICSS_IEP_SYNC0_PERIOD Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">SYNC0_PERIOD</td></tr>
<tr><td colspan="32">R/W-1h</td></tr>
</table>

Table 30-233. PRU_ICSS_IEP_SYNC0_PERIOD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SYNC0_PERIOD	R/W	1h	Defines the period between the rising edges of SYNC0. 0: reserved. 1: 2 clock cycles period. N: N+1 clock cycles period.

30.5.4.45 PRU_ICSS_IEP_SYNC1_DELAY Register (Offset = 118h) [reset = 0h]

PRU_ICSS_IEP_SYNC1_DELAY is shown in Figure 30-243 and described in Table 30-234.

Return to Summary Table.

SYNC CTRL

Figure 30-243. PRU_ICSS_IEP_SYNC1_DELAY Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNC1_DELAY																															
R/W-0h																															

Table 30-234. PRU_ICSS_IEP_SYNC1_DELAY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SYNC1_DELAY	R/W	0h	When SYNC1_IND_EN = 0, defines number of clock cycles from the start of SYNC0 to the start of SYNC1. Note this is the delay before the start of SYNC1. 0: No delay. 1: 1 clock cycle delay. N: N clock cycles delay. When SYNC1_IND_EN = 1, defines the period between the rising edges of SYNC1. 0: reserved. 1: 2 clock cycles period. N: N+1 clock cycles period.

30.5.4.46 PRU_ICSS_IEP_SYNC_START Register (Offset = 11Ch) [reset = 0h]

PRU_ICSS_IEP_SYNC_START is shown in Figure 30-244 and described in Table 30-235.

Return to Summary Table.

SYNC START CONFIGURE

Figure 30-244. PRU_ICSS_IEP_SYNC_START Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNC_START																															
R/W-0h																															

Table 30-235. PRU_ICSS_IEP_SYNC_START Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SYNC_START	R/W	0h	Defines the start time after the activation event. 0: 1 clock cycle delay. 1: 2 clock cycles delay. N: N+1 clock cycles delay.

30.5.4.47 PRU_ICSS_IEP_WD_PREDIV Register (Offset = 200h) [reset = 4E20h]

PRU_ICSS_IEP_WD_PREDIV is shown in Figure 30-245 and described in Table 30-236.

Return to Summary Table.

WATCHDOG PRE-DIVIDER

Figure 30-245. PRU_ICSS_IEP_WD_PREDIV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PRE_DIV															
R-0h																R/WC-4E20h															

Table 30-236. PRU_ICSS_IEP_WD_PREDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	PRE_DIV	R/WC	4E20h	Defines the number of iep_clk cycles per WD clock event. Note that the WD clock is a free-running clock. The value 0x4e20 (or 20000) generates a rate of 100 us if iep_clk is 200 MHz. seconds/(WD event) = (clock cycles per WD event)/(clock cycles per second) = 20000/(200 x [10]^6) = 100µs

30.5.4.48 PRU_ICSS_IEP_PDI_WD_TIM Register (Offset = 204h) [reset = 3E8h]

PRU_ICSS_IEP_PDI_WD_TIM is shown in Figure 30-246 and described in Table 30-237.

Return to Summary Table.

PDI WATCHDOG TIMER CONFIGURE

Figure 30-246. PRU_ICSS_IEP_PDI_WD_TIM Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RESERVED</td><td colspan="16">PDI_WD_TIME</td></tr>
<tr><td colspan="16">R-0h</td><td colspan="16">R/WtoReset-3E8h</td></tr>
</table>

Table 30-237. PRU_ICSS_IEP_PDI_WD_TIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	PDI_WD_TIME	R/WtoReset	3E8h	Defines the number of WD ticks (or increments) for PDI WD, that is, the number of WD increments. If PRE_DIV is set to 100µs, then the value 0x03e8 (or 1000) provides a rate of 100ms. Read returns the current count. Counter is reset by software write to register or when Digital Data In capture occurs. WD is disabled if WD time is set to 0x0. Note when an expiration event occurs, the expiration counter (PDI_EXP_CNT) increments and status (PDI_WD_STAT) clears.

30.5.4.49 PRU_ICSS_IEP_PD_WD_TIM Register (Offset = 208h) [reset = 3E8h]

PRU_ICSS_IEP_PD_WD_TIM is shown in Figure 30-247 and described in Table 30-238.

Return to Summary Table.

PD WATCHDOG TIMER CONFIGURE

Figure 30-247. PRU_ICSS_IEP_PD_WD_TIM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PD_WD_TIME															
R-0h																R/WtoReset-3E8h															

Table 30-238. PRU_ICSS_IEP_PD_WD_TIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	PD_WD_TIME	R/WtoReset	3E8h	Defines the number of WD ticks (or increments) for PDI WD, that is, the number of WD increments. If PRE_DIV is set to 100µs, then 0x03e8 (or 1000) provides a rate of 100ms. Read returns the current count. Counter is reset by software write to register or every write access to Sync Managers with WD trigger enable bit set. WD is disabled if WD time is set to 0x0. Expiration actions: Increment expiration counter, clear status. Digital Data out forced to zero if pr[k]_edio_oe_ext = 1 and DIGIO_EXT.SW_DATA_OUT_UPDATE = 0.

30.5.4.50 PRU_ICSS_IEP_WD_STS Register (Offset = 20Ch) [reset = 00010001h]

PRU_ICSS_IEP_WD_STS is shown in Figure 30-248 and described in Table 30-239.

Return to Summary Table.

WATCHDOG STATUS

Figure 30-248. PRU_ICSS_IEP_WD_STS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							PDI_WD_STAT
R-0h							R-1h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							PD_WD_STAT
R-0h							R-1h

Table 30-239. PRU_ICSS_IEP_WD_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	PDI_WD_STAT	R	1h	WD PDI status. 0: Expired (PDI_WD_EXP event generated) 1: Active or disabled
15-1	RESERVED	R	0h	
0	PD_WD_STAT	R	1h	WD PD status (triggered by Sync Mangers status). 0: Expired (PD_WD_EXP event generated) 1: Active or disabled

30.5.4.51 PRU_ICSS_IEP_WD_EXP_CNT Register (Offset = 210h) [reset = 0h]

PRU_ICSS_IEP_WD_EXP_CNT is shown in Figure 30-249 and described in Table 30-240.

Return to Summary Table.

WATCHDOG TIMER EXPIRATION COUNTER

Figure 30-249. PRU_ICSS_IEP_WD_EXP_CNT Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">PD_EXP_CNT</td><td colspan="8">PDI_EXP_CNT</td></tr>
<tr><td colspan="8">R/WC-0h</td><td colspan="8">R/WC-0h</td></tr>
</table>

Table 30-240. PRU_ICSS_IEP_WD_EXP_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	PD_EXP_CNT	R/WC	0h	WD PD expiration counter. Counter increments on every PD time out and stops at 0xff.
7-0	PDI_EXP_CNT	R/WC	0h	WD PDI expiration counter. Counter increments on every PDI time out and stops at 0xff.

30.5.4.52 PRU_ICSS_IEP_WD_CTRL Register (Offset = 214h) [reset = 0h]

PRU_ICSS_IEP_WD_CTRL is shown in Figure 30-250 and described in Table 30-241.

Return to Summary Table.

WATCHDOG CONTROL

Figure 30-250. PRU_ICSS_IEP_WD_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							PDI_WD_EN
R-0h							R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							PD_WD_EN
R-0h							R/W-0h

Table 30-241. PRU_ICSS_IEP_WD_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	PDI_WD_EN	R/W	0h	Enable WD PDI 0: Disable 1: Enable
15-1	RESERVED	R	0h	
0	PD_WD_EN	R/W	0h	Enable WD PD 0: Disable 1: Enable

30.5.4.53 PRU_ICSS_IEP_DIGIO_CTRL Register (Offset = 300h) [reset = 4h]

PRU_ICSS_IEP_DIGIO_CTRL is shown in Figure 30-251 and described in Table 30-242.

Return to Summary Table.

DIGITAL INPUT OUTPUT CONTROL

Figure 30-251. PRU_ICSS_IEP_DIGIO_CTRL Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">OUT_MODE</td><td colspan="2">IN_MODE</td><td>WD_MODE</td><td>BIDI_MODE</td><td>OUTVALID_MODE</td><td>OUTVALID_POL</td></tr>
<tr><td colspan="2">R/W-0h</td><td colspan="2">R/W-0h</td><td>R/W-0h</td><td>R-1h</td><td>R/W-0h</td><td>R-0h</td></tr>
</table>

Table 30-242. PRU_ICSS_IEP_DIGIO_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-6	OUT_MODE	R/W	0h	Defines event that triggers data out to be updated. Note if OUTVALID_MODE is set, then data out is forced to zero if a WD PD expiration occurs (PD_WD_EXP) from the WD block and pr[k]_edio_oe_ext = 1. 0h (R/W) = PRU0/1_RX_EOF 1h (R/W) = Reserved 2h (R/W) = DC SYNC0 event 3h (R/W) = DC SYNC1 event
5-4	IN_MODE	R/W	0h	Defines event that triggers data in to be sampled. 0h (R/W) = PRU0/1_RX_SOF 1h (R/W) = Rising edge of pr[k]_edio_latch_in 2h (R/W) = DC rising edge of SYNC0 event 3h (R/W) = DC rising edge of SYNC1 event
3	WD_MODE	R/W	0h	Defines watchdog behavior. 0h (R/W) = Outputs are reset immediately after watchdog expires 1h (R/W) = Outputs are reset with next output event that follows watchdog expiration
2	BIDI_MODE	R	1h	Defines the digital input/output direction. 0h (R) = Unidirectional mode: digital input/output direction of pins configured individually 1h (R) = Bidirectional mode: all I/O pins are bidirectional and direction configuration is ignored
1	OUTVALID_MODE	R/W	0h	Defines the outvalid mode behavior. 0h (R/W) = Output event signaling 1h (R/W) = Output data is updated if watchdog is triggered. Output data is forced to zero if PD_WD_EXP from the WD block and pr[k]_edio_oe_ext = 1
0	OUTVALID_POL	R	0h	Defines outvalid polarity. 0h (R) = Active high 1h (R) = Active low

30.5.4.54 PRU_ICSS_IEP_DIGIO_STATUS Register (Offset = 304h) [reset = 0h]

PRU_ICSS_IEP_DIGIO_STATUS is shown in Figure 30-252 and described in Table 30-243.

Return to Summary Table.

DIGITAL INPUT OUTPUT STATUS

Figure 30-252. PRU_ICSS_IEP_DIGIO_STATUS Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">DIGIO_STAT</td></tr>
<tr><td colspan="32">R-0h</td></tr>
</table>

Table 30-243. PRU_ICSS_IEP_DIGIO_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DIGIO_STAT	R	0h	Reserved

30.5.4.55 PRU_ICSS_IEP_DIGIO_DATA_IN Register (Offset = 308h) [reset = X]

PRU_ICSS_IEP_DIGIO_DATA_IN is shown in Figure 30-253 and described in Table 30-244.

Return to Summary Table.

DIGITAL DATA INPUT

Figure 30-253. PRU_ICSS_IEP_DIGIO_DATA_IN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_IN																															
R-X																															

Table 30-244. PRU_ICSS_IEP_DIGIO_DATA_IN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN	R	X	Data input. Digital inputs can be configured to be sampled in four ways. 1: Digital inputs are sampled at the start of each frame. The SOF signal can be used externally to update the input data, because the SOF is signaled before input data is sampled. 2: The sample time can be controlled externally by using the pr[k]_edio_latch_in signal. 3: Digital inputs are sampled at SYNC0 events. 4: Digital inputs are sampled at SYNC1 events. These can be configured by in_mode.

30.5.4.56 PRU_ICSS_IEP_DIGIO_DATA_IN_RAW Register (Offset = 30Ch) [reset = X]

PRU_ICSS_IEP_DIGIO_DATA_IN_RAW is shown in Figure 30-254 and described in Table 30-245.

Return to Summary Table.

DIGITAL DATA INPUT DIRECT SAMPLE

Figure 30-254. PRU_ICSS_IEP_DIGIO_DATA_IN_RAW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_IN_RAW																															
R-X																															

Table 30-245. PRU_ICSS_IEP_DIGIO_DATA_IN_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN_RAW	R	X	Data input which direct sample of pr[k]_edio_data_in[31 to 0].

30.5.4.57 PRU_ICSS_IEP_DIGIO_DATA_OUT Register (Offset = 310h) [reset = 0h]

PRU_ICSS_IEP_DIGIO_DATA_OUT is shown in Figure 30-255 and described in Table 30-246.

Return to Summary Table.

DIGITAL DATA OUTPUT

Figure 30-255. PRU_ICSS_IEP_DIGIO_DATA_OUT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_OUT																															
R/W-0h																															

Table 30-246. PRU_ICSS_IEP_DIGIO_DATA_OUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_OUT	R/W	0h	Data output. Digital outputs can be configured to be updated in four ways. 1: Digital outputs are updated at the end of each frame (EOF mode). 2: Digital outputs are updated with SYNC0 events 3: Digital outputs are updated SYNC1events. 4: Digital outputs are updated at the end of a frame which triggered the Process Data Watchdog. Digital Outputs are only updated if the frame was correct (WD_TRIG mode). These can be configured by out_mode.

30.5.4.58 PRU_ICSS_IEP_DIGIO_DATA_OUT_EN Register (Offset = 314h) [reset = 0h]

PRU_ICSS_IEP_DIGIO_DATA_OUT_EN is shown in Figure 30-256 and described in Table 30-247.

Return to Summary Table.

DIGITAL DATA OUT ENABLE

Figure 30-256. PRU_ICSS_IEP_DIGIO_DATA_OUT_EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_OUT_EN																															
R/W-0h																															

Table 30-247. PRU_ICSS_IEP_DIGIO_DATA_OUT_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_OUT_EN	R/W	0h	Data input which controls tri-state of pr[k]_edio_data_out_en[31 to 0], where a '1' sets output to Hi-Z..

30.5.4.59 PRU_ICSS_IEP_DIGIO_EXP Register (Offset = 318h) [reset = 20h]

PRU_ICSS_IEP_DIGIO_EXP is shown in Figure 30-257 and described in Table 30-248.

Return to Summary Table.

DIGIO EXPANSION REGISTER

Figure 30-257. PRU_ICSS_IEP_DIGIO_EXP Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED		EOF_SEL	SOF_SEL	SOF_DLY			
R-0h		R/W-0h	R/W-0h	R/W-0h			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OUTVALID_DLY				RESERVED	SW_OUTVALID	OUTVALID_OVR_EN	SW_DATA_OUT_UPDATE
R/W-2h				R-0h	R/W-0h	R/W-0h	R/W-0h

Table 30-248. PRU_ICSS_IEP_DIGIO_EXP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	
13	EOF_SEL	R/W	0h	Defines the source of RX_EOF is used for pr[k]_edio_data_in capture. 0h (R/W) = PRU0_RX_EOF 1h (R/W) = PRU1_RX_EOF
12	SOF_SEL	R/W	0h	Defines the source of RX_SOF is used for pr[k]_edio_data_in capture. 0h (R/W) = PRU0_RX_SOF 1h (R/W) = PRU1_RX_SOF
11-8	SOF_DLY	R/W	0h	Defines the number of iep_clk cycles delay before pr[k]_edio_data_in captures.
7-4	OUTVALID_DLY	R/W	2h	Defines the number of iep_clk cycles delay on assertion of pr1_edio_outvalid. Minimum is 2 clock cycles Maximum is 16 clock cycles
3	RESERVED	R	0h	
2	SW_OUTVALID	R/W	0h	pr1_edio_outvalid = SW_OUTVALID, only if OUTVALID_OVR_EN is set.
1	OUTVALID_OVR_EN	R/W	0h	Software override enable 0h (R/W) = Disable override 1h (R/W) = Enable override
0	SW_DATA_OUT_UPDATE	R/W	0h	Defines the value of pr[k]_edio_data_out when OUTVALID_OVR_EN = 1. 0h (W) = No effect 1h (W) = pr[k]_edio_data_out by software data out

30.5.5 PRU_ICSS_UART Registers

The system programmer has access to and control over any of the UART registers that are listed in Table 30-249. These registers, which control UART operations, receive data, and transmit data, are available at 32-bit addresses in the device memory map.

- RBR, THR, and DLL share one address. When the DLAB bit in LCR is 0, reading from the address gives the content of RBR, and writing to the address modifies THR. When DLAB = 1, all accesses at the address read or modify DLL. DLL can also be accessed with address offset 20h.
- IER and DLH share one address. When DLAB = 0, all accesses read or modify IER. When DLAB = 1, all accesses read or modify DLH. DLH can also be accessed with address offset 24h.
- IIR and FCR share one address. Regardless of the value of the DLAB bit, reading from the address gives the content of IIR, and writing modifies FCR.

Table 30-249. PRU_ICSS_UART Registers

Offset	Acronym	Register Description	Section
0h	RBR	Receiver Buffer Register (read only)	Section 30.5.5.1
0h	THR	Transmitter Holding Register (write only)	Section 30.5.5.2
4h	IER	Interrupt Enable Register	Section 30.5.5.3
8h	IIR	Interrupt Identification Register (read only)	Section 30.5.5.4
8h	FCR	FIFO Control Register (write only)	Section 30.5.5.5
Ch	LCR	Line Control Register	Section 30.5.5.6
10h	MCR	Modem Control Register	Section 30.5.5.7
14h	LSR	Line Status Register	Section 30.5.5.8
18h	MSR	Modem Status Register	Section 30.5.5.9
1Ch	SCR	Scratch Pad Register	Section 30.5.5.10
20h	DLL	Divisor LSB Latch	Section 30.5.5.11
24h	DLH	Divisor MSB Latch	Section 30.5.5.11
28h	REVID1	Revision Identification Register 1	Section 30.5.5.12
2Ch	REVID2	Revision Identification Register 2	Section 30.5.5.12
30h	PWREMU_MGMT	Power and Emulation Management Register	Section 30.5.5.13
34h	MDR	Mode Definition Register	Section 30.5.5.14

30.5.5.1 Receiver Buffer Register (RBR)

The receiver buffer register (RBR) is shown in Figure 30-258 and described in Table 30-250.

The UART receiver section consists of a receiver shift register (RSR) and a receiver buffer register (RBR). When the UART is in the FIFO mode, RBR is a 16-byte FIFO. Timing is supplied by the 16x receiver clock or 13x receiver clock by programming OSM_SEL bit field of MDR register. Receiver section control is a function of the line control register (LCR).

RSR receives serial data from the UARTn_RXD pin. Then RSR concatenates the data and moves it into RBR (or the receiver FIFO). In the non-FIFO mode, when a character is placed in RBR and the receiver data-ready interrupt is enabled (DR = 1 in IER), an interrupt is generated. This interrupt is cleared when the character is read from RBR. In the FIFO mode, the interrupt is generated when the FIFO is filled to the trigger level selected in the FIFO control register (FCR), and it is cleared when the FIFO contents drop below the trigger level.

Access considerations:

RBR, THR, and DLL share one address. To read RBR, write 0 to the DLAB bit in LCR, and read from the shared address. When DLAB = 0, writing to the shared address modifies THR. When DLAB = 1, all accesses at the shared address read or modify DLL.

DLL also has a dedicated address. If you use the dedicated address, you can keep DLAB = 0, so that RBR and THR are always selected at the shared address.

Figure 30-258. Receiver Buffer Register (RBR)

31 - 16
Reserved
R-0

15 - 8	7 - 0
Reserved	DATA
R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 30-250. Receiver Buffer Register (RBR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	DATA	0-FFh	Received data

30.5.5.2 Transmitter Holding Register (THR)

The transmitter holding register (THR) is shown in Figure 30-259 and described in Table 30-251.

The UART transmitter section consists of a transmitter hold register (THR) and a transmitter shift register (TSR). When the UART is in the FIFO mode, THR is a 16-byte FIFO. Transmitter section control is a function of the line control register (LCR).

THR receives data from the internal data bus and when TSR is idle, the UART moves the data from THR to TSR. The UART serializes the data in TSR and transmits the data on the TX pin. In the non-FIFO mode, if THR is empty and the THR empty (THRE) interrupt is enabled (ETBEI = 1 in IER), an interrupt is generated. This interrupt is cleared when a character is loaded into THR or the interrupt identification register (IIR) is read. In the FIFO mode, the interrupt is generated when the transmitter FIFO is empty, and it is cleared when at least one byte is loaded into the FIFO or IIR is read.

Access considerations:

RBR, THR, and DLL share one address. To load THR, write 0 to the DLAB bit of LCR, and write to the shared address. When DLAB = 0, reading from the shared address gives the content of RBR. When DLAB = 1, all accesses at the address read or modify DLL.

DLL also has a dedicated address. If you use the dedicated address, you can keep DLAB = 0, so that RBR and THR are always selected at the shared address.

Figure 30-259. Transmitter Holding Register (THR)

31–16
Reserved
R-0

15–8	7–0
Reserved	DATA
R-0	W-0

LEGEND: R = Read only; W = Write only; -*n* = value after reset

Table 30-251. Transmitter Holding Register (THR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	DATA	0-FFh	Data to transmit

30.5.5.3 Interrupt Enable Register (IER)

The interrupt enable register (IER) is used to individually enable or disable each type of interrupt request that can be generated by the UART. Each interrupt request that is enabled in IER is forwarded to the CPU. IER is shown in Figure 30-260 and described in Table 30-252.

Access considerations:

IER and DLH share one address. To read or modify IER, write 0 to the DLAB bit in LCR. When DLAB = 1, all accesses at the shared address read or modify DLH.

DLH also has a dedicated address. If you use the dedicated address, you can keep DLAB = 0, so that IER is always selected at the shared address.

Figure 30-260. Interrupt Enable Register (IER)

31-16
Reserved
R-0

15-4	3	2	1	0
Reserved	Rsvd	ELSI	ETBEI	ERBI
R-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 30-252. Interrupt Enable Register (IER) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved
3	EDSSI	0	Enable Modem Status Interrupt
2	ELSI		Receiver line status interrupt enable.
		0	Receiver line status interrupt is disabled.
		1	Receiver line status interrupt is enabled.
1	ETBEI		Transmitter holding register empty interrupt enable.
		0	Transmitter holding register empty interrupt is disabled.
		1	Transmitter holding register empty interrupt is enabled.
0	ERBI		Receiver data available interrupt and character timeout indication interrupt enable.
		0	Receiver data available interrupt and character timeout indication interrupt is disabled.
		1	Receiver data available interrupt and character timeout indication interrupt is enabled.

30.5.5.4 Interrupt Identification Register (IIR)

The interrupt identification register (IIR) is a read-only register at the same address as the FIFO control register (FCR), which is a write-only register. When an interrupt is generated and enabled in the interrupt enable register (IER), IIR indicates that an interrupt is pending in the IPEND bit and encodes the type of interrupt in the INTID bits. Reading IIR clears any THR empty (THRE) interrupts that are pending.

IIR is shown in Figure 30-261 and described in Figure 30-261.

The UART has an on-chip interrupt generation and prioritization capability that permits flexible communication with the CPU. The UART provides three priority levels of interrupts:

- Priority 1 - Receiver line status (highest priority)
- Priority 2 - Receiver data ready or receiver timeout
- Priority 3 - Transmitter holding register empty

The FIFOEN bit in IIR can be checked to determine whether the UART is in the FIFO mode or the non-FIFO mode.

Access consideration:

IIR and FCR share one address. Regardless of the value of the DLAB bit in LCR, reading from the address gives the content of IIR, and writing to the address modifies FCR.

Figure 30-261. Interrupt Identification Register (IIR)

31-16
Reserved
R-0

15-8	7-6	5-4	3-1	0
Reserved	FIFOEN	Reserved	INTID	IPEND
R-0	R-0	R-0	R-0	R-1

LEGEND: R = Read only; -*n* = value after reset

Table 30-253. Interrupt Identification Register (IIR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-6	FIFOEN	0-3h	FIFOs enabled.
		0	Non-FIFO mode
		1h-2h	Reserved
		3h	FIFOs are enabled. FIFOEN bit in the FIFO control register (FCR) is set to 1.
5-4	Reserved	0	Reserved
3-1	INTID	0-7h	Interrupt type. See Table 30-254.
		0	Reserved
		1h	Transmitter holding register empty (priority 3)
		2h	Receiver data available (priority 2)
		3h	Receiver line status (priority 1, highest)
		4h-5h	Reserved
		6h	Character timeout indication (priority 2)
		7h	Reserved
0	IPEND		Interrupt pending. When any UART interrupt is generated and is enabled in IER, IPEND is forced to 0. IPEND remains 0 until all pending interrupts are cleared or until a hardware reset occurs. If no interrupts are enabled, IPEND is never forced to 0.
		0	Interrupts pending.
		1	No interrupts pending.

Table 30-254. Interrupt Identification and Interrupt Clearing Information

Priority Level	IIR Bits				Interrupt Type	Interrupt Source	Event That Clears Interrupt
	3	2	1	0			
None	0	0	0	1	None	None	None
1	0	1	1	0	Receiver line status	Overrun error, parity error, framing error, or break is detected.	For an overrun error, reading the line status register (LSR) clears the interrupt. For a parity error, framing error, or break, the interrupt is cleared only after all the erroneous data have been read.
2	0	1	0	0	Receiver data-ready	Non-FIFO mode: Receiver data is ready.	Non-FIFO mode: The receiver buffer register (RBR) is read.
						FIFO mode: Trigger level reached. If four character times (see Table 30-36) pass with no access of the FIFO, the interrupt is asserted again.	FIFO mode: The FIFO drops below the trigger level.[1]
2	1	1	0	0	Receiver time-out	FIFO mode only: No characters have been removed from or input to the receiver FIFO during the last four character times (see Table 30-36), and there is at least one character in the receiver FIFO during this time.	One of the following events: • A character is read from the receiver FIFO.[1] • A new character arrives in the receiver FIFO. • The URRST bit in the power and emulation management register (PWREMU_MGMT) is loaded with 0.
3	0	0	1	0	Transmitter holding register empty	Non-FIFO mode: Transmitter holding register (THR) is empty. FIFO mode: Transmitter FIFO is empty.	A character is written to the transmitter holding register (THR) or the interrupt identification register (IIR) is read.

[1] In the FIFO mode, the receiver data-ready interrupt or receiver time-out interrupt is cleared by the CPU or by the DMA controller, whichever reads from the receiver FIFO first.

30.5.5.5 FIFO Control Register (FCR)

The FIFO control register (FCR) is a write-only register at the same address as the interrupt identification register (IIR), which is a read-only register. Use FCR to enable and clear the FIFOs and to select the receiver FIFO trigger level FCR is shown in Figure 30-262 and described in Table 30-255. The FIFOEN bit must be set to 1 before other FCR bits are written to or the FCR bits are not programmed.

Access consideration:

IIR and FCR share one address. Regardless of the value of the DLAB bit, reading from the address gives the content of IIR, and writing to the address modifies FCR.

> **CAUTION**
>
> For proper communication between the UART and the EDMA controller, the DMAMODE1 bit must be set to 1. Always write a 1 to the DMAMODE1 bit, and after a hardware reset, change the DMAMODE1 bit from 0 to 1.

Figure 30-262. FIFO Control Register (FCR)

31	16
Reserved	
R-0	

15	8
Reserved	
R-0	

7	6	5	4	3	2	1	0
RXFIFTL		Reserved		DMAMODE1[1]	TXCLR	RXCLR	FIFOEN
W-0		R-0		W-0	W1C-0	W1C-0	W-0

LEGEND: R = Read only; W = Write only; W1C = Write 1 to clear (writing 0 has no effect); -*n* = value after reset

[1] Always write 1 to the DMAMODE1 bit. After a hardware reset, change the DMAMODE1 bit from 0 to 1. DMAMODE = 1 is required for proper communication between the UART and the DMA controller.

Table 30-255. FIFO Control Register (FCR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-6	RXFIFTL	0-3h	Receiver FIFO trigger level. RXFIFTL sets the trigger level for the receiver FIFO. When the trigger level is reached, a receiver data-ready interrupt is generated (if the interrupt request is enabled). Once the FIFO drops below the trigger level, the interrupt is cleared.
		0	1 byte
		1h	4 bytes
		2h	8 bytes
		3h	14 bytes
5-4	Reserved	0	Reserved
3	DMAMODE1		DMA MODE1 enable if FIFOs are enabled. Always write 1 to DMAMODE1. After a hardware reset, change DMAMODE1 from 0 to 1. DMAMOD1 = 1 is a requirement for proper communication between the UART and the EDMA controller.
		0	DMA MODE1 is disabled.
		1	DMA MODE1 is enabled.
2	TXCLR		Transmitter FIFO clear. Write a 1 to TXCLR to clear the bit.
		0	No effect.
		1	Clears transmitter FIFO and resets the transmitter FIFO counter. The shift register is not cleared.
1	RXCLR		Receiver FIFO clear. Write a 1 to RXCLR to clear the bit.
		0	No effect.
		1	Clears receiver FIFO and resets the receiver FIFO counter. The shift register is not cleared.
0	FIFOEN		Transmitter and receiver FIFOs mode enable. FIFOEN must be set before other FCR bits are written to or the FCR bits are not programmed. Clearing this bit clears the FIFO counters.
		0	Non-FIFO mode. The transmitter and receiver FIFOs are disabled, and the FIFO pointers are cleared.
		1	FIFO mode. The transmitter and receiver FIFOs are enabled.

30.5.5.6 Line Control Register (LCR)

The line control register (LCR) is shown in Figure 30-263 and described in Table 30-256.

The system programmer controls the format of the asynchronous data communication exchange by using LCR. In addition, the programmer can retrieve, inspect, and modify the content of LCR; this eliminates the need for separate storage of the line characteristics in system memory.

Figure 30-263. Line Control Register (LCR)

31	16
Reserved	
R-0	

15	8	7	6	5	4	3	2	1	0
Reserved		DLAB	BC	SP	EPS	PEN	STB	WLS	
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 30-256. Line Control Register (LCR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7	DLAB		Divisor latch access bit. The divisor latch registers (DLL and DLH) can be accessed at dedicated addresses or at addresses shared by RBR, THR, and IER. Using the shared addresses requires toggling DLAB to change which registers are selected. If you use the dedicated addresses, you can keep DLAB = 0.
		0	Allows access to the receiver buffer register (RBR), the transmitter holding register (THR), and the interrupt enable register (IER) selected. At the address shared by RBR, THR, and DLL, the CPU can read from RBR and write to THR. At the address shared by IER and DLH, the CPU can read from and write to IER.
		1	Allows access to the divisor latches of the baud generator during a read or write operation (DLL and DLH). At the address shared by RBR, THR, and DLL, the CPU can read from and write to DLL. At the address shared by IER and DLH, the CPU can read from and write to DLH.
6	BC		Break control.
		0	Break condition is disabled.
		1	Break condition is transmitted to the receiving UART. A break condition is a condition where the UARTn_TXD signal is forced to the spacing (cleared) state.
5	SP		Stick parity. The SP bit works in conjunction with the EPS and PEN bits. The relationship between the SP, EPS, and PEN bits is summarized in Table 30-257.
		0	Stick parity is disabled.
		1	Stick parity is enabled. • When odd parity is selected (EPS = 0), the PARITY bit is transmitted and checked as set. • When even parity is selected (EPS = 1), the PARITY bit is transmitted and checked as cleared.
4	EPS		Even parity select. Selects the parity when parity is enabled (PEN = 1). The EPS bit works in conjunction with the SP and PEN bits. The relationship between the SP, EPS, and PEN bits is summarized in Table 30-257.
		0	Odd parity is selected (an odd number of logic 1s is transmitted or checked in the data and PARITY bits).
		1	Even parity is selected (an even number of logic 1s is transmitted or checked in the data and PARITY bits).
3	PEN		Parity enable. The PEN bit works in conjunction with the SP and EPS bits. The relationship between the SP, EPS, and PEN bits is summarized in Table 30-257.
		0	No PARITY bit is transmitted or checked.
		1	Parity bit is generated in transmitted data and is checked in received data between the last data word bit and the first STOP bit.

Table 30-256. Line Control Register (LCR) Field Descriptions (continued)

Bit	Field	Value	Description
2	STB		Number of STOP bits generated. STB specifies 1, 1.5, or 2 STOP bits in each transmitted character. When STB = 1, the WLS bit determines the number of STOP bits. The receiver clocks only the first STOP bit, regardless of the number of STOP bits selected. The number of STOP bits generated is summarized in Table 30-258.
		0	1 STOP bit is generated.
		1	WLS bit determines the number of STOP bits: • When WLS = 0, 1.5 STOP bits are generated. • When WLS = 1h, 2h, or 3h, 2 STOP bits are generated.
1-0	WLS	0-3h	Word length select. Number of bits in each transmitted or received serial character. When STB = 1, the WLS bit determines the number of STOP bits.
		0	5 bits
		1h	6 bits
		2h	7 bits
		3h	8 bits

Table 30-257. Relationship Between ST, EPS, and PEN Bits in LCR

ST Bit	EPS Bit	PEN Bit	Parity Option
x	x	0	Parity disabled: No PARITY bit is transmitted or checked
0	0	1	Odd parity selected: Odd number of logic 1s
0	1	1	Even parity selected: Even number of logic 1s
1	0	1	Stick parity selected with PARITY bit transmitted and checked as set
1	1	1	Stick parity selected with PARITY bit transmitted and checked as cleared

Table 30-258. Number of STOP Bits Generated

STB Bit	WLS Bits	Word Length Selected with WLS Bits	Number of STOP Bits Generated	Baud Clock (BCLK) Cycles
0	x	Any word length	1	16
1	0h	5 bits	1.5	24
1	1h	6 bits	2	32
1	2h	7 bits	2	32
1	3h	8 bits	2	32

30.5.5.7 Modem Control Register (MCR)

The modem control register (MCR) is shown in Figure 30-264 and described in Table 30-259. The modem control register provides the ability to enable/disable the autoflow functions, and enable/disable the loopback function for diagnostic purposes.

Figure 30-264. Modem Control Register (MCR)

31	16
Reserved	
R-0	

15	6	5	4	3	2	1	0
Reserved		AFE[1]	LOOP	OUT2	OUT1	RTS[1]	Rsvd
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

[1] All UARTs do not support this feature, see your device-specific data manual for supported features. If this feature is not available, this bit is reserved and should be cleared to 0.

Table 30-259. Modem Control Register (MCR) Field Descriptions

Bit	Field	Value	Description
31-6	Reserved	0	Reserved
5	AFE		Autoflow control enable. Autoflow control allows the $\overline{\text{UARTn\_RTS}}$ and $\overline{\text{UARTn\_CTS}}$ signals to provide handshaking between UARTs during data transfer. When AFE = 1, the RTS bit determines the autoflow control enabled. Note that all UARTs do not support this feature, see your device-specific data manual for supported features. If this feature is not available, this bit is reserved and should be cleared to 0.
		0	Autoflow control is disabled.
		1	Autoflow control is enabled: • When RTS = 0, $\overline{\text{UARTn\_CTS}}$ is only enabled. • When RTS = 1, $\overline{\text{UARTn\_RTS}}$ and $\overline{\text{UARTn\_CTS}}$ are enabled.
4	LOOP		Loop back mode enable. LOOP is used for the diagnostic testing using the loop back feature.
		0	Loop back mode is disabled.
		1	Loop back mode is enabled. When LOOP is set, the following occur: • The UARTn_TXD signal is set high. • The UARTn_RXD pin is disconnected • The output of the transmitter shift register (TSR) is lopped back in to the receiver shift register (RSR) input.
3	OUT2	0	OUT2 Control Bit
2	OUT1	0	OUT1 Control Bit
1	RTS		RTS control. When AFE = 1, the RTS bit determines the autoflow control enabled. Note that all UARTs do not support this feature, see your device-specific data manual for supported features. If this feature is not available, this bit is reserved and should be cleared to 0.
		0	$\overline{\text{UARTn\_RTS}}$ is disabled, $\overline{\text{UARTn\_CTS}}$ is only enabled.
		1	$\overline{\text{UARTn\_RTS}}$ and $\overline{\text{UARTn\_CTS}}$ are enabled.
0	Reserved	0	Reserved

30.5.5.8 Line Status Register (LSR)

The line status register (LSR) is shown in Figure 30-265 and described in Table 30-260. LSR provides information to the CPU concerning the status of data transfers. LSR is intended for read operations only; do not write to this register. Bits 1 through 4 record the error conditions that produce a receiver line status interrupt.

Figure 30-265. Line Status Register (LSR)

31–16
Reserved
R-0

15–8	7	6	5	4	3	2	1	0
Reserved	RXFIFOE	TEMT	THRE	BI	FE	PE	OE	DR
R-0	R-0	R-1	R-1	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 30-260. Line Status Register (LSR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7	RXFIFOE		Receiver FIFO error.
			In non-FIFO mode:
		0	There has been no error, or RXFIFOE was cleared because the CPU read the erroneous character from the receiver buffer register (RBR).
		1	There is a parity error, framing error, or break indicator in the receiver buffer register (RBR).
			In FIFO mode:
		0	There has been no error, or RXFIFOE was cleared because the CPU read the erroneous character from the receiver FIFO and there are no more errors in the receiver FIFO.
		1	At least one parity error, framing error, or break indicator in the receiver FIFO.
6	TEMT		Transmitter empty (TEMT) indicator.
			In non-FIFO mode:
		0	Either the transmitter holding register (THR) or the transmitter shift register (TSR) contains a data character.
		1	Both the transmitter holding register (THR) and the transmitter shift register (TSR) are empty.
			In FIFO mode:
		0	Either the transmitter FIFO or the transmitter shift register (TSR) contains a data character.
		1	Both the transmitter FIFO and the transmitter shift register (TSR) are empty.
5	THRE		Transmitter holding register empty (THRE) indicator. If the THRE bit is set and the corresponding interrupt enable bit is set (ETBEI = 1 in IER), an interrupt request is generated.
			In non-FIFO mode:
		0	Transmitter holding register (THR) is not empty. THR has been loaded by the CPU.
		1	Transmitter holding register (THR) is empty (ready to accept a new character). The content of THR has been transferred to the transmitter shift register (TSR).
			In FIFO mode:
		0	Transmitter FIFO is not empty. At least one character has been written to the transmitter FIFO. You can write to the transmitter FIFO if it is not full.
		1	Transmitter FIFO is empty. The last character in the FIFO has been transferred to the transmitter shift register (TSR).

Table 30-260. Line Status Register (LSR) Field Descriptions (continued)

Bit	Field	Value	Description
4	BI		Break indicator. The BI bit is set whenever the receive data input (UARTn_RXD) was held low for longer than a full-word transmission time. A full-word transmission time is defined as the total time to transmit the START, data, PARITY, and STOP bits. If the BI bit is set and the corresponding interrupt enable bit is set (ELSI = 1 in IER), an interrupt request is generated.
			In non-FIFO mode:
		0	No break has been detected, or the BI bit was cleared because the CPU read the erroneous character from the receiver buffer register (RBR).
		1	A break has been detected with the character in the receiver buffer register (RBR).
			In FIFO mode:
		0	No break has been detected, or the BI bit was cleared because the CPU read the erroneous character from the receiver FIFO and the next character to be read from the FIFO has no break indicator.
		1	A break has been detected with the character at the top of the receiver FIFO.
3	FE		Framing error (FE) indicator. A framing error occurs when the received character does not have a valid STOP bit. In response to a framing error, the UART sets the FE bit and waits until the signal on the RX pin goes high. Once the RX signal goes high, the receiver is ready to detect a new START bit and receive new data. If the FE bit is set and the corresponding interrupt enable bit is set (ELSI = 1 in IER), an interrupt request is generated.
			In non-FIFO mode:
		0	No framing error has been detected, or the FE bit was cleared because the CPU read the erroneous data from the receiver buffer register (RBR).
		1	A framing error has been detected with the character in the receiver buffer register (RBR).
			In FIFO mode:
		0	No framing error has been detected, or the FE bit was cleared because the CPU read the erroneous data from the receiver FIFO and the next character to be read from the FIFO has no framing error.
		1	A framing error has been detected with the character at the top of the receiver FIFO.
2	PE		Parity error (PE) indicator. A parity error occurs when the parity of the received character does not match the parity selected with the EPS bit in the line control register (LCR). If the PE bit is set and the corresponding interrupt enable bit is set (ELSI = 1 in IER), an interrupt request is generated.
			In non-FIFO mode:
		0	No parity error has been detected, or the PE bit was cleared because the CPU read the erroneous data from the receiver buffer register (RBR).
		1	A parity error has been detected with the character in the receiver buffer register (RBR).
			In FIFO mode:
		0	No parity error has been detected, or the PE bit was cleared because the CPU read the erroneous data from the receiver FIFO and the next character to be read from the FIFO has no parity error.
		1	A parity error has been detected with the character at the top of the receiver FIFO.
1	OE		Overrun error (OE) indicator. An overrun error in the non-FIFO mode is different from an overrun error in the FIFO mode. If the OE bit is set and the corresponding interrupt enable bit is set (ELSI = 1 in IER), an interrupt request is generated.
			In non-FIFO mode:
		0	No overrun error has been detected, or the OE bit was cleared because the CPU read the content of the line status register (LSR).
		1	Overrun error has been detected. Before the character in the receiver buffer register (RBR) could be read, it was overwritten by the next character arriving in RBR.
			In FIFO mode:
		0	No overrun error has been detected, or the OE bit was cleared because the CPU read the content of the line status register (LSR).
		1	Overrun error has been detected. If data continues to fill the FIFO beyond the trigger level, an overrun error occurs only after the FIFO is full and the next character has been completely received in the shift register. An overrun error is indicated to the CPU as soon as it happens. The new character overwrites the character in the shift register, but it is not transferred to the FIFO.

Table 30-260. Line Status Register (LSR) Field Descriptions (continued)

Bit	Field	Value	Description
0	DR		Data-ready (DR) indicator for the receiver. If the DR bit is set and the corresponding interrupt enable bit is set (ERBI = 1 in IER), an interrupt request is generated.
			In non-FIFO mode:
		0	Data is not ready, or the DR bit was cleared because the character was read from the receiver buffer register (RBR).
		1	Data is ready. A complete incoming character has been received and transferred into the receiver buffer register (RBR).
			In FIFO mode:
		0	Data is not ready, or the DR bit was cleared because all of the characters in the receiver FIFO have been read.
		1	Data is ready. There is at least one unread character in the receiver FIFO. If the FIFO is empty, the DR bit is set as soon as a complete incoming character has been received and transferred into the FIFO. The DR bit remains set until the FIFO is empty again.

30.5.5.9 Modem Status Register (MSR)

The Modem status register (MSR) is shown in Figure 30-266 and described in Table 30-261. MSR provides information to the CPU concerning the status of modem control signals. MSR is intended for read operations only; do not write to this register.

Figure 30-266. Modem Status Register (MSR)

31–16
Reserved
R-0

15–8	7	6	5	4	3	2	1	0
Reserved	CD	RI	DSR	CTS	DCD	TERI	DDSR	DCTS
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 30-261. Modem Status Register (MSR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7	CD	0	Complement of the Carrier Detect input. When the UART is in the diagnostic test mode (loopback mode MCR[4] = 1), this bit is equal to the MCR bit 3 (OUT2).
6	RI	0	Complement of the Ring Indicator input. When the UART is in the diagnostic test mode (loopback mode MCR[4] = 1), this bit is equal to the MCR bit 2 (OUT1).
5	DSR	0	Complement of the Data Set Ready input. When the UART is in the diagnostic test mode (loopback mode MCR[4] = 1), this bit is equal to the MCR bit 0 (DTR).
4	CTS	0	Complement of the Clear To Send input. When the UART is in the diagnostic test mode (loopback mode MCR[4] = 1), this bit is equal to the MCR bit 1 (RTS).
3	DCD	0	Change in DCD indicator bit. DCD indicates that the DCD input has changed state since the last time it was read by the CPU. When DCD is set and the modem status interrupt is enabled, a modem status interrupt is generated.
2	TERI	0	Trailing edge of RI (TERI) indicator bit. TERI indicates that the RI input has changed from a low to a high. When TERI is set and the modem status interrupt is enabled, a modem status interrupt is generated.
1	DDSR	0	Change in DSR indicator bit. DDSR indicates that the DSR input has changed state since the last time it was read by the CPU. When DDSR is set and the modem status interrupt is enabled, a modem status interrupt is generated.
0	DCTS	0	Change in CTS indicator bit. DCTS indicates that the CTS input has changed state since the last time it was read by the CPU. When DCTS is set (autoflow control is not enabled and the modem status interrupt is enabled), a modem status interrupt is generated. When autoflow control is enabled, no interrupt is generated.

30.5.5.10 Scratch Pad Register (SCR)

The Scratch Pad register (SCR) is shown in Figure 30-267 and described in Table 30-262. SCR is intended for programmer's use as a scratch pad. It temporarily holds the programmer's data without affecting UART operation.

Figure 30-267. Scratch Pad Register (SCR)

31–16
Reserved
R-0

15–8	7–0
Reserved	SCR
R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 30-262. Scratch Pad Register (MSR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	SCR	0	These bits are intended for the programmer's use as a scratch pad in the sense that it temporarily holds the programmer's data without affecting any other UART operation.

30.5.5.11 Divisor Latches (DLL and DLH)

Two 8-bit register fields (DLL and DLH), called divisor latches, store the 16-bit divisor for generation of the baud clock in the baud generator. The latches are in DLH and DLL. DLH holds the most-significant bits of the divisor, and DLL holds the least-significant bits of the divisor. These divisor latches must be loaded during initialization of the UART in order to ensure desired operation of the baud generator. Writing to the divisor latches results in two wait states being inserted during the write access while the baud generator is loaded with the new value.

Access considerations:

- RBR, THR, and DLL share one address. When DLAB = 1 in LCR, all accesses at the shared address are accesses to DLL. When DLAB = 0, reading from the shared address gives the content of RBR, and writing to the shared address modifies THR.
- IER and DLH share one address. When DLAB = 1 in LCR, accesses to the shared address read or modify to DLH. When DLAB = 0, all accesses at the shared address read or modify IER.

DLL and DLH also have dedicated addresses. If you use the dedicated addresses, you can keep the DLAB bit cleared, so that RBR, THR, and IER are always selected at the shared addresses.

The divisor LSB latch (DLL) is shown in Figure 30-268 and described in Table 30-263. The divisor MSB latch (DLH) is shown in Figure 30-269 and described in Table 30-264.

Figure 30-268. Divisor LSB Latch (DLL)

31	16
Reserved	
R-0	

15	8	7	0
Reserved		DLL	
R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 30-263. Divisor LSB Latch (DLL) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	DLL	0-Fh	The 8 least-significant bits (LSBs) of the 16-bit divisor for generation of the baud clock in the baud rate generator.

Figure 30-269. Divisor MSB Latch (DLH)

31	16
Reserved	
R-0	

15	8	7	0
Reserved		DLH	
R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 30-264. Divisor MSB Latch (DLH) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	DLH	0-Fh	The 8 most-significant bits (MSBs) of the 16-bit divisor for generation of the baud clock in the baud rate generator.

30.5.5.12 Revision Identification Registers (REVID1 and REVID2)

The revision identification registers (REVID1 and REVID2) contain peripheral identification data for the peripheral. REVID1 is shown in Figure 30-270 and described in Table 30-265. REVID2 is shown in Figure 30-271 and described in Table 30-266.

Figure 30-270. Revision Identification Register 1 (REVID1)

31 … 0
REVID1
R-1102 0002h

LEGEND: R = Read only; -*n* = value after reset

Table 30-265. Revision Identification Register 1 (REVID1) Field Descriptions

Bit	Field	Value	Description
31-0	REVID1	1102 0002h	Peripheral Identification Number

Figure 30-271. Revision Identification Register 2 (REVID2)

31 … 16
Reserved
R-0

15 … 8	7 … 0
Reserved	REVID2
R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 30-266. Revision Identification Register 2 (REVID2) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	REVID2	0	Peripheral Identification Number

30.5.5.13 Power and Emulation Management Register (PWREMU_MGMT)

The power and emulation management register (PWREMU_MGMT) is shown in Figure 30-272 and described in Table 30-267.

Figure 30-272. Power and Emulation Management Register (PWREMU_MGMT)

31	16
Reserved	
R-0	

15	14	13	12	1	0
Rsvd	UTRST	URRST	Reserved		FREE
R/W-0	R/W-0	R/W-0	R-1		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 30-267. Power and Emulation Management Register (PWREMU_MGMT) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15	Reserved	0	Reserved. This bit must always be written with a 0.
14	UTRST		UART transmitter reset. Resets and enables the transmitter.
		0	Transmitter is disabled and in reset state.
		1	Transmitter is enabled.
13	URRST		UART receiver reset. Resets and enables the receiver.
		0	Receiver is disabled and in reset state.
		1	Receiver is enabled.
12-1	Reserved	1	Reserved
0	FREE		Free-running enable mode bit. This bit determines the emulation mode functionality of the UART. When halted, the UART can handle register read/write requests, but does not generate any transmission/reception, interrupts or events.
		0	If a transmission is not in progress, the UART halts immediately. If a transmission is in progress, the UART halts after completion of the one-word transmission.
		1	Free-running mode is enabled; UART continues to run normally.

30.5.5.14 Mode Definition Register (MDR)

The Mode Definition register (MDR) determines the over-sampling mode for the UART. MDR is shown in Figure 30-273 and described in Table 30-268.

Figure 30-273. Mode Definition Register (MDR)

31	16
Reserved	
R-0	

15	1	0
Reserved		OSM_SEL
R-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 30-268. Mode Definition Register (MDR) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved
0	OSM_SEL		Over-Sampling Mode Select.
		0	16× over-sampling.
		1	13× over-sampling.

30.5.6 PRU_ICSS_ECAP Registers

For additional details about the ECAP registers, see *Enhanced Capture(eCAP) Module Registers* in the AM437x TRM (SPRUHL7).

30.5.7 PRU_ICSS_MII_RT Registers

Table 30-269 lists the memory-mapped registers for the PRU_ICSS_MII_RT. All register offset addresses not listed in Table 30-269 should be considered as reserved locations and the register contents should not be modified.

Table 30-269. PRU_ICSS_MII_RT Registers

Offset	Acronym	Register Name	Section
0h	RXCFG0		Section 30.5.7.1
4h	RXCFG1		Section 30.5.7.2
10h	TXCFG0		Section 30.5.7.3
14h	TXCFG1		Section 30.5.7.4
20h	TXCRC0		Section 30.5.7.5
24h	TXCRC1		Section 30.5.7.6
30h	TXIPG0		Section 30.5.7.7
34h	TXIPG1		Section 30.5.7.8
38h	PRS0		Section 30.5.7.9
3Ch	PRS1		Section 30.5.7.10
40h	RXFRMS0		Section 30.5.7.11
44h	RXFRMS1		Section 30.5.7.12
48h	RXPCNT0		Section 30.5.7.13
4Ch	RXPCNT1		Section 30.5.7.14
50h	RXERR0		Section 30.5.7.15
54h	RXERR1		Section 30.5.7.16
60h	RXFLV0		Section 30.5.7.17

Table 30-269. PRU_ICSS_MII_RT Registers (continued)

Offset	Acronym	Register Name	Section
64h	RXFLV1		Section 30.5.7.18
68h	TXFLV0		Section 30.5.7.19
6Ch	TXFLV1		Section 30.5.7.20

30.5.7.1 RXCFG0 Register (Offset = 0h) [reset = 0h]

RXCFG0 is shown in Figure 30-274 and described in Table 30-270.

Return to Summary Table.

RX CONFIG0

Figure 30-274. RXCFG0 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	RX_AUTOFWD_PRE	RX_BYTE_SWAP	RX_L2_ENABLE	RX_MUX_SEL	RX_CUT_PREAMBLE	RESERVED	RX_ENABLE
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h

Table 30-270. RXCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	RX_AUTOFWD_PRE	R/W	0h	Enables auto-forward of received preamble. When enabled, this will forward the preamble nibbles including the SFD to the TX L1 FIFO that is attached to the PRU. First data byte seen by PRU R31 and/or RX L2 is destination address (DA). Odd number of preamble nibbles is supported in this mode. For example, 0x55D. Note that new RX should only occur after the current TX completes. 0x 0: Disable 0x 1: Enable, it must disable RX_CUT_PREAMBLE and TX_AUTO_PREAMBLE
5	RX_BYTE_SWAP	R/W	0h	Defines the order of Byte0/1 placement for RX R31 and RX L2. Note that if TX_AUTO_SEQUENCE enabled, this bit cannot get enable since TX_BYTE_SWAP on swaps the PRU output. This bit must be selected/updated when the port is disabled or there is no traffic. 0x 0: R31 [15:8]/RXL2 [15:8] = Byte1{Nibble3,Nibble2} R31[7:0]/RXL2 [7:0] = Byte0{Nibble1,Nibble0} 0x 1: R31 [15:8]/RXL2 [15:8] = Byte0{Nibble1,Nibble0} R31[7:0]/RXL2 [7:0] = Byte1{Nibble3,Nibble2} Nibble0 is the first nibble received.
4	RX_L2_ENABLE	R/W	0h	Enables RX L2 buffer. 0x 0: Disable (RX L2 can function as generic scratch pad) 0x 1: Enable

Table 30-270. RXCFG0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	RX_MUX_SEL	R/W	0h	Selects receive data source. Typically, the setting for this will not be identical for the two MII receive configuration registers. 0x 0: MII RX Data from Port 0 (default for RXCFG0) 0x 1: MII RX Data from Port 1 (default for RXCFG1)
2	RX_CUT_PREAMBLE	R/W	0h	Removes received preamble. 0x 0: All data from Ethernet PHY are passed on to PRU register. This assumes Ethernet PHY which does not shorten the preamble. 0x 1: MII interface suppresses preamble and sync frame delimiter. First data byte seen by PRU register is destination address.
1	RESERVED	R	0h	
0	RX_ENABLE	R/W	0h	Enables the receive traffic currently selected by RX_MUX_SELECT. 0x 0: Disable 0x 1: Enable

30.5.7.2 RXCFG1 Register (Offset = 4h) [reset = 8h]

RXCFG1 is shown in Figure 30-275 and described in Table 30-271.

Return to Summary Table.

RX CONFIG1

Figure 30-275. RXCFG1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	RX_AUTOFWD_PRE	RX_BYTE_SWAP	RX_L2_ENABLE	RX_MUX_SEL	RX_CUT_PREAMBLE	RESERVED	RX_ENABLE
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R-0h	R/W-0h

Table 30-271. RXCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	RX_AUTOFWD_PRE	R/W	0h	Enables auto-forward of received preamble. When enabled, this will forward the preamble nibbles including the SFD to the TX L1 FIFO that is attached to the PRU. First data byte seen by PRU R31 and/or RX L2 is destination address (DA). Odd number of preamble nibbles is supported in this mode. For example, 0x55D. Note that new RX should only occur after the current TX completes. 0x 0: Disable 0x 1: Enable, it must disable RX_CUT_PREAMBLE and TX_AUTO_PREAMBLE
5	RX_BYTE_SWAP	R/W	0h	Defines the order of Byte0/1 placement for RX R31 and RX L2. Note that if TX_AUTO_SEQUENCE enabled, this bit cannot get enable since TX_BYTE_SWAP on swaps the PRU output. This bit must be selected/updated when the port is disabled or there is no traffic. 0x 0: R31 [15:8]/RXL2 [15:8] = Byte1{Nibble3,Nibble2} R31[7:0]/RXL2 [7:0] = Byte0{Nibble1,Nibble0} 0x 1: R31 [15:8]/RXL2 [15:8] = Byte0{Nibble1,Nibble0} R31[7:0]/RXL2 [7:0] = Byte1{Nibble3,Nibble2} Nibble0 is the first nibble received.
4	RX_L2_ENABLE	R/W	0h	Enables RX L2 buffer. 0x 0: Disable (RX L2 can function as generic scratch pad) 0x 1: Enable

Table 30-271. RXCFG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	RX_MUX_SEL	R/W	1h	Selects receive data source. Typically, the setting for this will not be identical for the two MII receive configuration registers. 0x 0: MII RX Data from Port 0 (default for RXCFG0) 0x 1: MII RX Data from Port 1 (default for RXCFG1)
2	RX_CUT_PREAMBLE	R/W	0h	Removes received preamble. 0x 0: All data from Ethernet PHY are passed on to PRU register. This assumes Ethernet PHY which does not shorten the preamble. 0x 1: MII interface suppresses preamble and sync frame delimiter. First data byte seen by PRU register is destination address.
1	RESERVED	R	0h	
0	RX_ENABLE	R/W	0h	Enables the receive traffic currently selected by RX_MUX_SELECT. 0x 0: Disable 0x 1: Enable

30.5.7.3 TXCFG0 Register (Offset = 10h) [reset = 00020010h]

TXCFG0 is shown in Figure 30-276 and described in Table 30-272.

Return to Summary Table.

TX CONFIG0

Figure 30-276. TXCFG0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td>RESERVED</td><td colspan="3">TX_CLK_DELAY</td><td colspan="2">RESERVED</td><td colspan="2">TX_START_DELAY</td></tr>
<tr><td>R-0h</td><td colspan="3">R/W-0h</td><td colspan="2">R-0h</td><td colspan="2">R/W-2h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">TX_START_DELAY</td></tr>
<tr><td colspan="8">R/W-2h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="6">RESERVED</td><td>TX_AUTO_SEQUENCE</td><td>TX_MUX_SEL</td></tr>
<tr><td colspan="6">R-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td>TX_BYTE_SWAP</td><td>TX_EN_MODE</td><td>TX_AUTO_PREAMBLE</td><td>TX_ENABLE</td></tr>
<tr><td colspan="4">R-1h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
</table>

Table 30-272. TXCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	TX_CLK_DELAY	R/W	0h	In order to guarantee the MII_RT IO timing values published in the device data manual, the ocp_clk must be configured for 200MHz and TX_CLK_DELAY must be set to 6h.
27-26	RESERVED	R	0h	
25-16	TX_START_DELAY	R/W	2h	Defines the minimum time interval (delay) between receiving the RXDV for the current frame and the start of the transmit interface sending data to the MII interface. Delay value is in units of MII_RT clock cycles, which uses the ocp_clk (default is 200MHz, or 5ns). Default TX_START_DELAY value is 320ns, which is optimized for minimum latency at 16 bit processing. Counter is started with RX_DV signal going active. Transmit interface stops sending data when no more data is written into transmit interface by PRU along with TX_EOF marker bit set. If the TX FIFO has data when the delay expires, then TX will start sending data. But if the TX FIFO is empty, it will not start until the TX FIFO is not empty. It is possible to overflow the TX FIFO with the max delay setting when auto-forwarding is enabled since the time delay is larger than the amount of data it needs to store. As long as TX L1 FIFO overflows, software will need to issue a TX_RESET to reset the TX FIFO. The total delay is 64-byte times (size of TX FIFO), but you need to allow delays for synchronization. Do to this fact, the maximum delay should be 80ns less when auto forwarding is enabled. Therefore, 0x3F0 is the maximum in this configuration.
15-10	RESERVED	R	0h	

Table 30-272. TXCFG0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	TX_AUTO_SEQUENCE	R/W	0h	Enables transmit auto-sequence. Note the transmit data source is determined by TX_MUX_SEL setting. 0x 0: Disable 0x 1: Enable, transmit state machine based on events on receiver path that is connected to the respective transmitter. Also, the masking logic is disabled and only the MII data is used.
8	TX_MUX_SEL	R/W	0h	Selects transmit data source. The default/reset setting for TX Port 0 is 1. This setting permits MII TX Port 0 to receive data from PRU1 and the MII TX Port 1 which is connected to PRU0 by default. 0x 0: Data from PRU0 (default for TXCFG1) 0x 1: Data from PRU1 (default for TXCFG0)
7-4	RESERVED	R	1h	
3	TX_BYTE_SWAP	R/W	0h	Defines the order of Byte0/1 placement for TX R30. This bit must be selected/updated when the port is disabled or there is no traffic. 0x 0: R30 [15:8] = Byte1{Nibble3,Nibble2} R30[7:0] = Byte0{Nibble1,Nibble0} R30 [31:24] = TX_MASK [15:8] R30 [23:16] = TX_MASK [7:0] 0x 1: R30 [15:8] = Byte0{Nibble1,Nibble0} R30[7:0] = Byte1{Nibble3,Nibble2} R30 [31:24] = TX_MASK [7:0] R30 [23:16] = TX_MASK [15:8] Nibble0 is the first nibble received.
2	TX_EN_MODE	R/W	0h	Enables transmit self clear on TX_EOF event. Note that iep.cmp[3] must be set before transmission will start for TX0, and iep_cmp[4] for TX1. This is a new dependency, in addition to TX L1 FIFO not empty and TX_START_DELAY expiration, to start transmission. 0x 0: Disable 0x 1: Enable, TX_ENABLE will be clear for a TX_EOF event by itself.
1	TX_AUTO_PREAMBLE	R/W	0h	Transmit data auto-preamble. 0x 0: PRU will provide full preamble 0x 1: TX FIFO will insert pre-amble automatically. Note that the TX FIFO does not get preloaded with the preamble until the first write occurs. This can cause the latency to be larger than the minimum latency.
0	TX_ENABLE	R/W	0h	Enables transmit traffic on TX PORT. If TX_EN_MODE is set, then TX_ENABLE will self clear during a TX_EOF event. Note Software can use this to pre-fill the TX FIFO and then start the TX frame during non-ECS operations. 0x 0: TX PORT is disabled/stopped immediately 0x 1: TX PORT is enabled and the frame will start once the IPG counter expired and TX Start Delay counter has expired

30.5.7.4 TXCFG1 Register (Offset = 14h) [reset = 00400000h]

TXCFG1 is shown in Figure 30-277 and described in Table 30-273.

Return to Summary Table.

TX CONFIG1

Figure 30-277. TXCFG1 Register

31	30	29	28	27	26	25	24
RESERVED	TX_CLK_DELAY			RESERVED		TX_START_DELAY	
R-0h	R/W-0h			R-0h		R/W-40h	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TX_START_DELAY							
R/W-40h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED						TX_AUTO_SEQUENCE	TX_MUX_SEL
R-0h						R/W-0h	R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				TX_BYTE_SWAP	TX_EN_MODE	TX_AUTO_PREAMBLE	TX_ENABLE
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 30-273. TXCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	TX_CLK_DELAY	R/W	0h	In order to guarantee the MII_RT IO timing values published in the device data manual, the ocp_clk must be configured for 200MHz and TX_CLK_DELAY must be set to 6h.
27-26	RESERVED	R	0h	
25-16	TX_START_DELAY	R/W	40h	Defines the minimum time interval (delay) between receiving the RXDV for the current frame and the start of the transmit interface sending data to the MII interface. Delay value is in units of MII_RT clock cycles, which uses the ocp_clk (default is 200MHz, or 5ns). Default TX_START_DELAY value is 320ns, which is optimized for minimum latency at 16 bit processing. Counter is started with RX_DV signal going active. Transmit interface stops sending data when no more data is written into transmit interface by PRU along with TX_EOF marker bit set. If the TX FIFO has data when the delay expires, then TX will start sending data. But if the TX FIFO is empty, it will not start until the TX FIFO is not empty. It is possible to overflow the TX FIFO with the max delay setting when auto-forwarding is enabled since the time delay is larger than the amount of data it needs to store. As long as TX L1 FIFO overflows, software will need to issue a TX_RESET to reset the TX FIFO. The total delay is 64-byte times (size of TX FIFO), but you need to allow delays for synchronization. Do to this fact, the maximum delay should be 80ns less when auto forwarding is enabled. Therefore, 0x3F0 is the maximum in this configuration.
15-10	RESERVED	R	0h	

Table 30-273. TXCFG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	TX_AUTO_SEQUENCE	R/W	0h	Enables transmit auto-sequence. Note the transmit data source is determined by TX_MUX_SEL setting. 0x 0: Disable 0x 1: Enable, transmit state machine based on events on receiver path that is connected to the respective transmitter. Also, the masking logic is disabled and only the MII data is used.
8	TX_MUX_SEL	R/W	0h	Selects transmit data source. The default/reset setting for TX Port 0 is 1. This setting permits MII TX Port 0 to receive data from PRU1 and the MII TX Port 1 which is connected to PRU0 by default. 0x 0: Data from PRU0 (default for TXCFG1) 0x 1: Data from PRU1 (default for TXCFG0)
7-4	RESERVED	R	0h	
3	TX_BYTE_SWAP	R/W	0h	Defines the order of Byte0/1 placement for TX R30. This bit must be selected/updated when the port is disabled or there is no traffic. 0x 0: R30 [15:8] = Byte1{Nibble3,Nibble2} R30[7:0] = Byte0{Nibble1,Nibble0} R30 [31:24] = TX_MASK [15:8] R30 [23:16] = TX_MASK [7:0] 0x 1: R30 [15:8] = Byte0{Nibble1,Nibble0} R30[7:0] = Byte1{Nibble3,Nibble2} R30 [31:24] = TX_MASK [7:0] R30 [23:16] = TX_MASK [15:8] Nibble0 is the first nibble received.
2	TX_EN_MODE	R/W	0h	Enables transmit self clear on TX_EOF event. Note that iep.cmp[3] must be set before transmission will start for TX0, and iep_cmp[4] for TX1. This is a new dependency, in addition to TX L1 FIFO not empty and TX_START_DELAY expiration, to start transmission. 0x 0: Disable 0x 1: Enable, TX_ENABLE will be clear for a TX_EOF event by itself.
1	TX_AUTO_PREAMBLE	R/W	0h	Transmit data auto-preamble. 0x 0: PRU will provide full preamble 0x 1: TX FIFO will insert pre-amble automatically. Note that the TX FIFO does not get preloaded with the preamble until the first write occurs. This can cause the latency to be larger than the minimum latency.
0	TX_ENABLE	R/W	0h	Enables transmit traffic on TX PORT. If TX_EN_MODE is set, then TX_ENABLE will self clear during a TX_EOF event. Note Software can use this to pre-fill the TX FIFO and then start the TX frame during non-ECS operations. 0x 0: TX PORT is disabled/stopped immediately 0x 1: TX PORT is enabled and the frame will start once the IPG counter expired and TX Start Delay counter has expired

30.5.7.5 TXCRC0 Register (Offset = 20h) [reset = 0h]

TXCRC0 is shown in Figure 30-278 and described in Table 30-274.

Return to Summary Table.

TX CYCLIC REDUNDANCY CHECK0

Figure 30-278. TXCRC0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CRC32																															
R-0h																															

Table 30-274. TXCRC0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CRC32	R	0h	FCS (CRC32) data can be read by PRU for diagnostics. It is only valid after 6 clocks after a TX_CRC_HIGH command is given.

30.5.7.6 TXCRC1 Register (Offset = 24h) [reset = 0h]

TXCRC1 is shown in Figure 30-279 and described in Table 30-275.

Return to Summary Table.

TX CYCLIC REDUNDANCY CHECK1

Figure 30-279. TXCRC1 Register

31-0
TX_CRC32
R-0h

Table 30-275. TXCRC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CRC32	R	0h	FCS (CRC32) data can be read by PRU for diagnostics. It is only valid after 6 clocks after a TX_CRC_HIGH command is given.

30.5.7.7 TXIPG0 Register (Offset = 30h) [reset = 28h]

TXIPG0 is shown in Figure 30-280 and described in Table 30-276.

Return to Summary Table.

TX INTERPACKET GAP0

Figure 30-280. TXIPG0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="22">RESERVED</td><td colspan="10">TX_IPG</td></tr>
<tr><td colspan="22">R-0h</td><td colspan="10">R/W-28h</td></tr>
</table>

Table 30-276. TXIPG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9-0	TX_IPG	R/W	28h	Defines the minimum of transmit Inter Packet Gap (IPG) which is the number of ocp_clk cycles between the de-assertion of TX_EN and the assertion of TX_EN. The start of the TX will get delayed when the incoming packet IPG is less than defined minimum value. In general, software should program in increments of 8, 40ns to insure the extra delays takes effect.

30.5.7.8 TXIPG1 Register (Offset = 34h) [reset = 28h]

TXIPG1 is shown in Figure 30-281 and described in Table 30-277.

Return to Summary Table.

TX INTERPACKET GAP1

Figure 30-281. TXIPG1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="22">RESERVED</td><td colspan="10">TX_IPG</td></tr>
<tr><td colspan="22">R-0h</td><td colspan="10">R/W-28h</td></tr>
</table>

Table 30-277. TXIPG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9-0	TX_IPG	R/W	28h	Defines the minimum of transmit Inter Packet Gap (IPG) which is the number of ocp_clk cycles between the de-assertion of TX_EN and the assertion of TX_EN. The start of the TX will get delayed when the incoming packet IPG is less than defined minimum value. In general, software should program in increments of 8, 40ns to insure the extra delays takes effect.

30.5.7.9 PRS0 Register (Offset = 38h) [reset = 0h]

PRS0 is shown in Figure 30-282 and described in Table 30-278.

Return to Summary Table.

PORT RAW STATUS0

Figure 30-282. PRS0 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						pr1_mii0_crs	pr1_mii0_col
R-0h						R-0h	R-0h

Table 30-278. PRS0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	pr1_mii0_crs	R	0h	Current state of pr<k>_mii0_crs
0	pr1_mii0_col	R	0h	Current state of pr<k>_mii0_col

30.5.7.10 PRS1 Register (Offset = 3Ch) [reset = 0h]

PRS1 is shown in Figure 30-283 and described in Table 30-279.

Return to Summary Table.

PORT RAW STATUS1

Figure 30-283. PRS1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">RESERVED</td><td>pr1_mii1_crs</td><td>pr1_mii1_col</td></tr>
<tr><td colspan="6">R-0h</td><td>R-0h</td><td>R-0h</td></tr>
</table>

Table 30-279. PRS1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	pr1_mii1_crs	R	0h	Current state of pr<k>_mii1_crs
0	pr1_mii1_col	R	0h	Current state of pr<k>_mii1_col

30.5.7.11 RXFRMS0 Register (Offset = 40h) [reset = 05F1003Fh]

RXFRMS0 is shown in Figure 30-284 and described in Table 30-280.

Return to Summary Table.

RX FRAME SIZE0

Figure 30-284. RXFRMS0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RX_MAX_FRM_CNT</td><td colspan="16">RX_MIN_FRM_CNT</td></tr>
<tr><td colspan="16">R/W-5F1h</td><td colspan="16">R/W-3Fh</td></tr>
</table>

Table 30-280. RXFRMS0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RX_MAX_FRM_CNT	R/W	5F1h	Defines the maximum received frame count. If the total byte count of the received frame is more than defined value, RX_MAX_FRM_ERR will get set. 0x 0: 1 byte after SFD and including CRC. N: N+1 bytes after SFD and including CRC Note if the incoming frame is truncated at the marker, RX_CRC and RX_NIBBLE_ODD will not get asserted.
15-0	RX_MIN_FRM_CNT	R/W	3Fh	Defines the minimum received frame count. If the total byte count of received frame is less than defined value, RX_MIN_FRM_ERR will get set. 0x 0: 1 byte after SFD and including CRC. N: N+1 bytes after SFD and including CRC

30.5.7.12 RXFRMS1 Register (Offset = 44h) [reset = 05F1003Fh]

RXFRMS1 is shown in Figure 30-285 and described in Table 30-281.

Return to Summary Table.

RX FRAME SIZE1

Figure 30-285. RXFRMS1 Register

31-16	15-0
RX_MAX_FRM_CNT	RX_MIN_FRM_CNT
R/W-5F1h	R/W-3Fh

Table 30-281. RXFRMS1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RX_MAX_FRM_CNT	R/W	5F1h	Defines the maximum received frame count. If the total byte count of the received frame is more than defined value, RX_MAX_FRM_ERR will get set. 0x 0: 1 byte after SFD and including CRC. N: N+1 bytes after SFD and including CRC Note if the incoming frame is truncated at the marker, RX_CRC and RX_NIBBLE_ODD will not get asserted.
15-0	RX_MIN_FRM_CNT	R/W	3Fh	Defines the minimum received frame count. If the total byte count of received frame is less than defined value, RX_MIN_FRM_ERR will get set. 0x 0: 1 byte after SFD and including CRC. N: N+1 bytes after SFD and including CRC

30.5.7.13 RXPCNT0 Register (Offset = 48h) [reset = E1h]

RXPCNT0 is shown in Figure 30-286 and described in Table 30-282.

Return to Summary Table.

RX PREAMABLE COUNT0

Figure 30-286. RXPCNT0 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RX_MAX_PRE_CNT				RX_MIN_PRE_CNT			
R/W-Eh				R/W-1h			

Table 30-282. RXPCNT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-4	RX_MAX_PRE_CNT	R/W	Eh	Defines the maximum number of nibbles until the start of frame delimiter (SFD) event occurred (i.e. matches 0x5D). RX_MAX_PRE_COUNT_ERR will be set if the preamble counts more than the value of RX_MAX_PRE_CNT. If the SFD does not occur within 16 nibbles, the error will assert and the incoming frame will be truncated. 0x 0: Disabled 0x 1: Reserved 0x 2: 4th nibble needs to have built 0x5D 0xe: 16th nibble needs to have built 0x5D Note the 16th nibble is transmitted.
3-0	RX_MIN_PRE_CNT	R/W	1h	Defines the minimum number of nibbles until the start of frame delimiter (SFD) event occurred, which is matched the value 0x5D. RX_MIN_PRE_COUNT_ERR will be set if the preamble counts less than the value of RX_MIN_PRE_CNT. 0x 0: Disabled 0x 1: 1 0x5 before 0x5D 0x 2: 2 0x5 before 0x5D N: N 0x5 before 0x5D Note it does not need to be 0x5.

30.5.7.14 RXPCNT1 Register (Offset = 4Ch) [reset = E1h]

RXPCNT1 is shown in Figure 30-287 and described in Table 30-283.

Return to Summary Table.

RX PREAMABLE COUNT1

Figure 30-287. RXPCNT1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RX_MAX_PRE_CNT				RX_MIN_PRE_CNT			
R/W-Eh				R/W-1h			

Table 30-283. RXPCNT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-4	RX_MAX_PRE_CNT	R/W	Eh	Defines the maximum number of nibbles until the start of frame delimiter (SFD) event occurred (i.e. matches 0x5D). RX_MAX_PRE_COUNT_ERR will be set if the preamble counts more than the value of RX_MAX_PRE_CNT. If the SFD does not occur within 16 nibbles, the error will assert and the incoming frame will be truncated. 0x 0: Disabled 0x 1: Reserved 0x 2: 4th nibble needs to have built 0x5D 0xe: 16th nibble needs to have built 0x5D Note the 16th nibble is transmitted.
3-0	RX_MIN_PRE_CNT	R/W	1h	Defines the minimum number of nibbles until the start of frame delimiter (SFD) event occurred, which is matched the value 0x5D. RX_MIN_PRE_COUNT_ERR will be set if the preamble counts less than the value of RX_MIN_PRE_CNT. 0x 0: Disabled 0x 1: 1 0x5 before 0x5D 0x 2: 2 0x5 before 0x5D N: N 0x5 before 0x5D Note it does not need to be 0x5.

30.5.7.15 RXERR0 Register (Offset = 50h) [reset = 0h]

RXERR0 is shown in Figure 30-288 and described in Table 30-284.

Return to Summary Table.

RX ERROR0

Figure 30-288. RXERR0 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RX_MAX_FRM_CNT_ERR	RX_MIN_FRM_CNT_ERR	RX_MAX_PRE_CNT_ERR	RX_MIN_PRE_CNT_ERR
R-0h				R-0h	R-0h	R-0h	R-0h

Table 30-284. RXERR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	RX_MAX_FRM_CNT_ERR	R	0h	Error status of received frame is more than the value of RX_MAX_FRM_CNT. 0x 0: No error occurred 0x 1: Error occurred
2	RX_MIN_FRM_CNT_ERR	R	0h	Error status of received frame is less than the value of RX_MIN_FRM_CNT. 0x 0: No error occurred 0x 1: Error occurred
1	RX_MAX_PRE_CNT_ERR	R	0h	Error status of received preamble nibble is more than the value of RX_MAX_PRE_CNT. 0x 0: No error occurred 0x 1: Error occurred
0	RX_MIN_PRE_CNT_ERR	R	0h	Error status of received preamble nibble is less than the value of RX_MIN_PRE_CNT. 0x 0: No error occurred 0x 1: Error occurred

30.5.7.16 RXERR1 Register (Offset = 54h) [reset = 0h]

RXERR1 is shown in Figure 30-289 and described in Table 30-285.

Return to Summary Table.

RX ERROR1

Figure 30-289. RXERR1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RX_MAX_FRM_CNT_ERR	RX_MIN_FRM_CNT_ERR	RX_MAX_PRE_CNT_ERR	RX_MIN_PRE_CNT_ERR
R-0h				R-0h	R-0h	R-0h	R-0h

Table 30-285. RXERR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	RX_MAX_FRM_CNT_ERR	R	0h	Error status of received frame is more than the value of RX_MAX_FRM_CNT. 0x 0: No error occurred 0x 1: Error occurred
2	RX_MIN_FRM_CNT_ERR	R	0h	Error status of received frame is less than the value of RX_MIN_FRM_CNT. 0x 0: No error occurred 0x 1: Error occurred
1	RX_MAX_PRE_CNT_ERR	R	0h	Error status of received preamble nibble is more than the value of RX_MAX_PRE_CNT. 0x 0: No error occurred 0x 1: Error occurred
0	RX_MIN_PRE_CNT_ERR	R	0h	Error status of received preamble nibble is less than the value of RX_MIN_PRE_CNT. 0x 0: No error occurred 0x 1: Error occurred

30.5.7.17 RXFLV0 Register (Offset = 60h) [reset = 0h]

RXFLV0 is shown in Figure 30-290 and described in Table 30-286.

Return to Summary Table.

RX FIFO0 Level

Figure 30-290. RXFLV0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">RX_FIFO_LEVEL</td></tr>
<tr><td colspan="8">R-0h</td><td colspan="8">R-0h</td></tr>
</table>

Table 30-286. RXFLV0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	RX_FIFO_LEVEL	R	0h	Define the number of valid bytes in the RX FIFO 0 = empty 1 = 1 Byte/ 2 Nibbles 2 = 2 Byte/ 4 Nibble ... 32 = 32 Bytes/ 64 Nibbles

30.5.7.18 RXFLV1 Register (Offset = 64h) [reset = 0h]

RXFLV1 is shown in Figure 30-291 and described in Table 30-287.

Return to Summary Table.

RX FIFO1 Level

Figure 30-291. RXFLV1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RX_FIFO_LEVEL							
R-0h								R-0h							

Table 30-287. RXFLV1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	RX_FIFO_LEVEL	R	0h	Define the number of valid bytes in the RX FIFO 0 = empty 1 = 1 Byte/ 2 Nibbles 2 = 2 Byte/ 4 Nibble ... 32 = 32 Bytes/ 64 Nibbles

30.5.7.19 TXFLV0 Register (Offset = 68h) [reset = X]

TXFLV0 is shown in Figure 30-292 and described in Table 30-288.

Return to Summary Table.

TX FIFO0 Level

Figure 30-292. TXFLV0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-X</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">TX_FIFO_LEVEL</td></tr>
<tr><td colspan="8">R-X</td><td colspan="8">R-0h</td></tr>
</table>

Table 30-288. TXFLV0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	X	
7-0	TX_FIFO_LEVEL	R	0h	Define the number of valid nibbles in the TX FIFO 0 = empty 1 = 1 Nibble 2 = 1 Byte/ 2 Nibble ... 128 = 64 Bytes/ 128 Nibbles

30.5.7.20 TXFLV1 Register (Offset = 6Ch) [reset = X]

TXFLV1 is shown in Figure 30-293 and described in Table 30-289.

Return to Summary Table.

TX FIFO1 Level

Figure 30-293. TXFLV1 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R-X</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">TX_FIFO_LEVEL</td></tr>
<tr><td colspan="8">R-X</td><td colspan="8">R-0h</td></tr>
</table>

Table 30-289. TXFLV1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	X	
7-0	TX_FIFO_LEVEL	R	0h	Define the number of valid nibbles in the TX FIFO 0 = empty 1 = 1 Nibble 2 = 1 Byte/ 2 Nibble ... 128 = 64 Bytes/ 128 Nibbles

30.5.8 PRU_ICSS_MDIO Registers

For additional details about the MDIO registers, see *MDIO Registers* in the AM437x TRM (SPRUHL7).

30.5.9 PRU_ICSS_CFG Registers

Table 30-290 lists the memory-mapped registers for the PRU_ICSS_CFG. All register offset addresses not listed in Table 30-290 should be considered as reserved locations and the register contents should not be modified.

Table 30-290. PRU_ICSS_CFG Registers

Offset	Acronym	Register Name	Section
0h	PRU_ICSS_CFG_REVID		Section 30.5.9.1
4h	PRU_ICSS_CFG_SYSCFG		Section 30.5.9.2
8h	PRU_ICSS_CFG_GPCFG0		Section 30.5.9.3
Ch	PRU_ICSS_CFG_GPCFG1		Section 30.5.9.4
10h	PRU_ICSS_CFG_CGR		Section 30.5.9.5
14h	PRU_ICSS_CFG_ISRP		Section 30.5.9.6
18h	PRU_ICSS_CFG_ISP		Section 30.5.9.7
1Ch	PRU_ICSS_CFG_IESP		Section 30.5.9.8
20h	PRU_ICSS_CFG_IECP		Section 30.5.9.9
28h	PRU_ICSS_CFG_PMAO		Section 30.5.9.10
2Ch	PRU_ICSS_CFG_MII_RT		Section 30.5.9.11
30h	PRU_ICSS_CFG_IEPCLK		Section 30.5.9.12
34h	PRU_ICSS_CFG_SPP		Section 30.5.9.13
40h	PRU_ICSS_CFG_PIN_MX		Section 30.5.9.14
48h + formula	PRU_ICSS_CFG_SD_P0_CLK_i	PRU0 SD0-8 CLK/ACC Select	Section 30.5.9.15
4Ch + formula	PRU_ICSS_CFG_SD_P0_SS_i	PRU0 SD0-8 Sample Size	Section 30.5.9.16

Table 30-290. PRU_ICSS_CFG Registers (continued)

Offset	Acronym	Register Name	Section
94h + formula	PRU_ICSS_CFG_SD_P1_CLK_i	PRU1 SD0-8 CLK/ACC Select	Section 30.5.9.17
98h + formula	PRU_ICSS_CFG_SD_P1_SS_i	PRU1 SD0-8 Sample Size	Section 30.5.9.18
E0h	PRU_ICSS_CFG_ED_P0_RXCFG	PRU0 ED Receive Global Configuration	Section 30.5.9.19
E4h	PRU_ICSS_CFG_ED_P0_TXCFG	PRU0 ED Transmit Global S Configuration	Section 30.5.9.20
E8h + formula	PRU_ICSS_CFG_ED_P0_CFG0_i	PRU0 ED Channel0-2 CFG0	Section 30.5.9.21
ECh + formula	PRU_ICSS_CFG_ED_P0_CFG1_i	PRU0 ED Channel0-2 CFG1	Section 30.5.9.22
100h	PRU_ICSS_CFG_ED_P1_RXCFG	PRU1 ED Receive Global Configuration	Section 30.5.9.23
104h	PRU_ICSS_CFG_ED_P1_TXCFG	PRU1 ED Transmit Global S Configuration	Section 30.5.9.24
108h + formula	PRU_ICSS_CFG_ED_P1_CFG0_i	PRU1 ED Channel0-2 CFG0	Section 30.5.9.25
10Ch + formula	PRU_ICSS_CFG_ED_P1_CFG1_i	PRU1 ED Channel0-2 CFG1	Section 30.5.9.26

30.5.9.1 PRU_ICSS_CFG_REVID Register (Offset = 0h) [reset = 47000000h]

PRU_ICSS_CFG_REVID is shown in Figure 30-294 and described in Table 30-291.

Return to Summary Table.

The Revision Register contains the ID and revision information.

Figure 30-294. PRU_ICSS_CFG_REVID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVID																															
R-47000000h																															

Table 30-291. PRU_ICSS_CFG_REVID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REVID	R	47000000h	Revision ID. Reset value for PRU-ICSS1 is 0x4700_0200 and for PRU-ICSS0 is 0x4701_0100.

30.5.9.2 PRU_ICSS_CFG_SYSCFG Register (Offset = 4h) [reset = 1Ah]

PRU_ICSS_CFG_SYSCFG is shown in Figure 30-295 and described in Table 30-292.

Return to Summary Table.

The System Configuration Register defines the power IDLE and STANDBY modes.
Note this register is only available for PRU-ICSS1. In PRU-ICSS0, it is reserved.

Figure 30-295. PRU_ICSS_CFG_SYSCFG Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED		SUB_MWAIT	STANDBY_INIT	STANDBY_MODE		IDLE_MODE	
R/W-0h		R-0h	R/W-1h	R/W-2h		R/W-2h	

Table 30-292. PRU_ICSS_CFG_SYSCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5	SUB_MWAIT	R	0h	Status bit for wait state. 0h (R/W) = Ready for Transaction 1h (R/W) = Wait until 0
4	STANDBY_INIT	R/W	1h	0h (R/W) = Enable OCP master ports. 1h (R/W) = Initiate standby sequence.
3-2	STANDBY_MODE	R/W	2h	Status bit for standby mode. 0h (R/W) = Force standby mode: Initiator unconditionally in standby (standby = 1). 1h (R/W) = No standby mode: Initiator unconditionally out of standby (standby = 0). 2h (R/W) = Smart standby mode: Standby requested by initiator depending on internal conditions. 3h (R/W) = Reserved.
1-0	IDLE_MODE	R/W	2h	Status bit for idle mode. 0h (R/W) = Force-idle mode. 1h (R/W) = No-idle mode. 2h (R/W) = Smart-idle mode. 3h (R/W) = Reserved.

30.5.9.3 PRU_ICSS_CFG_GPCFG0 Register (Offset = 8h) [reset = 0h]

PRU_ICSS_CFG_GPCFG0 is shown in Figure 30-296 and described in Table 30-293.

Return to Summary Table.

The General Purpose Configuration 0 Register defines the GPI/O configuration for PRU0.

Figure 30-296. PRU_ICSS_CFG_GPCFG0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="4">RESERVED</td><td colspan="2">PR1_PRU0_GP_MUX_SEL</td><td>PRU0_GPO_SH_SEL</td><td>PRU0_GPO_DIV1</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="2">R-0h</td><td>R-0h</td><td>R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="4">PRU0_GPO_DIV1</td><td colspan="4">PRU0_GPO_DIV0</td></tr>
<tr><td colspan="4">R/W-0h</td><td colspan="4">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>PRU0_GPO_DIV0</td><td>PRU0_GPO_MODE</td><td>PRU0_GPI_SB</td><td colspan="5">PRU0_GPI_DIV1</td></tr>
<tr><td>R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td colspan="5">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">PRU0_GPI_DIV0</td><td>PRU0_GPI_CLK_MODE</td><td colspan="2">PRU0_GPI_MODE</td></tr>
<tr><td colspan="5">R/W-0h</td><td>R/W-0h</td><td colspan="2">R/W-0h</td></tr>
</table>

Table 30-293. PRU_ICSS_CFG_GPCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-26	PR1_PRU0_GP_MUX_SEL	R	0h	Controls the PRU-ICSS wrap mux selection 0h (R/W) = GP selected 1h (R/W) = ENDAT mode 2h (R/W) = Reserved. 3h (R/W) = SD mode
25	PRU0_GPO_SH_SEL	R	0h	Defines which shadow register is currently getting used for GPO shifting. 0h (R/W) = gpo_sh0 is selected 1h (R/W) = gpo_sh1 is selected
24-20	PRU0_GPO_DIV1	R/W	0h	Divisor value (divide by PRU0_GPO_DIV1 + 1). 0h = div 1.0. 1h = div 1.5. 2h = div 2.0. .. 1eh = div 16.0. 1fh = reserved.
19-15	PRU0_GPO_DIV0	R/W	0h	Divisor value (divide by PRU0_GPO_DIV0 + 1). 0h = div 1.0. 1h = div 1.5. 2h = div 2.0. .. 1eh = div 16.0. 1fh = reserved.
14	PRU0_GPO_MODE	R/W	0h	PRU GPO (R30) modes: 0h (R/W) = Direct output mode 1h (R/W) = Serial output mode

Table 30-293. PRU_ICSS_CFG_GPCFG0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	PRU0_GPI_SB	R/W	0h	Start Bit event for 28-bit shift in mode. PRU0_GPI_SB (pru0_r31_status[29]) is set when first capture of a 1 on pru0_r31_status[0]. 0h (W) = No Effect. 0h (R) = Start Bit event has not occurred. 1h (W) = Will clear PRU0_GPI_SB and clear the whole shift register. 1h (R) = Start Bit event occurred.
12-8	PRU0_GPI_DIV1	R/W	0h	Divisor value (divide by PRU0_GPI_DIV1 + 1). 0h = div 1.0. 1h = div 1.5. 2h = div 2.0. .. 1eh = div 16.0. 1fh = reserved.
7-3	PRU0_GPI_DIV0	R/W	0h	Divisor value (divide by PRU0_GPI_DIV0 + 1). 0h = div 1.0. 1h = div 1.5. 2h = div 2.0. .. 1eh = div 16.0. 1fh = reserved.
2	PRU0_GPI_CLK_MODE	R/W	0h	Parallel 16bit capture mode clock edge. 0h (R/W) = Use the positive edge of pru0_r31_status[16] 1h (R/W) = Use the negative edge of pru0_r31_status[16]
1-0	PRU0_GPI_MODE	R/W	0h	PRU GPI (R31) modes: 0h (R/W) = Direct input mode 1h (R/W) = 16bit parallel capture mode. 2h (R/W) = 28bit shift in mode 3h (R/W) = Mii_rt mode

30.5.9.4 PRU_ICSS_CFG_GPCFG1 Register (Offset = Ch) [reset = 0h]

PRU_ICSS_CFG_GPCFG1 is shown in Figure 30-297 and described in Table 30-294.

Return to Summary Table.

The General Purpose Configuration 1 Register defines the GPI/O configuration for PRU1.

Figure 30-297. PRU_ICSS_CFG_GPCFG1 Register

31	30	29	28	27	26	25	24
RESERVED				PR1_PRU1_GP_MUX_SEL		PRU1_GPO_SH_SEL	PRU1_GPO_DIV1
R/W-0h				R-0h		R-0h	R/W-0h
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRU1_GPO_DIV1				PRU1_GPO_DIV0			
R/W-0h				R/W-0h			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PRU1_GPO_DIV0	PRU1_GPO_MODE	PRU1_GPI_SB	PRU1_GPI_DIV1				
R/W-0h	R/W-0h	R/W1C-0h	R/W-0h				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRU1_GPI_DIV0					PRU1_GPI_CLK_MODE	PRU1_GPI_MODE	
R/W-0h					R/W-0h	R/W-0h	

Table 30-294. PRU_ICSS_CFG_GPCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-26	PR1_PRU1_GP_MUX_SEL	R	0h	Controls the PRU-ICSS wrap mux selection 0h (R/W) = GP selected 1h (R/W) = ENDAT mode 2h (R/W) = Reserved. 3h (R/W) = SD mode
25	PRU1_GPO_SH_SEL	R	0h	Defines which shadow register is currently getting used for GPO shifting. 0h (R/W) = gpo_sh0 is selected 1h (R/W) = gpo_sh1 is selected
24-20	PRU1_GPO_DIV1	R/W	0h	Divisor value (divide by PRU1_GPO_DIV1 + 1). 0h = div 1.0. 1h = div 1.5. 2h = div 2.0. .. 1eh = div 16.0. 1fh = reserved.
19-15	PRU1_GPO_DIV0	R/W	0h	Divisor value (divide by PRU1_GPO_DIV0 + 1). 0h = div 1.0. 1h = div 1.5. 2h = div 2.0. .. 1eh = div 16.0. 1fh = reserved.
14	PRU1_GPO_MODE	R/W	0h	PRU GPO (R30) modes: 0h (R/W) = Direct output mode 1h (R/W) = Serial output mode

Table 30-294. PRU_ICSS_CFG_GPCFG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	PRU1_GPI_SB	R/W1C	0h	28-bit shift in mode Start Bit event. PRU1_GPI_SB (pru1_r31_status[29]) is set when first capture of a 1 on pru1_r31_status[0]. 0h (W) = No Effect. 0h (R) = Start Bit event has not occurred. 1h (W) = Will clear PRU1_GPI_SB and clear the whole shift register. 1h (R) = Start Bit event occurred.
12-8	PRU1_GPI_DIV1	R/W	0h	Divisor value (divide by PRU1_GPI_DIV1 + 1). 0h = div 1.0. 1h = div 1.5. 2h = div 2.0. .. 1eh = div 16.0. 1fh = reserved.
7-3	PRU1_GPI_DIV0	R/W	0h	Divisor value (divide by PRU1_GPI_DIV0 + 1). 0h = div 1.0. 1h = div 1.5. 2h = div 2.0. .. 1eh = div 16.0. 1fh = reserved.
2	PRU1_GPI_CLK_MODE	R/W	0h	Parallel 16bit capture mode clock edge. 0h (R/W) = Use the positive edge of pru1_r31_status[16] 1h (R/W) = Use the negative edge of pru1_r31_status[16]
1-0	PRU1_GPI_MODE	R/W	0h	PRU GPI (R31) modes: 0h (R/W) = Direct input mode 1h (R/W) = 16bit parallel capture mode. 2h (R/W) = 28bit shift in mode 3h (R/W) = Mii_rt mode

30.5.9.5 PRU_ICSS_CFG_CGR Register (Offset = 10h) [reset = 00024924h]

PRU_ICSS_CFG_CGR is shown in Figure 30-298 and described in Table 30-295.

Return to Summary Table.

The Clock Gating Register controls the state of Clock Management of the different modules. Software should not clear {module}_CLK_EN until {module}_CLK_STOP_ACK is 0x1.

Figure 30-298. PRU_ICSS_CFG_CGR Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED						IEP_CLK_EN	IEP_CLK_STOP_ACK
R/W-0h						R/W-1h	R-0h
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
IEP_CLK_STOP_REQ	ECAP_CLK_EN	ECAP_CLK_STOP_ACK	ECAP_CLK_STOP_REQ	UART_CLK_EN	UART_CLK_STOP_ACK	UART_CLK_STOP_REQ	INTC_CLK_EN
R/W-0h	R/W-1h	R-0h	R/W-0h	R/W-1h	R-0h	R/W-0h	R/W-1h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTC_CLK_STOP_ACK	INTC_CLK_STOP_REQ	PRU1_CLK_EN	PRU1_CLK_STOP_ACK	PRU1_CLK_STOP_REQ	PRU0_CLK_EN	PRU0_CLK_STOP_ACK	PRU0_CLK_STOP_REQ
R-0h	R/W-0h	R/W-1h	R-0h	R/W-0h	R/W-1h	R-0h	R/W-0h

Table 30-295. PRU_ICSS_CFG_CGR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17	IEP_CLK_EN	R/W	1h	IEP clock enable. 0h (R/W) = Disable Clock 1h (R/W) = Enable Clock
16	IEP_CLK_STOP_ACK	R	0h	Acknowledgement that IEP clock can be stopped. 0h (R/W) = Not Ready to Gate Clock 1h (R/W) = Ready to Gate Clock
15	IEP_CLK_STOP_REQ	R/W	0h	IEP request to stop clock. 0h (R/W) = Do not request to stop Clock 1h (R/W) = Request to stop Clock
14	ECAP_CLK_EN	R/W	1h	ECAP clock enable. 0h (R/W) = Disable Clock 1h (R/W) = Enable Clock
13	ECAP_CLK_STOP_ACK	R	0h	Acknowledgement that ECAP clock can be stopped. 0h (R/W) = Not Ready to Gate Clock 1h (R/W) = Ready to Gate Clock
12	ECAP_CLK_STOP_REQ	R/W	0h	ECAP request to stop clock. 0h (R/W) = Do not request to stop Clock 1h (R/W) = Request to stop Clock
11	UART_CLK_EN	R/W	1h	UART clock enable. 0h (R/W) = Disable Clock 1h (R/W) = Enable Clock
10	UART_CLK_STOP_ACK	R	0h	Acknowledgement that UART clock can be stopped. 0h (R/W) = Not Ready to Gate Clock 1h (R/W) = Ready to Gate Clock

Table 30-295. PRU_ICSS_CFG_CGR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	UART_CLK_STOP_REQ	R/W	0h	UART request to stop clock. 0h (R/W) = Do not request to stop Clock 1h (R/W) = Request to stop Clock
8	INTC_CLK_EN	R/W	1h	INTC clock enable. 0h (R/W) = Disable Clock 1h (R/W) = Enable Clock
7	INTC_CLK_STOP_ACK	R	0h	Acknowledgement that INTC clock can be stopped. 0h (R/W) = Not Ready to Gate Clock 1h (R/W) = Ready to Gate Clock
6	INTC_CLK_STOP_REQ	R/W	0h	INTC request to stop clock. 0h (R/W) = Do not request to stop Clock 1h (R/W) = Request to stop Clock
5	PRU1_CLK_EN	R/W	1h	PRU1 clock enable. 0h (R/W) = Disable Clock 1h (R/W) = Enable Clock
4	PRU1_CLK_STOP_ACK	R	0h	Acknowledgement that PRU1 clock can be stopped. 0h (R/W) = Not Ready to Gate Clock 1h (R/W) = Ready to Gate Clock
3	PRU1_CLK_STOP_REQ	R/W	0h	PRU1 request to stop clock. 0h (R/W) = Do not request to stop Clock 1h (R/W) = Request to stop Clock
2	PRU0_CLK_EN	R/W	1h	PRU0 clock enable. 0h (R/W) = Disable Clock 1h (R/W) = Enable Clock
1	PRU0_CLK_STOP_ACK	R	0h	Acknowledgement that PRU0 clock can be stopped. 0h (R/W) = Not Ready to Gate Clock 1h (R/W) = Ready to Gate Clock
0	PRU0_CLK_STOP_REQ	R/W	0h	PRU0 request to stop clock. 0h (R/W) = Do not request to stop Clock 1h (R/W) = Request to stop Clock

30.5.9.6 PRU_ICSS_CFG_ISRP Register (Offset = 14h) [reset = 0h]

PRU_ICSS_CFG_ISRP is shown in Figure 30-299 and described in Table 30-296.

Return to Summary Table.

The IRQ Status Raw Parity register is a snapshot of the IRQ raw status for the PRU ICSS memory parity events. The raw status is set even if the event is not enabled.

Figure 30-299. PRU_ICSS_CFG_ISRP Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				RAM_PE_RAW			
R/W-0h				R/W-0h			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PRU1_DMEM_PE_RAW				PRU1_IMEM_PE_RAW			
R/W-0h				R/W-0h			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRU0_DMEM_PE_RAW				PRU0_IMEM_PE_RAW			
R/W-0h				R/W-0h			

Table 30-296. PRU_ICSS_CFG_ISRP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19-16	RAM_PE_RAW	R/W	0h	PRU-ICSS1 Only. RAM Parity Error RAW for Byte3, Byte2, Byte1, Byte0. Note RAM_PE_RAW[0] maps to Byte0. 0h (W) = No action 0h (R) = No event pending 1h (W) = Set event (debug) 1h (R) = Event pending
15-12	PRU1_DMEM_PE_RAW	R/W	0h	PRU1 DMEM Parity Error RAW for Byte3, Byte2, Byte1, Byte0. Note PRU1_DMEM_PE_RAW[0] maps to Byte0. 0h (W) = No action 0h (R) = No event pending 1h (W) = Set event (debug) 1h (R) = Event pending
11-8	PRU1_IMEM_PE_RAW	R/W	0h	PRU1 IMEM Parity Error RAW for Byte3, Byte2, Byte1, Byte0. Note PRU1_IMEM_PE_RAW[0] maps to Byte0. 0h (W) = No action 0h (R) = No event pending 1h (W) = Set event (debug) 1h (R) = Event pending
7-4	PRU0_DMEM_PE_RAW	R/W	0h	PRU0 DMEM Parity Error RAW for Byte3, Byte2, Byte1, Byte0. Note PRU0_DMEM_PE_RAW[0] maps to Byte0. 0h (W) = No action 0h (R) = No event pending 1h (W) = Set event (debug) 1h (R) = Event pending
3-0	PRU0_IMEM_PE_RAW	R/W	0h	PRU0 IMEM Parity Error RAW for Byte3, Byte2, Byte1, Byte0. Note PRU0_IRAM_PE_RAW[0] maps to Byte0. 0h (W) = No action 0h (R) = No event pending 1h (W) = Set event (debug) 1h (R) = Event pending

30.5.9.7 PRU_ICSS_CFG_ISP Register (Offset = 18h) [reset = 0h]

PRU_ICSS_CFG_ISP is shown in Figure 30-300 and described in Table 30-297.

Return to Summary Table.

The IRQ Status Parity Register is a snapshot of the IRQ status for the PRU ICSS memory parity events. The status is set only if the event is enabled. Write 1 to clear the status after the interrupt has been serviced.

Figure 30-300. PRU_ICSS_CFG_ISP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED												RAM_PE			
R-0h												0h			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRU1_DMEM_PE				PRU1_IMEM_PE				PRU0_DMEM_PE				PRU0_IMEM_PE			
0h				0h				0h				0h			

Table 30-297. PRU_ICSS_CFG_ISP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-16	RAM_PE		0h	PRU-ICSS1 Only. RAM Parity Error for Byte3, Byte2, Byte1, Byte0. Note RAM_PE[0] maps to Byte0. 0h (W) = No action 0h (R) = No (enabled) event pending 1h (W) = Clear event 1h (R) = Event pending
15-12	PRU1_DMEM_PE		0h	PRU1 DMEM Parity Error for Byte3, Byte2, Byte1, Byte0. Note PRU1_DMEM_PE[0] maps to Byte0. 0h (W) = No action 0h (R) = No (enabled) event pending 1h (W) = Clear event 1h (R) = Event pending
11-8	PRU1_IMEM_PE		0h	PRU1 IMEM Parity Error for Byte3, Byte2, Byte1, Byte0. Note PRU1_IMEM_PE[0] maps to Byte0. 0h (W) = No action 0h (R) = No (enabled) event pending 1h (W) = Clear event 1h (R) = Event pending
7-4	PRU0_DMEM_PE		0h	PRU0 DMEM Parity Error for Byte3, Byte2, Byte1, Byte0. Note PRU0_DMEM_PE[0] maps to Byte0. 0h (W) = No action 0h (R) = No (enabled) event pending 1h (W) = Clear event 1h (R) = Event pending
3-0	PRU0_IMEM_PE		0h	PRU0 IMEM Parity Error for Byte3, Byte2, Byte1, Byte0. Note PRU0_IMEM_PE[0] maps to Byte0. 0h (W) = No action 0h (R) = No (enabled) event pending 1h (W) = Clear event 1h (R) = Event pending

30.5.9.8 PRU_ICSS_CFG_IESP Register (Offset = 1Ch) [reset = 0h]

PRU_ICSS_CFG_IESP is shown in Figure 30-301 and described in Table 30-298.

Return to Summary Table.

The IRQ Enable Set Parity Register enables the IRQ PRU ICSS memory parity events.

Figure 30-301. PRU_ICSS_CFG_IESP Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED				RAM_PE_SET			
R/W-0h				R/W-0h			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PRU1_DMEM_PE_SET				PRU1_IMEM_PE_SET			
R/W-0h				R/W-0h			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRU0_DMEM_PE_SET				PRU0_IMEM_PE_SET			
R/W-0h				R/W-0h			

Table 30-298. PRU_ICSS_CFG_IESP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19-16	RAM_PE_SET	R/W	0h	PRU-ICSS1 Only. RAM Parity Error Set Enable for Byte3, Byte2, Byte1, Byte0. Note RAM_PE_SET[0] maps to Byte0. 0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Enable interrupt 1h (R) = Interrupt enabled
15-12	PRU1_DMEM_PE_SET	R/W	0h	PRU1 DMEM Parity Error Set Enable for Byte3, Byte2, Byte1, Byte0. Note PRU1_DMEM_PE_SET[0] maps to Byte0. 0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Enable interrupt 1h (R) = Interrupt enabled
11-8	PRU1_IMEM_PE_SET	R/W	0h	PRU1 IMEM Parity Error Set Enable for Byte3, Byte2, Byte1, Byte0. Note PRU1_IMEM_PE_SET[0] maps to Byte0. 0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Enable interrupt 1h (R) = Interrupt enabled
7-4	PRU0_DMEM_PE_SET	R/W	0h	PRU0 DMEM Parity Error Set Enable for Byte3, Byte2, Byte1, Byte0. Note PRU0_DMEM_PE_SET[0] maps to Byte0. 0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Enable interrupt 1h (R) = Interrupt enabled
3-0	PRU0_IMEM_PE_SET	R/W	0h	PRU0 IMEM Parity Error Set Enable for Byte3, Byte2, Byte1, Byte0. Note PRU0_IMEM_PE_SET[0] maps to Byte0. 0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Enable interrupt 1h (R) = Interrupt enabled

30.5.9.9 PRU_ICSS_CFG_IECP Register (Offset = 20h) [reset = 0h]

PRU_ICSS_CFG_IECP is shown in Figure 30-302 and described in Table 30-299.

Return to Summary Table.

The IRQ Enable Clear Parity Register disables the IRQ PRU ICSS memory parity events.

Figure 30-302. PRU_ICSS_CFG_IECP Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PRU1_DMEM_PE_CLR				PRU1_IMEM_PE_CLR			
R/W-0h				R/W-0h			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRU0_DMEM_PE_CLR				PRU0_IMEM_PE_CLR			
R/W-0h				R/W-0h			

Table 30-299. PRU_ICSS_CFG_IECP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-12	PRU1_DMEM_PE_CLR	R/W	0h	PRU1 DMEM Parity Error Clear Enable for Byte3, Byte2, Byte1, Byte0. Note PRU1_DMEM_PE_CLR[0] maps to Byte0. 0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Disable interrupt 1h (R) = Interrupt enabled
11-8	PRU1_IMEM_PE_CLR	R/W	0h	PRU1 IMEM Parity Error Clear Enable for Byte3, Byte2, Byte1, Byte0. Note PRU1_IMEM_PE_CLR[0] maps to Byte0. 0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Disable interrupt 1h (R) = Interrupt enabled
7-4	PRU0_DMEM_PE_CLR	R/W	0h	PRU0 DMEM Parity Error Clear Enable for Byte3, Byte2, Byte1, Byte0. Note PRU0_DMEM_PE_CLR[0] maps to Byte0. 0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Disable interrupt 1h (R) = Interrupt enabled
3-0	PRU0_IMEM_PE_CLR	R/W	0h	PRU0 IMEM Parity Error Clear Enable for Byte3, Byte2, Byte1, Byte0. Note PRU0_IMEM_PE_CLR[0] maps to Byte0. 0h (W) = No action 0h (R) = Interrupt disabled (masked) 1h (W) = Disable interrupt 1h (R) = Interrupt enabled

30.5.9.10 PRU_ICSS_CFG_PMAO Register (Offset = 28h) [reset = 0h]

PRU_ICSS_CFG_PMAO is shown in Figure 30-303 and described in Table 30-300.

Return to Summary Table.

The PRU Master OCP Address Offset Register enables for the PRU OCP Master Port Address to have an offset of minus 0x0008_0000. This enables the PRU to access External Host address space starting at 0x0000_0000.

Figure 30-303. PRU_ICSS_CFG_PMAO Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED						PMAO_PRU1	PMAO_PRU0
R/W-0h						R/W-0h	R/W-0h

Table 30-300. PRU_ICSS_CFG_PMAO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	PMAO_PRU1	R/W	0h	PRU1 OCP Master Port Address Offset Enable. 0h (R/W) = Disable address offset. 1h (R/W) = Enable address offset of -0x0008_0000.
0	PMAO_PRU0	R/W	0h	PRU0 OCP Master Port Address Offset Enable. 0h (R/W) = Disable address offset. 1h (R/W) = Enable address offset of -0x0008_0000.

30.5.9.11 PRU_ICSS_CFG_MII_RT Register (Offset = 2Ch) [reset = 1h]

PRU_ICSS_CFG_MII_RT is shown in Figure 30-304 and described in Table 30-301.

Return to Summary Table.

The MII_RT Event Enable Register enables Ethercat (or MII_RT) mode events to the PRU ICSS INTC.

Figure 30-304. PRU_ICSS_CFG_MII_RT Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							MII_RT_EVT_EN
R/W-0h							R/W-1h

Table 30-301. PRU_ICSS_CFG_MII_RT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	MII_RT_EVT_EN	R/W	1h	Enables the MII_RT Events to the INTC. 0h (R/W) = Disabled (use the external events). 1h (R/W) = Enabled (use MII_RT events).

30.5.9.12 PRU_ICSS_CFG_IEPCLK Register (Offset = 30h) [reset = 0h]

PRU_ICSS_CFG_IEPCLK is shown in Figure 30-305 and described in Table 30-302.

Return to Summary Table.

The IEP Clock Source Register defines the source of the IEP clock.

Figure 30-305. PRU_ICSS_CFG_IEPCLK Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED							OCP_EN
R/W-0h							R/W-0h

Table 30-302. PRU_ICSS_CFG_IEPCLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	OCP_EN	R/W	0h	Selects IEP clock source 0h (R/W) = iep_clk is the source 1h (R/W) = ocp_clk is the source

30.5.9.13 PRU_ICSS_CFG_SPP Register (Offset = 34h) [reset = 0h]

PRU_ICSS_CFG_SPP is shown in Figure 30-306 and described in Table 30-303.

Return to Summary Table.

The Scratch Pad Priority and Configuration Register defines the access priority assigned to the PRU cores and configures the scratch pad XFR shift functionality.

Figure 30-306. PRU_ICSS_CFG_SPP Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED						XFR_SHIFT_EN	PRU1_PAD_HP_EN
R/W-0h						R/W-0h	R/W-0h

Table 30-303. PRU_ICSS_CFG_SPP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	XFR_SHIFT_EN	R/W	0h	Enables XIN/XOUT shift functionality. When enabled, R0 [4:0] (internal to PRU) defines the 32-bit offset for XIN and XOUT operations with the scratch pad. 0h (R/W) = Disabled 1h (R/W) = Enabled
0	PRU1_PAD_HP_EN	R/W	0h	Defines which PRU wins write cycle arbitration to a common scratch pad bank. The PRU which has higher priority will always perform the write cycle with no wait states. The lower PRU will get stalled/wait states until higher PRU is not performing write cycles. If the lower priority PRU writes to the same byte has the higher priority PRU, then the lower priority PRU will over write the bytes. 0h (R/W) = PRU0 has highest priority 1h (R/W) = PRU1 has highest priority

30.5.9.14 PRU_ICSS_CFG_PIN_MX Register (Offset = 40h) [reset = 0h]

PRU_ICSS_CFG_PIN_MX is shown in Figure 30-307 and described in Table 30-304.

Return to Summary Table.

The Pin Mux Select Register defines the state of the PRU ICSS internal pinmuxing.

Figure 30-307. PRU_ICSS_CFG_PIN_MX Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="6">RESERVED</td><td>PWM3_REMAP_EN</td><td>PWM0_REMAP_EN</td></tr>
<tr><td colspan="6">R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
</table>

Table 30-304. PRU_ICSS_CFG_PIN_MX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	0h	
9	PWM3_REMAP_EN	R/W	0h	If enabled, PRU-ICSS0 Host Interrupt 6 controls pwm0_sync_in. See Chapter 20 (PWMSS) for more details about pwm0_sync_in. Note this bitfield is only available for PRU-ICSS0. In PRU-ICSS1, it is reserved. 0 = Disabled 1 = Enabled
8	PWM0_REMAP_EN	R/W	0h	If enabled, PRU-ICSS0 Host Interrupt 7 controls pwm3_sync_in. See Chapter 20 (PWMSS) for more details about pwm3_sync_in. Note this bitfield is only available for PRU-ICSS0. In PRU-ICSS1, it is reserved. 0 = Disabled 1 = Enabled
7-0	RESERVED	R/W	0h	

30.5.9.15 PRU_ICSS_CFG_SD_P0_CLK_i Register (Offset = 48h + formula) [reset = 0h]

PRU_ICSS_CFG_SD_P0_CLK_i is shown in Figure 30-308 and described in Table 30-305.

Return to Summary Table.

PRU0 SD0-8 CLK/ACC Select

Offset = 48h + (i * 8h); where i = 0h to 8h

Figure 30-308. PRU_ICSS_CFG_SD_P0_CLK_i Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R/W-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
R/W-0h							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED			ACC1_SEL	RESERVED	CLKINV	CLKSEL	
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	

Table 30-305. PRU_ICSS_CFG_SD_P0_CLK_i Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4	ACC1_SEL	R/W	0h	acc3/acc2 mux 0h (R/W) = acc3 is selected 1h (R/W) = acc2 is selected
3	RESERVED	R/W	0h	
2	CLKINV	R/W	0h	Optional clock inversion post clock selection mux 0h (R/W) = No inversion 1h (R/W) = Inversion
1-0	CLKSEL	R/W	0h	Selects the clock source 0h (R/W) = pru0_gpi[16] 1h (R/W) = pru0_sd<m>_clk 2h (R/W) = pru0_sd0_clk for sd0,sd1, and sd2; pru0_sd3_clk for sd3,sd4, and sd5; pru0_sd6_clk for sd6,sd7, and sd8 3h (R/W) = RESERVED

30.5.9.16 PRU_ICSS_CFG_SD_P0_SS_i Register (Offset = 4Ch + formula) [reset = 7h]

PRU_ICSS_CFG_SD_P0_SS_i is shown in Figure 30-309 and described in Table 30-306.

Return to Summary Table.

Offset = 4Ch + (i * 8h); where i = 0h to 8h

Figure 30-309. PRU_ICSS_CFG_SD_P0_SS_i Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">SAMPLE_SIZE</td></tr>
<tr><td colspan="8">R/W-0h</td><td colspan="8">R/W-7h</td></tr>
</table>

Table 30-306. PRU_ICSS_CFG_SD_P0_SS_i Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	SAMPLE_SIZE	R/W	7h	SD Sample Size. This field defines how many samples to take before giving output. The effect count is (P0_SD_SS + 1). 0 = Reserved 1 = Reserved 2 = Reserved 3 = Over Sample of 4 4 = Over Sample of 5 5 = Over Sample of 6 6 = Over Sample of 7 7 = Over Sample of 8 Note this value is only loaded into a shadow copy when channel_en is 0 OR re_init is asserted.

30.5.9.17 PRU_ICSS_CFG_SD_P1_CLK_i Register (Offset = 94h + formula) [reset = 0h]

PRU_ICSS_CFG_SD_P1_CLK_i is shown in Figure 30-310 and described in Table 30-307.

Return to Summary Table.

Offset = 94h + (i * 8h); where i = 0h to 8h

Figure 30-310. PRU_ICSS_CFG_SD_P1_CLK_i Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">RESERVED</td><td>ACC1_SEL</td><td>CLKINV</td><td colspan="2">CLKSEL</td></tr>
<tr><td colspan="4">R/W-</td><td>R/W-</td><td>R/W-</td><td colspan="2">R/W-</td></tr>
</table>

Table 30-307. PRU_ICSS_CFG_SD_P1_CLK_i Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W		
3	ACC1_SEL	R/W		acc3/acc2 mux 0h (R/W) = acc3 is selected 1h (R/W) = acc2 is selected
2	CLKINV	R/W		Optional clock inversion post clock selection mux 0h (R/W) = No inversion 1h (R/W) = Inversion
1-0	CLKSEL	R/W		Selects the clock source 0h (R/W) = pru1_gpi[16] 1h (R/W) = pru1_sd<m>_clk 2h (R/W) = pru1_sd0_clk for sd0,sd1, and sd2; pru1_sd3_clk for sd3,sd4, and sd5; pru1_sd6_clk for sd6,sd7, and sd8 3h (R/W) = Reserved

30.5.9.18 PRU_ICSS_CFG_SD_P1_SS_i Register (Offset = 98h + formula) [reset = 0h]

PRU_ICSS_CFG_SD_P1_SS_i is shown in Figure 30-311 and described in Table 30-308.

Return to Summary Table.

Offset = 98h + (i * 8h); where i = 0h to 8h

Figure 30-311. PRU_ICSS_CFG_SD_P1_SS_i Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">RESERVED</td></tr>
<tr><td colspan="16">R/W-</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RESERVED</td><td colspan="8">SAMPLE_SIZE</td></tr>
<tr><td colspan="8">R/W-</td><td colspan="8">R/W-</td></tr>
</table>

Table 30-308. PRU_ICSS_CFG_SD_P1_SS_i Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W		
7-0	SAMPLE_SIZE	R/W		SD Sample Size. This field defines how many samples to take before giving output. The effect count is (P1_SD_SS + 1). 0 = Reserved 1 = Reserved 2 = Reserved 3 = Over Sample of 4 4 = Over Sample of 5 5 = Over Sample of 6 6 = Over Sample of 7 7 = Over Sample of 8 Note this value is only loaded into a shadow copy when channel_en is 0 OR re_init is asserted.

30.5.9.19 PRU_ICSS_CFG_ED_P0_RXCFG Register (Offset = E0h) [reset = 7h]

PRU_ICSS_CFG_ED_P0_RXCFG is shown in Figure 30-312 and described in Table 30-309.

Return to Summary Table.

Figure 30-312. PRU_ICSS_CFG_ED_P0_RXCFG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RX_DIV_FACTOR</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RX_DIV_FACTOR</td></tr>
<tr><td colspan="8">R/W-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>RX_DIV_FACTOR_FRAC</td><td colspan="7">RESERVED</td></tr>
<tr><td>R/W-0h</td><td colspan="7">R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">RESERVED</td><td>RX_CLK_SEL</td><td>RESERVED</td><td colspan="3">RX_SAMPLE_SIZE</td></tr>
<tr><td colspan="3">R/W-0h</td><td>R/W-0h</td><td>R/W-0h</td><td colspan="3">R/W-7h</td></tr>
</table>

Table 30-309. PRU_ICSS_CFG_ED_P0_RXCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RX_DIV_FACTOR	R/W	0h	Division factor for divh16. Effective values is rx_div_factor + 1. 0 = Div 1 1 = Div 2 N = Div (N+1)
15	RX_DIV_FACTOR_FRAC	R/W	0h	Enable Fractional division before the divh16. 0h (R/W) = Div 1 1h (R/W) = Div 1.5
14-5	RESERVED	R/W	0h	
4	RX_CLK_SEL	R/W	0h	Selects the clock source for the divh16fr. 0h (R/W) = uart_clk 1h (R/W) = ocp_clk
3	RESERVED	R/W	0h	
2-0	RX_SAMPLE_SIZE	R/W	7h	Over Sample size. This field defines the number of samples before the shadow copy gets updated and the VAL flag gets set. The effect count is (rx_sample_size + 1). 0 = Reserved 1 = Reserved 2 = Reserved 3 = Over Sample of 4 4 = Over Sample of 5 5 = Over Sample of 6 6 = Over Sample of 7 7 = Over Sample of 8 Note the Over Sample Clock rate divided by the TX Clock rate must equal the Over Sample size.

30.5.9.20 PRU_ICSS_CFG_ED_P0_TXCFG Register (Offset = E4h) [reset = 0h]

PRU_ICSS_CFG_ED_P0_TXCFG is shown in Figure 30-313 and described in Table 30-310.

Return to Summary Table.

Figure 30-313. PRU_ICSS_CFG_ED_P0_TXCFG Register

31	30	29	28	27	26	25	24
TX_DIV_FACTOR							
R/W-							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TX_DIV_FACTOR							
R/W-							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RX_DIV_FACTOR_FRAC	RESERVED				PRUn_ENDAT2_CLK	PRUn_ENDAT1_CLK	PRUn_ENDAT0_CLK
R/W-	R/W-				R-	R-	R-
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TX_BUSY2	TX_BUSY1	TX_BUSY0	TX_CLK_SEL	RESERVED			
R-	R-	R-	R/W-	R/W-			

Table 30-310. PRU_ICSS_CFG_ED_P0_TXCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TX_DIV_FACTOR	R/W		Division factor for divh16. Effective value is TX_DIV_FACTOR + 1.
15	RX_DIV_FACTOR_FRAC	R/W		Enable Fractional division before the divh16. 0h (R/W) = Div 1 1h (R/W) = Div 1.5
14-11	RESERVED	R/W		
10	PRUn_ENDAT2_CLK	R		Observation of pru<n>_endat2_clk pin
9	PRUn_ENDAT1_CLK	R		Observation of pru<n>_endat1_clk pin
8	PRUn_ENDAT0_CLK	R		Observation of pru<n>_endat0_clk pin
7	TX_BUSY2	R		Determines when is allowed to assert tx go for channel 2. 0h = Ready to go 1h = Busy
6	TX_BUSY1	R		Determines when is allowed to assert tx go for channel 1. 0h = Ready to go 1h = Busy
5	TX_BUSY0	R		Determines when is allowed to assert tx go for channel 0. 0h = Ready to go 1h = Busy
4	TX_CLK_SEL	R/W		Selects the clock source for the divh16fr. 0h (R/W) = uart_clk 1h (R/W) = ocp_clk
3-0	RESERVED	R/W		

30.5.9.21 PRU_ICSS_CFG_ED_P0_CFG0_i Register (Offset = E8h + formula) [reset = 0h]

PRU_ICSS_CFG_ED_P0_CFG0_i is shown in Figure 30-314 and described in Table 30-311.

Return to Summary Table.

Offset = E8h + (i * 8h); where i = 0h to 2h

Figure 30-314. PRU_ICSS_CFG_ED_P0_CFG0_i Register

31	30	29	28	27	26	25	24
TX_FIFO_SWAP_BITS	PRUn_EDm_CLK	ED_CLK_OUT_OVERRIDE_EN	PRUn_EDm_RX_SNOOP	RX_FRAME_SIZE			
R/W-	R/W-	R/W-	R-	R/W-			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RX_FRAME_SIZE							
R/W-							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TX_FRAME_SIZE					TX_WIRE_DLY		
R/W-					R/W-		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TX_WIRE_DLY							
R/W-							

Table 30-311. PRU_ICSS_CFG_ED_P0_CFG0_i Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TX_FIFO_SWAP_BITS	R/W		Enables the swapping of the bits when they are loaded into the TX FIFO. NOTE: FIFO MSB always exports bit [7] first. 0h = No swap 1h = Swap [7:0] -> [0:7].
30	PRUn_EDm_CLK	R/W		This controls the state of pru<n>_ endat<m>_clk pin when endat_clk_out_override_en is set.
29	ED_CLK_OUT_OVERRIDE_EN	R/W		When set, enables the software to control pr<k>_pru0_endatj_clk pin. WARNING: Do not override clock during free running mode. This will cause clock duty cycle violation.
28	PRUn_EDm_RX_SNOOP	R		Direct view of pru<n>_ ed<m>_in pin.
27-16	RX_FRAME_SIZE	R/W		RX frame size, after start bit is detected 0 = Special case for TX only phase, ignores start bit, in this case TX CLK_OUT will stop after last TX. 1 = TX CLK_OUT will stop after 1 X Over Sample 8 = TX CLK_OUT will stop after 8 X Over Sample 9 = TX CLK_OUT will stop after 9 X Over Sample . .. 4095 = TX CLK_OUT will stop after 2047 X Over Sample. Note X Over Sample means the number of VAL events. 1 VAL per Over Sample event. When the TX CLK_MODE [1:0] is either 0x0 or 0x1, when this RX_FRAME_SIZE is reached, the tx master CLK_OUT will remain high or low. WARNING: Software must not de-assert RX_EN before RX_FRAME_SIZE expires.

Table 30-311. PRU_ICSS_CFG_ED_P0_CFG0_i Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-11	TX_FRAME_SIZE	R/W		TX frame size 0: disabled, the FIFO will transmit until empty then stop 1: TX FIFO will transmit 1 bits then stop 2: TX FIFO will transmit 2 bits then stop . . 31: TX FIFO will transmit 31 bits then stop Note: At TX completion, pr<k>_pru0_endatj_out_en will deassert.
10-0	TX_WIRE_DLY	R/W		EnDAT TX wire delay using 200-MHz steps (ICLK). Software must program a number divisible by 5. This is used during TX state when CLK_OUT goes from high to low, this transmission can be compensated. Hardware will keep count of clocks and add 5 each time. Note the first rising edge of EnDAT CLK from a TX GO event is tx_wire_dly + tst_delay_counter + 1/2 EnDAT CLK period +/- 15ns . 0h = No delay 5h = 5 ns delay Ah = 10 ns delay Fh = 15 ns delay 7E8h = 10.24 us delay

30.5.9.22 PRU_ICSS_CFG_ED_P0_CFG1_i Register (Offset = ECh + formula) [reset = 0h]

PRU_ICSS_CFG_ED_P0_CFG1_i is shown in Figure 30-315 and described in Table 30-312.

Return to Summary Table.

Offset = ECh + (i * 8h); where i = 0h to 2h

Figure 30-315. PRU_ICSS_CFG_ED_P0_CFG1_i Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RX_EN_COUNTER															
R/W-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TST_DELAY_COUNTER															
R/W-															

Table 30-312. PRU_ICSS_CFG_ED_P0_CFG1_i Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RX_EN_COUNTER	R/W		
15-0	TST_DELAY_COUNTER	R/W		

30.5.9.23 PRU_ICSS_CFG_ED_P1_RXCFG Register (Offset = 100h) [reset = 0h]

PRU_ICSS_CFG_ED_P1_RXCFG is shown in Figure 30-316 and described in Table 30-313.

Return to Summary Table.

Figure 30-316. PRU_ICSS_CFG_ED_P1_RXCFG Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R/W-</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="3">RESERVED</td><td colspan="5">RX_DIV_FACTOR</td></tr>
<tr><td colspan="3">R/W-</td><td colspan="5">R/W-</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RX_DIV_FACTOR</td></tr>
<tr><td colspan="8">R/W-</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">RX_DIV_FACTOR</td><td>RX_DIV_FACTOR_FRAC</td><td>RX_CLK_SEL</td><td colspan="3">RX_SAMPLE_SIZE</td></tr>
<tr><td colspan="3">R/W-</td><td>R/W-</td><td>R/W-</td><td colspan="3">R/W-</td></tr>
</table>

Table 30-313. PRU_ICSS_CFG_ED_P1_RXCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R/W		
20-5	RX_DIV_FACTOR	R/W		Division factor for divh16. Effective values is rx_div_factor + 1. 0 = Div 1 1 = Div 2 N = Div (N+1)
4	RX_DIV_FACTOR_FRAC	R/W		Enable Fractional division before the divh16. 0h (R/W) = Div 1 1h (R/W) = Div 1.5
3	RX_CLK_SEL	R/W		Selects the clock source for the divh16fr. 0h (R/W) = uart_clk 1h (R/W) = ocp_clk
2-0	RX_SAMPLE_SIZE	R/W		Over Sample size. This field defines the number of samples before the shadow copy gets updated and the VAL flag gets set. The effect count is (rx_sample_size + 1). 0 = Reserved 1 = Reserved 2 = Reserved 3 = Over Sample of 4 4 = Over Sample of 5 5 = Over Sample of 6 6 = Over Sample of 7 7 = Over Sample of 8 Note the Over Sample Clock rate divided by the TX Clock rate must equal the Over Sample size.

30.5.9.24 PRU_ICSS_CFG_ED_P1_TXCFG Register (Offset = 104h) [reset = 0h]

PRU_ICSS_CFG_ED_P1_TXCFG is shown in Figure 30-317 and described in Table 30-314.

Return to Summary Table.

Figure 30-317. PRU_ICSS_CFG_ED_P1_TXCFG Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TX_DIV_FACTOR							
R/W-							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TX_DIV_FACTOR							
R/W-							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RX_DIV_FACTOR_FRAC	PRUn_ENDAT2_CLK	PRUn_ENDAT1_CLK	PRUn_ENDAT0_CLK	TX_BUSY2	TX_BUSY1	TX_BUSY0	TX_CLK_SEL
R/W-	R-	R-	R-	R-	R-	R-	R/W-

Table 30-314. PRU_ICSS_CFG_ED_P1_TXCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W		
23-8	TX_DIV_FACTOR	R/W		Division factor for divh16. Effective value is TX_DIV_FACTOR + 1.
7	RX_DIV_FACTOR_FRAC	R/W		Enable Fractional division before the divh16. 0h (R/W) = Div 1 1h (R/W) = Div 1.5
6	PRUn_ENDAT2_CLK	R		Observation of pru<n>_endat2_clk pin
5	PRUn_ENDAT1_CLK	R		Observation of pru<n>_endat1_clk pin
4	PRUn_ENDAT0_CLK	R		Observation of pru<n>_endat0_clk pin
3	TX_BUSY2	R		Determines when is allowed to assert tx go for channel 2. 0h = Ready to go 1h = Busy
2	TX_BUSY1	R		Determines when is allowed to assert tx go for channel 1. 0h = Ready to go 1h = Busy
1	TX_BUSY0	R		Determines when is allowed to assert tx go for channel 0. 0h = Ready to go 1h = Busy
0	TX_CLK_SEL	R/W		Selects the clock source for the divh16fr. 0h (R/W) = uart_clk 1h (R/W) = ocp_clk

30.5.9.25 PRU_ICSS_CFG_ED_P1_CFG0_i Register (Offset = 108h + formula) [reset = 0h]

PRU_ICSS_CFG_ED_P1_CFG0_i is shown in Figure 30-318 and described in Table 30-315.

Return to Summary Table.

Offset = 108h + (i * 8h); where i = 0h to 2h

Figure 30-318. PRU_ICSS_CFG_ED_P1_CFG0_i Register

31	30	29	28	27	26	25	24
TX_FIFO_SWAP_BITS	PRUn_EDm_CLK	ED_CLK_OUT_OVERRIDE_EN	PRUn_EDm_RX_SNOOP	RX_FRAME_SIZE			
R/W-	R/W-	R/W-	R-	R/W-			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RX_FRAME_SIZE							
R/W-							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TX_FRAME_SIZE					TX_WIRE_DLY		
R/W-					R/W-		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TX_WIRE_DLY							
R/W-							

Table 30-315. PRU_ICSS_CFG_ED_P1_CFG0_i Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TX_FIFO_SWAP_BITS	R/W		Enables the swapping of the bits when they are loaded into the TX FIFO. NOTE: FIFO MSB always exports bit [7] first. 0h = No swap 1h = Swap [7:0] -> [0:7].
30	PRUn_EDm_CLK	R/W		This controls the state of pru<n>_ endat<m>_clk pin when endat_clk_out_override_en is set.
29	ED_CLK_OUT_OVERRIDE_EN	R/W		When set, enables the software to control pr<k>_pru0_endatj_clk pin. WARNING: Do not override clock during free running mode. This will cause clock duty cycle violation.
28	PRUn_EDm_RX_SNOOP	R		Direct view of pru<n>_ ed<m>_in pin.
27-16	RX_FRAME_SIZE	R/W		RX frame size, after start bit is detected 0 = Special case for TX only phase, ignores start bit, in this case TX CLK_OUT will stop after last TX. 1 = TX CLK_OUT will stop after 1 X Over Sample 8 = TX CLK_OUT will stop after 8 X Over Sample 9 = TX CLK_OUT will stop after 9 X Over Sample . .. 4095 = TX CLK_OUT will stop after 2047 X Over Sample. Note X Over Sample means the number of VAL events. 1 VAL per Over Sample event. When the TX CLK_MODE [1:0] is either 0x0 or 0x1, when this RX_FRAME_SIZE is reached, the tx master CLK_OUT will remain high or low. WARNING: Software must not de-assert RX_EN before RX_FRAME_SIZE expires.

Table 30-315. PRU_ICSS_CFG_ED_P1_CFG0_i Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-11	TX_FRAME_SIZE	R/W		TX frame size 0: disabled, the FIFO will transmit until empty then stop 1: TX FIFO will transmit 1 bits then stop 2: TX FIFO will transmit 2 bits then stop . . 31: TX FIFO will transmit 31 bits then stop Note: At TX completion, pr<k>_pru0_endatj_out_en will deassert.
10-0	TX_WIRE_DLY	R/W		EnDAT TX wire delay using 200-MHz steps (ICLK). Software must program a number divisible by 5. This is used during TX state when CLK_OUT goes from high to low, this transmission can be compensated. Hardware will keep count of clocks and add 5 each time. Note the first rising edge of EnDAT CLK from a TX GO event is tx_wire_dly + tst_delay_counter + 1/2 EnDAT CLK period +/- 15ns . 0h = No delay 5h = 5 ns delay Ah = 10 ns delay Fh = 15 ns delay 7E8h = 10.24 us delay

30.5.9.26 PRU_ICSS_CFG_ED_P1_CFG1_i Register (Offset = 10Ch + formula) [reset = 0h]

PRU_ICSS_CFG_ED_P1_CFG1_i is shown in Figure 30-319 and described in Table 30-316.

Return to Summary Table.

Offset = 10Ch + (i * 8h); where i = 0h to 2h

Figure 30-319. PRU_ICSS_CFG_ED_P1_CFG1_i Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RX_EN_COUNTER															
R/W-															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TST_DELAY_COUNTER															
R/W-															

Table 30-316. PRU_ICSS_CFG_ED_P1_CFG1_i Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RX_EN_COUNTER	R/W		
15-0	TST_DELAY_COUNTER	R/W		

Debug Subsystem

This chapter describes the debug subsystem of the device.

Topic **Page**

31.1 Chip Architecture Specification

31.1.1 Debug Resource Manager (DRM)

31.1.1.1 Debug Suspend Support for Peripherals

When a processor is halted, peripherals associated with the processor must appropriately respond to this event to avoid incorrect actions.

An example of this is the action of the Watchdog Timer (WDT) during a debug halt. Typically watchdog timers fire a reset to restart a system after a timeout. The reset could be misfired during debug if a processor is halted for a fairly long time and prevents a WDT monitor from refreshing the timer.

To prevent this incorrect action, the watchdog timer supports a debug suspend event. This event allows the WDT to stop counting during a CPU halt.

Other peripherals also support a debug suspend event. The list of supported peripherals is shown in Section 31.5.

Note that several peripherals have local control to gate the suspend event coming from the Debug Subsystem. For example, the WDT has an EMUFREE bit in the WDSC register to block the suspend event coming from the Debug Subsystem. Ensure this bit is set correctly to allow the suspend event to properly control the peripheral module.

Recommended Suspend Control Register Value:
  Normal mode: 0x0
  Suspend peripheral during debug halt: 0x9

31.1.2 On-Chip Debug and Trace

31.1.2.1 Introduction

Debugging a system containing an embedded processor involves an environment that connects high-level debugging software, executing on a host computer, to a low-level debug interface supported by the target device. In between these levels is a debug and trace controller (DTC) that facilitates communication between the host debugger and the debug support logic on the target chip.

A combination of hardware and software that connects the host debugger to the target system, the DTC uses one or more hardware interfaces and protocols to convert actions dictated by the debugger user to JTAG commands and scans that execute the core hardware. The debug software and hardware components let the user control multiple central processing unit (CPU) cores embedded in the device in a global or local manner. This environment provides:

- Synchronized global starting and stopping of multiple processors
- Starting and stopping of an individual processor
- Each processor can generate triggers than can be used to alter the execution flow of other processors.

System topics include but are not limited to:

- System clocking and power-down issues
- Interconnection of multiple devices
- Trigger channels

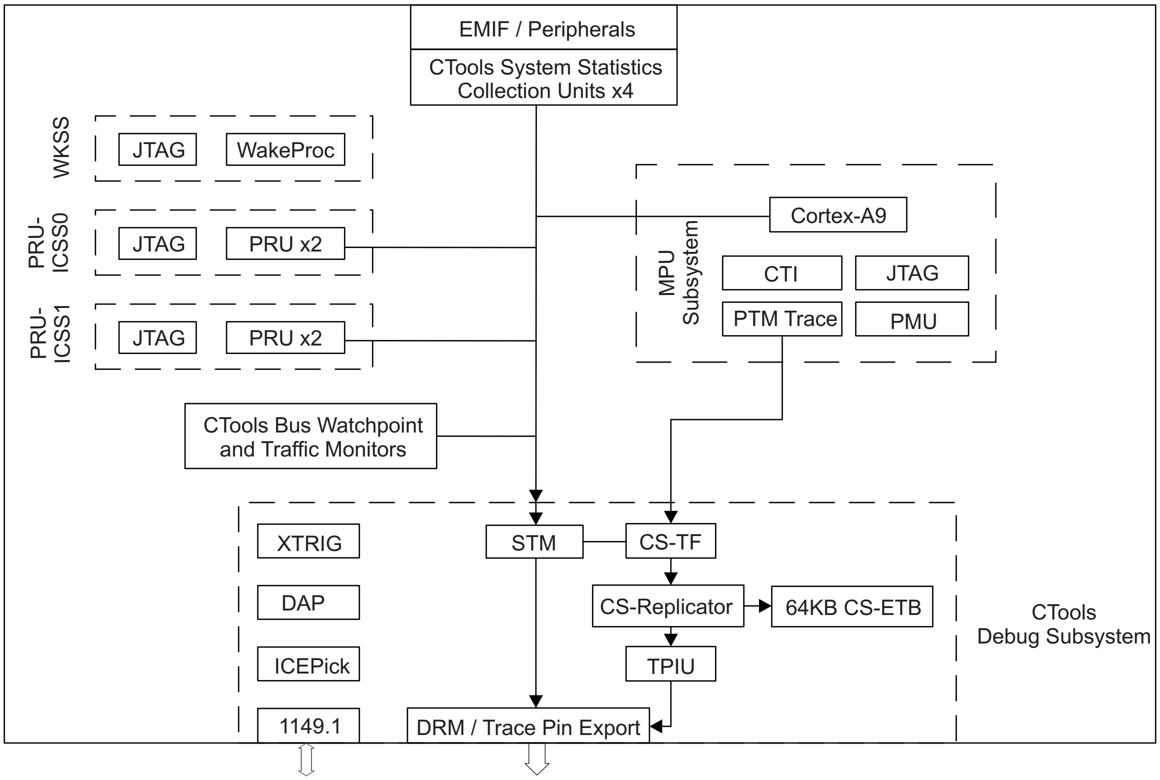
For easy integration into applications, a set of libraries (APIs) for debug-IP programming and a software message library are provided. More information about the APIs, download files, and other useful links for available libraries can be found on the CToolsLib Wiki site:

http://processors.wiki.ti.com/index.php/CToolsLib

CToolsLib is a collection of embedded target APIs and libraries that enable easy access to the chip tools (CTools), which are system-level debug facilities included in the debug subsystem of TI devices.

Figure 31-1 shows a high-level debug view of the functional block diagram.

Figure 31-1. Functional Block Diagram Debug View



31.1.2.1.1 Key Features

This device deploys the Texas Instruments CTools debug technology for on-chip debug and trace support.

CTools includes the following features:

- JTAG based single processor debugging on:
  - Cortex-A9 in MPU Subsystem
  - Wakeup Processor in Wakeup Subsystem
  - PRU x2 in PRU-ICSS1
  - PRU x2 in PRU-ICSS0
- Multiprocessor debugging lets users control multiple CPU cores embedded in the device, such as:
  - Global starting and stopping of individual or multiple processors
  - Each processor can generate triggers to alter the execution flow of other processors
  - System clocking and power down
  - Interconnection of multiple devices
  - Channel triggering
- Target debugging using the JTAG (IEEE1149.1) port
- Reduced power consumption in normal operating mode
- Real-time software trace allows software masters to transmit trace data from OS processes or tasks on 256 different channels

The debug subsystem includes:

- CTools Power Aware JTAG Router (ICEPick-D™)
- Debug access port (DAP)
- Processor trace subsystem
  - 64KB CoreSight Embedded Trace Buffer (CS-ETB)
- System trace subsystem
- EMU configuration interconnect
- Cross-triggering unit (XTRIGGER)
- Debug resource manager (DRM)
- CoreSight Trace Port Interface Unit (CS-TPIU)
- CORE instrumentation interconnect:
  - Initiator ports:
    - L3 interconnect for software instrumentation and performance probes
    - CTools System Bus Watchpoint and Traffic Monitors (OCP-WP)
  - Target port:
    - EMU instrumentation interconnect

CTools System Bus Watchpoint and Traffic Monitors (OCP-WP):

- Monitors L3 interconnect transaction when target transaction attributes match the user-defined attributes or trigger on external debug event
- Only one instance, shared among the following L3 targets:
  - GPMC
  - L4_FAST
  - L4_SLOW
  - L4_WAKEUP
  - EMIF
  - OCM_RAM

CTools System Statistics Collection Units (Statistics Collector): The L3 interconnect supports a built-in performance monitoring feature by implementing a statistics collector component, which computes traffic statistics within a user-defined window and periodically reports to the user through the MIPI-STM interface.

31.1.2.2 Debug Ports

31.1.2.2.1 IEEE 1149.1

The target debug interface has the following signals:

- Five standard IEEE1149.1 JTAG signals: nTRST, TCK, TMS, TDI, and TDO
- Two EMU [1:0] or five EMU [4:0] TI extensions, depending on the pin count (14 pins or 20 pins) in the JTAG header of the device.

Table 31-1 describes the IEEE1149.1 signals.

Table 31-1. IEEE 1149.1 Signals

Pin	Type	Name	Description
nTRST	I	Test Logic Reset	When asserted (active low) causes all test and debug logic in the device to be reset along with the IEEE 1149.1 interface
TCK	I	Test Clock	This is the test clock used to drive an IEEE 1149.1 TAP state machine and logic. Depending on the DTS attached to the device, this is a free running clock or a gated clock depending on RTCK monitoring.

Table 31-1. IEEE 1149.1 Signals (continued)

Pin	Type	Name	Description		
RTCK	O	Returned Test Clock	Synchronized TCK. Depending on the DTS attached to the device, the JTAG signals are clocked from RTCK or RTCK is monitored by the DTS to gate TCK.		
TMS	I	Test Mode Select	Directs the next state of the IEEE 1149.1 test access port state machine		
TDI	I	Test Data Input	Scan data input to the device		
TDO	O	Test Data Output	Scan data output of the device		
EMU0	I/O	Emulation 0	Channel 0 trigger	ICEPick Boot Mode	Trace port
EMU1	I/O	Emulation 1	Channel 1 trigger		
EMU2	O	Emulation 2	20-pins JTAG header only		
EMU3	O	Emulation 3			
EMU4	O	Emulation 4			

Table 31-2 describes the device's JTAG ID code, which is accessed through the ICEPick module embedded in the debug subsystem.

Table 31-2. JTAG ID Code

Device	JTAG ID
AM43xx	0000-1011-1001-1000-1100-0000-0010-1111
	0x0B98C02F

31.1.2.2.2 Trace Port

On-chip debug and trace events can be exported to external equipment through the trace port in the device. The following exportable debug events and trace sources are supported:

- Debug events:
  - Triggers. For more information about triggers, see Section 12.4.2, *Cross-Triggering*.
- Trace sources
  - Processor trace: Cortex-A9 MPU trace supported by CS_PTM module.
  - System trace: Trace coming from various system Instrumentation modules and supported by system trace module (STM). For more information about system trace, see Section 31.2, *System Instrumentation*.

Note that not all debug and trace features can be used simultaneously due to limited amount of pins allocated to debug. Thus, a multiplexing among debug and trace sources is implemented. The configuration and the selection of debug and trace sources is done through the debug resource manager (DRM) module embedded in the debug subsystem.

Table 31-3 describes the trace port signals.

Table 31-3. Trace Port Signals

Pin Name	Internal Signal Name	I/O	Description
dpm_emu11	EMU11	O	Debug Resource manager pin 11
dpm_emu10	EMU10	O	Debug Resource manager pin 10
dpm_emu9	EMU9	O	Debug Resource manager pin 9
dpm_emu8	EMU8	O	Debug Resource manager pin 8
dpm_emu7	EMU7	O	Debug Resource manager pin 7

Table 31-3. Trace Port Signals (continued)

Pin Name	Internal Signal Name	I/O	Description
dpm_emu6	EMU6	O	Debug Resource manager pin 6
dpm_emu5	EMU5	O	Debug Resource manager pin 5
dpm_emu4	EMU4	O	Debug Resource manager pin 4
dpm_emu3	EMU3	O	Debug Resource manager pin 3
dpm_emu2	EMU2	O	Debug Resource manager pin 2
dpm_emu1	EMU1	I/O	Debug Resource manager pin 1
dpm_emu0	EMU0	I/O	Debug Resource manager pin 0

NOTE: The dpm_emu[11:2] pins are shared with other functional (application) pins on the device boundary. To use the dpm_emu[4:2] pins, user must program the device application pin manager (control module) appropriately. For more information, see Chapter 7, *Control Module*.

31.1.2.2.3 Target Connection

TI supports a variety of eXtended Development System (XDS) JTAG controllers with various debug capabilities beyond only JTAG support. If your device supports the export of core trace or system trace over the EMU pins, and you want the target to be compatible with XDS products that are capable of acquiring either trace type, see the *Emulation and Trace Headers Technical Reference Manual* (SPRU655). You can also find more information at the “XDS Target Connection Guide” wiki page:

http://processors.wiki.ti.com/index.php/XDS_Target_Connection_Guide

31.1.3 Debugger Connection

31.1.3.1 ICEPick Module

The debugger connects to the device through its JTAG interface. The first level of debug interface seen by the debugger is connected to the ICEPick (version D) module that is embedded in the debug system.

System-on-chip (SoC) designs typically have multiple processors, each having a JTAG test access port (TAP) embedded in the processor. The ICEPick module manages these TAPs and the power, reset, and clock controls for modules that have TAPs.

ICEPick provides the following debug capabilities:

- Debug connect logic for enabling or disabling most ICEPick instructions
- Dynamic TAP insertion
  - Serially link up to 32 TAP controllers
  - Individually select one or more of the TAPS for scan without disrupting the instruction register (IR) state of other TAPs
- Power, reset and clock management
  - Provides the power and clock states of each domain
  - Provides debugger control of the processor power domain. Can force the domain power and clocks on, and prohibit the domain from being clock-gated or powered down while a debugger is connected.
  - Applies system reset

  - Provides wait-in-reset (WIR) boot mode
  - Provides global and local WIR release
  - Provides global and local reset block

The ICEPick module implements a connect register, which must be configured with a predefined key to enable the full set of JTAG instructions. Once the debug connect key is properly programmed, the emulation logics for the ICEPick signals and subsystems emulation logics should be turned on.

For more information about ICEPick dynamic TAP insertion, see Section 31.1.3.3, Dynamic TAP insertion.

For more information about ICEPick power, reset, and clock management features, see Section 31.1.6, *Power, Reset, and Clock Management Debug Support*.

31.1.3.2 Boot Modes

The initial configuration of ICEPick is determined by the level of the dpm_emu0 and dpm_emu1 pins upon POR release. At POR, dpm_emu0 and dpm_emu1 are automatically configured as inputs. The dpm_emu0 and dpm_emu1 pins are free once POR has been released.

Table 31-4 summarizes the ICEPick boot modes.

Table 31-4. ICEPick Boot Modes upon POR

dpm_emu1	dpm_emu0	TAPs in the TDI → TDO Path	Other Effects/Comments
0	0	None	Reserved (do not use)
0	1	None	Reserved (do not use)
1	0	ICEPick	TAP only + wait-in-reset mode
1	1	ICEPick	TAP only (default mode)

31.1.3.2.1 Default Boot Mode

In ICEPick-only configuration, none of the secondary TAPs are selected. The ICEPick TAP is the only TAP between device-level TDI and TDO pins. This is the recommended boot mode.

31.1.3.2.2 Wait-In-Reset (WIR)

The device can boot to invoke WIR mode. If the device is booted in WIR mode, all processors within the device that support a TAP through ICEPick are held in reset until released. Individual processors may be released from reset (local), or all processors held in the reset state may be released at the same time (global).

Note: The PRU processors in PRU-ICSS do not support WIR mode.

31.1.3.3 Dynamic TAP insertion

31.1.3.3.1 ICEPick Secondary TAPS

To include additional or fewer secondary TAPS in the scan chain, the debugger **must** use the ICEPick TAP router to program the TAPs. At its root, ICEPick is a scan-path linker that lets the DTC selectively choose which subsystem TAPs are accessible through the device-level debug interface. Each secondary TAP can be dynamically included in or excluded from the scan path. To the external JTAG interface, the secondary TAPs that are not selected appear not to exist.

Table 31-5 shows the secondary debug TAPs connect to the ICEPick scan chain along with the modules that can be accessed. The TAP number shows the position of the TAP in the scan chain.

Table 31-5. ICEPick Secondary Debug TAP Mapping

Secondary JTAG port	CoreSight	TAP	Modules Accessed through that JTAG Port	
Debug bank				
Reserved	No	0	-	
Reserved	No	1	-	
Reserved	No	2	-	
Reserved	No	3	-	
Reserved	No	4	-	
Reserved	No	5	-	
Reserved	No	6	-	
Reserved	No	7	-	
Reserved	No	8	-	
Reserved	No	9	-	
	No	10	-	
Wakeup Subsystem	No	11	Wakeup Processor	
CS-DAP (APB-AP)	Yes	12	Cortex-A9	MPU Subsystem
	Yes		PTM	
	No		PMU	
	Yes		CS-ETB	Debug Subsystem
	Yes		CS-TF	
	No		DRM	
	No		MIPI-STM	
AHB-AP	No		PRU-ICSS1	PRU x2
			PRU-ICSS0	PRU x2
	No		NOC	NOC Statistics Collectors (4)
	No		OCP-WP	L3 OCP Watch-point
Test bank				
Reserved	No	0	Reserved	
Reserved	No	1	Reserved	
Reserved	No	2	Reserved	
Reserved	No	3	Reserved	
Reserved	No	4	Reserved	

Besides secondary debug TAPs, ICEPick also supports power, reset, and clock controls for non-JTAG debug cores. The debug cores are accessible through DAP.

Table 31-6 summarizes the ICEPick debug core mapping

Table 31-6. ICEPick Debug Core Mapping

Debug Core #	Module
0	Cortex-A9 MPU Subsystem
1	PRU0 in PRU-ICSS1
2	PRU1 in PRU-ICSS1
3	PRU0 in PRU-ICSS0
4	PRU1 in PRU-ICSS0
5-8	Reserved

31.1.4 Primary Debug Support

31.1.4.1 Processor Native Debug Support

31.1.4.1.1 Cortex-A9 MPU Subsystem

The Cortex-A9 processor supports the following native features:

- Halt mode and monitor mode debug
- Hardware breakpoints and watch points
- Performance monitoring

For more information about Cortex-A9 subsystem native debug support features, see the ARM Cortex-A9 Technical Reference Manual.

The PTM can send trace data to the ETB. The DAP handles all scan communications. The ARM Cortex-A9 core, the TI debug extender, the PTM, and the ETB are all accessible via memory-map transactions controlled by the DAP.

For details about the core, the PTM, and the ETB, see the ARM Limited documentation.

31.1.4.1.2 Wakeup Processor MPU

The wakeup processor supports the following native debug features:

- Program halt and stepping
- Hardware breakpoints, breakpoint instruction
- Data watch point on access to data add, add range, and data value
- Register value accesses
- Debug monitor exception
- Memory accesses

The Wake-up Subsystem includes one Cortex-M3 processor.

For more information about Cortex-M3 MPU native debug support features, see the ARM Cortex-M3 Technical Reference Manual.

31.1.4.1.3 PRU

The PRU processors in the PRU-ICSS1 and PRU-ICSS0 support the following native debug features:

- Manual halt
- Single-step execution
- Software Breakpoint

31.1.4.2 Cross Triggering

The device supports a cross-triggering feature, which propagates debug (trigger) events from one processor subsystem or module to another. For example, a given subsystem, A, can be programmed to generate a debug event, which can then be exported as a global trigger across the device. Another subsystem, B, can be programmed to be sensitive to the trigger line input and to generate an action upon trigger detection.

Examples of debug events include:

- Processor entering debug state
- Watch point match
- PTM trigger
- ETB full
- ETB acquisition complete

Examples of debug actions include:

- Debug request generation
- Restart (Cortex-A9 synchronous run)
- Interrupt request generation
- Start and stop trace

Subsystems cross-triggering is consolidated at the device level by the XTRIGGER module, which is embedded in the debug subsystem.

NOTE: XTRIGGER is not programmatically visible from the JTAG interface or any device processor. Cross-triggering is programmed at the subsystem level.

31.1.4.3 SOC Level Cross-Triggering

Table 31-7 summarizes the device cross-triggering capabilities.

Table 31-7. Cross-Triggering

	Trigger Source		MPU Cores	PTM/ ETB	PRU-ICSS1 PRU0	PRU-ICSS0 PRU1	PRU-ICSS0 PRU0	PRU-ICSS0 PRU1	Wakeup Processor	NOC_ SC	OCP_ WP	EMU0, EMU1
		Trigger Input	●	●	●	●	●	●	●	●	●	●
MPU	●	MPU Cores		●					●	●	●	●
	●	PTM / ETB	●							●	●	●
PRU-ICSS1	●	PRU0										
	●	PRU1										
PRU-ICSS0	●	PRU0										
	●	PRU1										
Wakeup	●	Wakeup Processor	●							●	●	●
SOC	-	NOC_ SC										
	●	OCP_ WP	●	●					●	●		●
	●	EMU0, EMU1	●	●					●	●	●	

The cross-trigger lines are shared by all the subsystems implementing cross-triggering. An MPU subsystem trigger event can, therefore, be propagated to any application subsystem or system trace component. The remote subsystem or system trace component can be programmed to be sensitive to the global SOC trigger lines for the following actions:

- Generate a processor debug request
- Generate an interrupt request
- Start processor trace
- Stop processor trace
- Start OCP target traffic monitoring
- Stop OCP target traffic monitoring
- Start NOC performance monitoring
- Stop NOC performance monitoring
- Start external logic analyzer trace
- Stop external logic analyzer trace

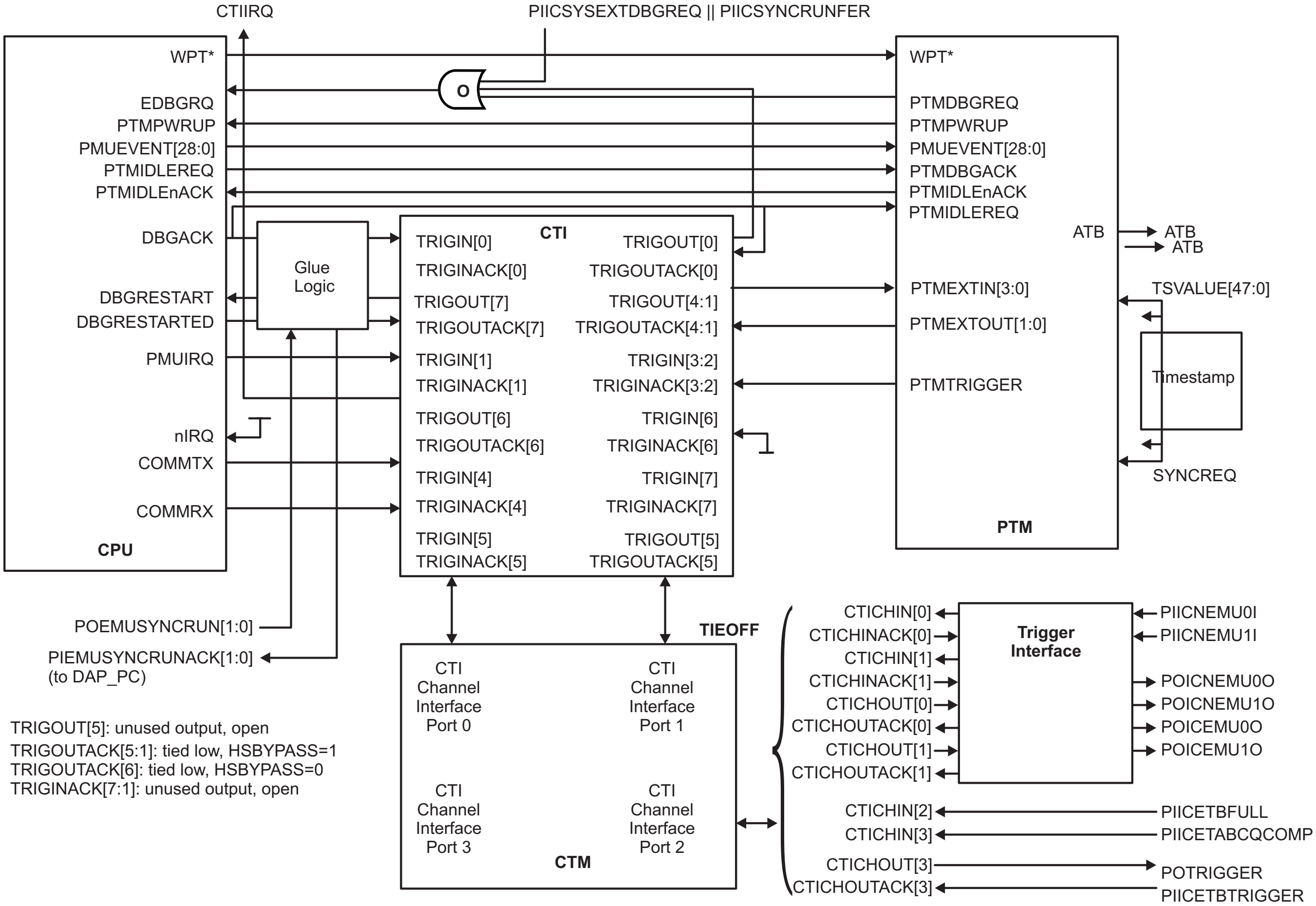
31.1.4.4 MPU Subsystem Cross-Triggering

The MPU subsystem (MPUSS) implements standard CoreSight components to perform SMP debug and emulation. There is one CTI and PTM module to handle cross-trigger and trace. The CTI is connected to a CTM (Cross Trigger Matrix) to enable cross trigger across CPUs. Since the MPUSS supports only one CPU, the other input of the CTM is tied-off. A trace funnel selects traces from the CPU and sends them to a single ATB interface. A time-stamp counter is included to enable correlation of trace stream from multiple processors.

Figure 31-2 shows the cross trigger connections of the MPUSS. The MPUSS does not include the ICECrusherCS. A trigger interface block is used to convert TI trigger format (EMU0/EMU1) to ARM CTI format.

The four ports of the CTM are connected to CTI0, CTI1 (tied-off), and the Trigger interface.

Figure 31-2. MPU Subsystem Cross Trigger Connections



TRIGOUT[5]: unused output, open
TRIGOUTACK[5:1]: tied low, HSBYPASS=1
TRIGOUTACK[6]: tied low, HSBYPASS=0
TRIGINACK[7:1]: unused output, open

31.1.5 Suspend

The device supports a suspend feature, which provides a way to stop a "closely-coupled" hardware process running on a peripheral-IP when the host processor enters debug state. The suspend mechanism is important for debug, because it ensures that peripheral-IPs operate in a lock-step manner with a host controller processor.

An entry is provided for each peripheral-IP that must consider the suspend signals from a number of processors (MPU or DSP). For each peripheral-IP, sensitivity to the suspend signals is defined within two possibilities (and so coded using 1bit):

- Peripheral-IP is sensitive to the suspend line request.
- Peripheral-IP ignores the suspend line request.

For more information about how to program the sensitivity, see the peripheral-IP chapter in this document.

31.1.5.1 Debug Aware Peripherals and Host Processors

Table 31-8 lists the mapping of the device peripheral-IPs to the suspend control output lines.

Table 31-8. Debug Subsystem Suspend Output Lines

Suspend Output Line	Peripheral-IP Module
0	System Watchdog Timer1
1	DM Timer0
2	DM Timer1
3	DM Timer2
4	DM Timer3
5	DM Timer4
6	DM Timer5
7	DM Timer6
8	EMAC
9	Reserved
10	I2C0
11	I2C1
12	I2C2
13	eHRPWM0
14	eHRPWM1
15	eHRPWM2
16	DCAN1
17	DCAN2
18	PRU-ICSS
19	Sync_Timer
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	DM Timer7
	DM Timer8
	DM Timer9
	DM Timer10
	DM Timer11
25	Reserved
26	Reserved
27	PWM3
28	PWM4
29	PWM5

NOTE: DM Timer7–11 are grouped together to share the same suspend output. Additional logic is needed at the SoC level to combine the suspend out for all the timers.

31.1.6 Power, Reset and Clock Management Debug Support

The global PRCM module implements facilities to support debug across power and clock domain cycles. The debugger can control or receive the status of each power and clock domain associated with an ICEPick secondary TAP.

ICEPick provides a set of directives that allow the debugger to:

- Access information on the associated power and clock domains state. This includes:
  - Current power setting indicating whether the power domain is on or off
  - Loss of power detected since the software last checked the power status
  - Current clock setting indicating whether the clock domain is on or off
  - Sleep desired (PM and CM indicate that the debug settings in ICEPick are changing the application state. If it were not for the ICEPick controls, the power or clock would be turned off.)
  - Subsystem reset state
  - Subsystem has entered a debug state that requires the attention of the host debug software
- Override power and clock control settings to wake up a power or clock domain or to prevent a power or clock domain from going to sleep once it is in ACTIVE state
- Assert, block, or extend reset and also release from extended reset (WIR)

ICEPick handles debug power management at the device level.

31.1.7 Performance Monitoring

31.1.7.1 Cortex-A9 MPU Subsystem Performance Monitoring

31.1.7.1.1 Performance Monitoring Unit

The Cortex-A9 processor includes a performance monitoring unit (PMU) that enables events, such as cache misses and instructions executed, to be counted over a period of time. The PMU gathers statistics about the operation of the processor and memory system.

For details of PMU events, please refer to the Cortex-A9 TRM.

31.1.8 Processor Trace

The device supports Cortex-A9 processor trace.

31.1.8.1 Cortex-A9 Processor Trace

The Cortex-A9 processor trace characteristics are:

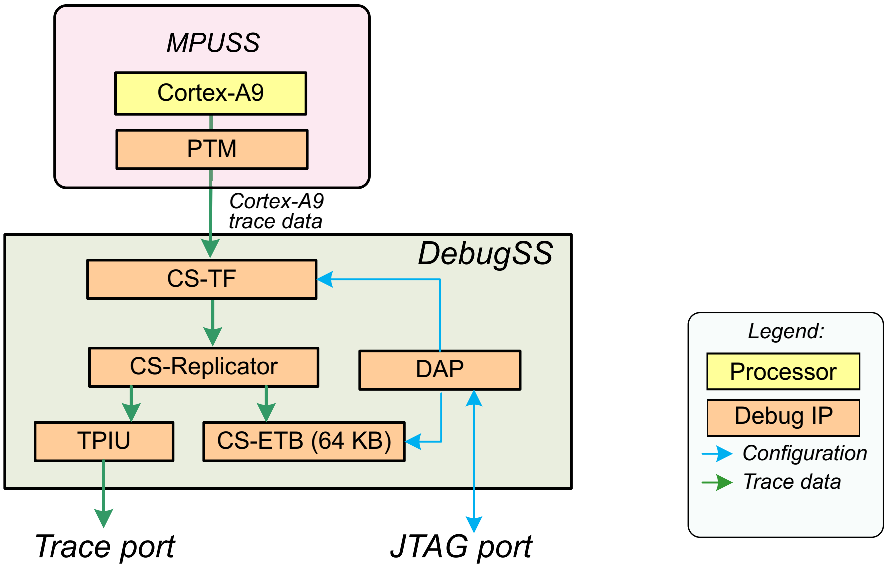
- Program trace
- Data trace (data address only, no data value)
- Trace can be stored on-chip to ETB or exported off chip via TPIU through CS-TF in the Debug Subsystem
- Optional: Trace can be cycle accurate (useful to profile sections of code)

For more details on the Cortex-A9 trace features, see the Cortex-A9 TRM.

31.1.8.2 SoC Processor Trace Flow

Figure 31-3 shows an overview of the SoC level processor trace flow for a Cortex-A9 trace.

Figure 31-3. SoC Processor Trace Flow



31.1.8.3 Trace Exported to an External Trace Receiver

31.1.8.3.1 Debug Resources Manager

Processor trace can be exported to an external trace receiver through the TPIU module. For this purpose, the debugger or application software must program the DRM module properly.

31.1.8.3.2 Trace Port Width Configuration

The TPIU has a software configurable export width of 10 or 12 data pins (TRACEDATA) plus a dedicated export clock (TRACECLK) and a control signal (TRACECTL).

31.1.8.3.3 Tuning Export Clock Frequency

The device implements two processor trace export clock (TRACECLK) generation schemes:

- Trace export clock generated from the Debug Subsystem Programmable Delay Line Oscillator (PDLO)
- Programmable DPLL reference clock shared with application with programmable post-dividers

---

NOTE:

1. Selection between PDLO and PMD schemes is done at the DRM level.
2. The debug subsystem also implements a dedicated PDLO module for STM.

31.1.8.3.4 Tuning Export Clock Frequency Through PDLO

The Debug Subsystem implements a dedicated PDLO module for Processor Trace export clock generation. The PDLO provides a wide operating frequency range to compensate for process, voltage and temperature variations. The PDLO also contains a post clock divider to further extend the operating frequency range.

The PDLO contains two identical delay lines with 32 x TAPs each. The delays lines are operated in ring oscillator mode. The PDLO output frequency will be programmed by the debug software by measuring the delay line output frequency, adjusting the delay TAP and post divider until the desired frequency is reached. After the delay lines are set to the correct operating frequency, the high and low limit registers are programmed with the maximum and minimum tolerances. The PDLO periodically monitors the output frequency, and if it detects the frequency is drifting out of tolerance, it initiates a calibration cycle (typically 3 to 79 times per second).

The calibration cycle will select the alternate delay line and measure its operating frequency. The delay tap select value is incremented or decremented, adjusting the frequency until it is within tolerance. Once the alternate delay line frequency is in tolerance, the controller will synchronously switch between the two delay lines. When the switch occurs, the export clock will stop for two clock cycles. During these two cycles, the processor trace is not exported.

The system clock input frequency is used as the reference for all measurements. The PDLO allows for a range of system clock frequencies.

31.1.8.3.5 Tuning Export Clock Frequency Through DPLL

The processor trace export clock can be derived from the CORE DPLL. Selection of the DPLL reference clock is done through the Clock Manager (CM) module. The CM module also provides a means of tuning the frequency of the processor trace export clock by dividing it to 1, 2, or 4 at the PRCM level. It is then possible to slow down the interface clock at the debug subsystem level.

For more information, see the EMU Clock Domain in Chapter 6, *Power, Reset, and Clock Management (PRCM)*.

31.1.8.4 Trace Captured Into On-chip Trace Buffer

The processor trace flow can be re-directed to the on-chip trace buffer (ETB). The ETB provides on-chip storage of trace information using a 64KB RAM memory.

Through its ATB port, the ETB receives trace flow from CoreSight trace source components. The debugger can then access trace information through DAP-APB.

The ETB can act like a circular buffer, that is, continuously capturing and writing data into memory when trace capture is enabled and either its trigger counter is static at 0, or has not yet decremented to 0.

Trace window can be adjusted around a specific trigger spot using ETB trigger counter:

- Trace before
- Trace after
- Trace around

For more information about ETB, see the ARM CoreSight™ Components Technical Reference Manual.

31.1.9 Crash Dump

Because the ETB is mapped to the emulation power domain, the trace history (last 64KB) can be preserved even while the Cortex-A9 processor domain is shut down. For a post-mortem analysis use case, the ETB memory can be dumped at a later stage independent of the processor domain state. Upon crash alert, ETB trace data is typically moved to external memory.

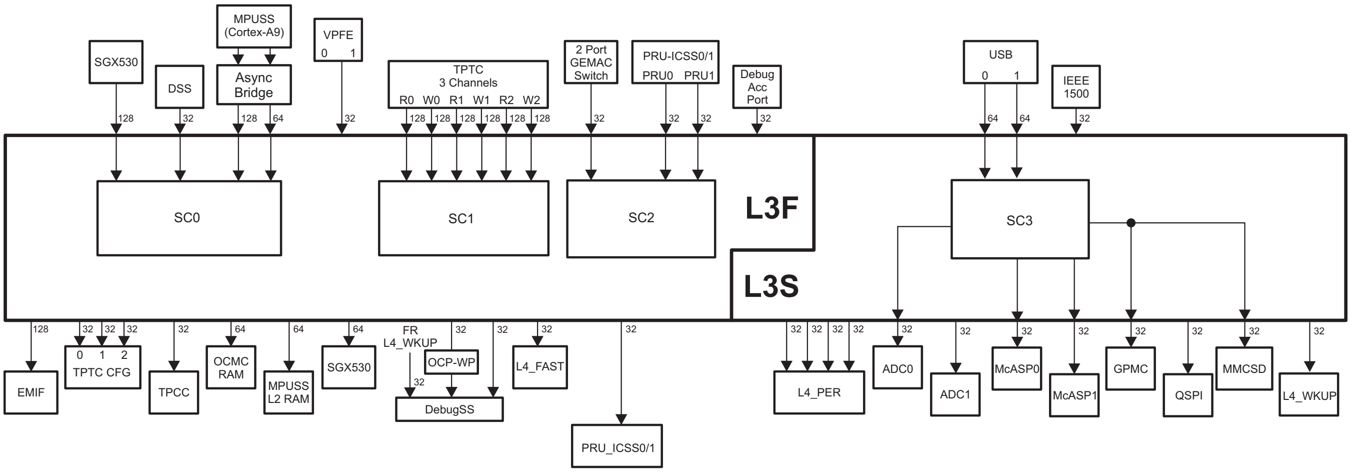
31.2 System Instrumentation

The device supports the following system instrumentation features:

- Real-time software trace (see Section 31.2.4, *Software Instrumentation*)
- OCP target traffic monitoring (see Section 31.2.5, *OCP Watch-Point*)
- OCP target load and master latency monitoring (see Section 31.2.6, *NoC Statistics Collector*)

Figure 31-4 is an overview of the device system instrumentation framework.

Figure 31-4. SoC L3 System Instrumentation Topology



31.2.1 MIPI STM

MIPI-STM is a trace module that aids in software debugging. The main features of this module are:

- Implements MIPI STP protocol (rev 1.0) with the following characteristics:
  - Highly optimized for software-generated traces
  - Automatic timestamping of messages
  - Support for 8-, 16-, and 32-bit data types
- Collects the following information:
  - Software messages
  - Hardware instrumentation trace from hardware agents:
    - OCP-WP
    - L3 NoC CTools System Statistics Collection Units (Statistics Collectors)
- Exports trace data to:
  - Off chip to external trace receiver via device pins
  - On-chip ETB buffer
- Available in 1-, 2-, or 4-pin mode with single- or dual-edge clock, depending on the trace bandwidth requirements and the trace receiver characteristics
- Dedicated 128 × 32-bit FIFO buffer

A maximum of 255 different bus masters can be connected to the STM trace port through a bus arbiter. STP recognizes two distinct modes of tracing, software and hardware types, which use slightly different message combinations to output different types of data. The bus masters can be configured for either type to optimize the system for the different types of trace data.

31.2.2 Trace Exported to an External Trace Receiver

31.2.2.1 Debug Resources Manager

System trace data can be exported to an external trace receiver through the STM module. To achieve this, the debugger or application software must properly program the DRM module.

31.2.2.2 Trace Port Width Configuration

The STM has a configurable export width of 1, 2, or 4 data pins (STM_DATA) plus a dedicated export clock (STM_CLK).

31.2.2.3 Tuning Export Clock Frequency

The device implements two system trace export clock (STM_CLK) generation schemes:

- Trace export clock, generated from programmable DPLL reference clock, shared with application with programmable post-dividers
- Trace export clock, generated from the programmable delay line oscillator (PDLO), embedded in the debug subsystem

31.2.2.3.1 Tuning Export Clock Frequency Through DPLL

The MIPI-STM trace export clock (STM_CLK) can be derived from the CORE DPLL. Selection of the DPLL reference clock is done through the CM module. The CM module also provides a way to tune the STM export clock frequency by dividing it to 1, 2, or 4 at the PRCM level.

31.2.2.3.2 Tuning Export Clock Frequency Through PDLO

The debug subsystem implements a dedicated PDLO module for system trace export clock generation. Because the STM PDLO is functionally equivalent to the processor trace PDLO. For more information about PDLO functionality, see Section 31.1.8.3.3, *Tuning Export Clock Frequency Through PDLO*.

31.2.2.4 Trace Streams Interleaving

Two levels of interleaving system instrumentation flow and arbitration are implemented between instrumentation masters:

- CORE L3 instrumentation interconnect interleaves data coming from the following bus masters:
  - OCP-WP
  - L3 NoC statistics collectors, or software instrumentation (interleaving at the L3 level)
- EMU L3 instrumentation interconnect interleaves data coming from the following bus masters:
  - CORE L3 instrumentation interconnect

31.2.3 Trace Captured into On-chip Trace Buffer

If a trace receiver cannot be attached to the device, or relevant STM trace port pins are unavailable for a particular reason, the user can configure the MIPI-STM module to redirect the STP trace stream to the on-chip trace buffer (ETB) and enable local timestamping. This is accomplished by outputting a local timestamp granularity (LTSG) message, which is a TI addition to the MIPI standard messages.

NOTE: For STM timestamps when output to the ETB, the TS field is no longer the STM FIFO depth, but is the time from the previous message.

Granularity is a function of the instrumentation port clock frequency.

31.2.4 Software Instrumentation

The device provides support for real-time software trace through user-defined message writes to specific, memory mapped register (MMR) locations. Software masters can transmit trace data from the operating system (OS) processes or tasks on 256 different channels, with each channel being defined by the implemented software protocol. The different channels logically group different types of data so that it is easy to filter out the data that is irrelevant to the on-going debugging task. The message structures in STP are optimized to provide an efficient transport for software data through the STM module.

The software masters are:

- Cortex-A9 MPU subsystem
- DAP (for testing purpose)
- Dual PRU processors in PRU-ICSS1
- Dual PRU processors in PRU-ICSS0
- EDMA TPTC WR0 Channels

Each software master has a master-ID assigned to it.

Software messages can be interleaved with other hardware messages. Software messages are intrusive and use both processor cycles and memory.

Table 31-9 describes the master-ID of the software masters.

Table 31-9. STM Message Software Masters

Initiator		MReqMstID			Restriction / Comment
		7	6:02	1:00	
MPU		0	1	0	Cortex-A9 MPU
CS_DAP		0	100	-	STP link testing
PRU0	PRU-ICSS1	0	1100	-	Software messages
	PRU-ICSS0	0	1100	-	
PRU1	PRU-ICSS1	0	1101	-	Software messages
	PRU-ICSS0	0	1101	-	
Wakeup Processor		0	10100	-	Via L4 interconnect
TPTC0_WR		0	11001	-	Data logging

NOTE: The Wakeup Processor can access the instrumentation port via the L4 interconnect. It cannot directly access the instrumentation port via the L3 interconnect.

31.2.5 CTools System Bus Watchpoint and Traffic Monitors (OCP_WP)

The L3 interconnect provides five functional probes, which are embedded and attached to the following L3 targets:

- GPMC
- L4-FAST/SLOW/WAKEUP
- OCMC RAM
- EMIF

The output of all probes is multiplexed together and then sent to the L3 interconnect debug port. A component called OCP-WP is used to collect data from functional probes and then transmit captured data to the STM module. The OCP-WP drives a Probe-ID signal to the L3 interconnect for probe selection. The probe selection is exclusive, meaning that interleaving is not possible.

The OCP-WP provides the following main features:

- Monitoring the OCP traffic originated by all initiators that can access the selected target where the probe is attached
- Filtering OCP monitored bus traffic by:
  - Address range
  - Initiator -ID
  - Transaction type
  - Transaction qualifier
- Generating a trigger upon watch-point match
- Starting and stopping OCP traffic monitoring upon:
  - WP address match
  - External trigger
- OCP-WP messages can be interleaved with software messages
- Programming from:
  - Debugger
  - Application

NOTE: The OCP-WP module is restricted to monitor request flow.

Table 31-10 summarizes the OCP targets that can be monitored by the OCP-WP module and their respective probe-ID.

Table 31-10. L3 Interconnect Functional Probe Mapping

Probe-ID	L3 OCP Target
0	Reserved
1	GPMC
10	L4_FAST
11	L4_SLOW
100	OCMC_RAM
101	L4_WAKEUP
110	EMIF
111	Reserved

The user can program the OCP_WP to extract the traffic from a specific set of initiators (maximum 4 Master-IDs).Table 31-11 shows the Master-ID reported by the L3_MAIN debug port for the device initiators

Table 31-11. L3 Master ID Mapping (Debug View)

Initiator	6-bit MConnID (Debug)
MPUSS M2 (64-bit)	0x01
DAP	0x04
P1500	0x05
PRU-ICSS1 PRU0	0x0C
PRU-ICSS0 PRU0	
PRU-ICSS1 PRU1	0x0D
PRU-ICSS0 PRU1	
Wakeup Processor	0x14
TPTC0 Read	0x18
TPTC0 Write	0x19
TPTC1 Read	0x1A
TPTC1 Write	0x1B
TPTC2 Read	0x1C
TPTC2 Write	0x1D
SGX530	0x20
DSS	0x25
GEMAC	0x30
USB0 Read	0x34
USB0 Write	0x35
USB1 Read	0x36
USB1 Write	0x37
VPFE0	0x2C
VPFE1	0x2D

31.2.6 L3 NoC Statistics Collector

The L3 interconnect supports a built-in performance monitoring feature by implementing a statistics collector (STATCOLL) component, which computes traffic statistics within a user-defined window and periodically reports to the user through the MIPI-STM interface. Four STATCOLL instances are instantiated in the device:

- Load monitoring, see Section 31.2.6.1, *OCP Target Load Monitoring*, for more information
- Master latency monitoring

Statistics collectors can report:

- Average burst length in bytes/packet per sampling window
- Average throughput in bytes per cycle
- Link occupancy on the request link (for store transactions) during a sampling window
- Link occupancy on the response link (for load transactions) during a sampling window
- Arbitration conflict cycles on the request link
- Initiator busy cycles on the response link
- Histogram of payload length in bytes (for example, 0–16 , 16–32, 32–128) for each sampling window.

The performance metrics are interleaved with software instrumentation data at the L3 interconnect level.

The performance monitoring probes implement three main functions:

- Events detection
- Transactions filtering
- Aggregation

The probes can be configured to detect the events summarized in Table 31-12.

Table 31-12. Performance Monitoring Events Detection

Event	Definition
NONE	No event selected
ANY	Any clock cycles
TRANSFER	Word has been accepted by the receiver
WAIT	Transfer has been initiated but the transmitter currently has no data to send
BUSY	Receiver applies flow control
PKT	Transfer of a new packet header
DATA	Transfer of a payload word
IDLES	No communication over the link
LATENCY	Debug bit detection

The probes can be configured to filter the traffic based on the criteria summarized in Table 31-13.

Table 31-13. Performance Filtering Options

Filters	Comments
Master address	Mask and match
Slave address	
User Info	
Read	Opcode is a load
Write	Opcode is a store
Error	-

The probes implement a user-defined set of counters that aggregate the events sampled by the detector and filtered according to the user setup.

NOTE: Statistics collectors counter values are not accessible by application software.

The master address mapping for all statistics collectors are summarized in Table 31-14.

Table 31-14. Statistics Collector Master Address Mapping

Master	Address (hex)
MPUSS M1 (128-bit)	0x00
MPUSS M2 (64-bit)	0x01
DAP	0x04
P1500	0x05
PRU-ICSS 0	0x0C
PRU-ICSS 1	0x0D
Wakeup Processor	0x14
TPTC0 Read	0x18
TPTC0 Write	0x19
TPTC1 Read	0x1A
TPTC1 Write	0x1B
TPTC2 Read	0x1C
TPTC2 Write	0x1D
SGX530	0x20
DSS	0x25

Table 31-14. Statistics Collector Master Address Mapping (continued)

Master	Address (hex)
GEMAC	0x30
USB0 Read	0x34
USB0 Write	0x35
USB1 Read	0x36
USB1 Write	0x37
VPFE1	0x2C
VPFE2	0x2D

The slave address mapping for all statistics collectors are summarized in Table 31-15.

Table 31-15. Statistics Collector Slave Address Mapping

Slave	Address (hex)
Host200F	0
emif_targ	1
Reserved	2
l2ram_targ	3
Reserved	4
l4_fast_targ	5
exp_targ	6
tptc_targ0	7
tptc_targ1	8
tptc_targ2	9
adc0_targ	A
tpcc_targ	B
Host100S	C
l4_wkup_targ	D
sgx530_targ	E
adc1_card_targ	F
ocmcram0_targ	10
l4_per_targ0	11
l4_per_targ1	12
l4_per_targ2	13
l4_per_targ3	14
Reserved	15
Reserved	1B
gpmc_targ	1E
debug_targ	1F
mcasp_targ0	20
mcasp_targ1	21
mmcsd2_targ	26
pru_icss_targ	1A
qspi_targ	1C

Table 31-16 summarizes the performance probe aggregation modes.

Table 31-16. Performance Filtering Options

Aggregation Mode	Description
FILTER_HIT	The counter increments by 1 when filter hits.
MIN_MAX_HIT	The counter increments by 1 when the filter hits and the sleeted event information is within range. • Payload length (bytes) • Pressure value • Request and response latency (clock cycles)
EVT_INFO	The selected event information is added to the counter value when the filter hits.
AND_FILTER	The counter increments by 1 when all unit filters hit
OR-FILTER	The counter increments by 1 when at least one unit filter hit
SUM_REQ_EVT	The counter sums the events from any request port
SUM_RSP_EVT	The counter sums the events from any response port
SUM_ALL_EVT	The counter sums the events from any port.
EXT_EVT	The counter increments by 1 when selected external event input signal is sampled high.

31.2.6.1 OCP Target Load Monitoring

The L3 interconnect uses performance monitoring probes on target (slave) interfaces. The OCP traffic statistics are computed within a user-defined window and periodically reported to the user through the MIPI-STM interface.

The performance metrics, the software events, and the system hardware events being exported through a unified export channel, it allows correlating latency trends versus on-going execution and system context.

31.2.6.2 Statistics Collectors Configuration

The device instantiates four statistics collectors in the L3 interconnect. Table 31-17, Table 31-18, Table 31-19, Tables below, and Tables below describe the performance probes and configuration for each statistics collector.

All the statistic collectors will dump frames:

- At slave address 0x1F (DEBUGSS). This cannot be changed.
- By default at slave offset 0x800 inside DEBUGSS address range.

Table 31-17. Statistics Collector Counters

statcoll	Number of Counters	Domains	Bits of Counters	Identifier
statcoll_0	4	L3 Fast	12	0x0
statcoll_1	6	L3 Fast	12	0x1
statcoll_2	4	L3 Fast	12	0x2
statcoll_3	4	L3 Slow	12	0x3

Table 31-18. Statistics Collector 0 Probes

Probe #	Description	Link	Port #
0	MPU Subsystem (128-bit)	NTTP REQ	0
		NTTP RSP	1
1	MPU Subsystem (64-bit)	NTTP REQ	2
		NTTP RSP	3
2	SGX530	NTTP REQ	4
		NTTP RSP	5

Table 31-18. Statistics Collector 0 Probes (continued)

Probe #	Description	Link	Port #
3	DSS	NTTP REQ	6
		NTTP RSP	7
4	EMIF	NTTP REQ	8
		NTTP RSP	9

Table 31-19. Statistics Collector 1 Probes

Probe #	Description	Link	Port #
0	TPTC RD0	NTTP REQ	0
		NTTP RSP	1
1	TPTC RD1	NTTP REQ	2
		NTTP RSP	3
2	TPTC RD2	NTTP REQ	4
		NTTP RSP	5
3	TPTC WR0	NTTP REQ	6
		NTTP RSP	7
4	TPTC WR1	NTTP REQ	8
		NTTP RSP	9
5	TPTC WR2	NTTP REQ	10
		NTTP RSP	11

Table 31-20. Statistics Collector 2 Probes

Probe #	Description	Link	Port #
0	PRU-ICSS PRU1	NTTP REQ	0
		NTTP RSP	1
2	GEMAC SW	NTTP REQ	4
		NTTP RSP	5
3	PRU-ICSS PRU0	NTTP REQ	6
		NTTP RSP	7

Table 31-21. Statistics Collector 3 Probes

Probe #	Description	Link	Port #
0	USB 0	NTTP REQ	0
		NTTP RSP	1
1	USB 0	NTTP REQ	2
		NTTP RSP	3
2	GPMC	NTTP REQ	4
		NTTP RSP	5

Table 31-21. Statistics Collector 3 Probes (continued)

Probe #	Description	Link	Port #
3	MMCSD2	NTTP REQ	6
		NTTP RSP	7
4	McASP0	NTTP REQ	8
		NTTP RSP	9
5	McASP1	NTTP REQ	10
		NTTP RSP	11
6	USB1	NTTP REQ	12
		NTTP RSP	13
7	USB1	NTTP REQ	14
		NTTP RSP	15

31.2.7 Hardware Masters

The Master-ID of Hardware message reaching to STM module is described in Table 31-22.

Table 31-22. Master-ID for Hardware Masters

Initiator	MReqMstID			Restriction / Comment
	7	6:02	1:00	
OCP_WP	1	0	0	Traffic probe
OCP_WP	1	1	0	DMA profiling
OCP_WP	1	10	0	System event
Stat_Coll 0	1	11100	0	Statistics Collector 0
Stat_Coll 1	1	11101	0	Statistics Collector 1
Stat_Coll 2	1	11110	0	Statistics Collector 2
Stat_Coll 3	1	11111	0	Statistics Collector 3

31.3 Concurrent Debug Mode

The debugger or application software can program the DRM to route a specific debug function to each device debug port pin. Because of the limited number of pins allocated to debug, debug and trace source signals are multiplexed.

Table 31-23 summarizes the trace port configuration.

Table 31-23. Trace Port Configuration

Pin Name	Internal Signal Name	I/O	Trigger	JTAG 20-pin Header	CS_TPIU (MPU Trace)		STM (System Trace)
					8-bit Mode	10-bit Mode	
dmp_emu11	EMU11	O			TRACEDATA[7]	TRACEDATA[9]	STM_DATA[x]/ STM_CLK(1)
dmp_emu10	EMU10	O			TRACEDATA[6]	TRACEDATA[8]	STM_DATA[x]/ STM_CLK(1)

(1) STM data and clock can be configured on any debug pin that is available to support various devices with limited pin options.

Table 31-23. Trace Port Configuration (continued)

Pin Name	Internal Signal Name	I/O	Trigger	JTAG 20-pin Header	CS_TPIU (MPU Trace)		STM (System Trace)
					8-bit Mode	10-bit Mode	
dmp_emu9	EMU9	O			TRACEDATA[5]	TRACEDATA[7]	STM_DATA[x]/ STM_CLK (1)
dmp_emu8	EMU8	I/O			TRACEDATA[4]	TRACEDATA[6]	STM_DATA[x]/ STM_CLK (1)
dmp_emu7	EMU7	I/O			TRACEDATA[3]	TRACEDATA[5]	STM_DATA[x]/ STM_CLK (1)
dmp_emu6	EMU6	I/O			TRACEDATA[2]	TRACEDATA[4]	STM_DATA[x]/ STM_CLK (1)
dmp_emu5	EMU5	I/O			TRACEDATA[1]	TRACEDATA[3]	STM_DATA[x]/ STM_CLK (1)
dmp_emu4	EMU4	O			TRACEDATA[0]	TRACEDATA[2]	STM_DATA[x]/ STM_CLK (1)
dmp_emu3	EMU3	O			TRACECTL	TRACECTL	STM_DATA[x]/ STM_CLK (1)
dmp_emu2	EMU2	O			TRACECLK	TRACECLK	STM_DATA[x]/ STM_CLK (1)
dmp_emu1	EMU1	I/O	Trigger 1	EMU1		TRACEDATA[1]	STM_DATA[x]/ STM_CLK (1)
dmp_emu0	EMU0	I/O	Trigger 0	EMU0		TRACEDATA[0]	STM_DATA[x]/ STM_CLK (1)

NOTE: The configuration of the trace port must comply with Table 31-23; otherwise, it will be ignored by DRM hardware. For example, if Trigger0 is programmed on EMU3 and Trigger1 is programmed on EMU4, this configuration will be ignored.

The application or debugger software can program the DRM to route a specific debug function to each debug port pin. The programming model is making provision for debug support enhancement and reuse by other platforms.

Table 31-24 summarizes the concurrent debug and trace use cases.

Table 31-24. Concurrent Debug and Trace

Debug Use case	Concurrent Debug flows	Debug pins	Trace pins		
		Triggers	Data	Control	Clock
0	PTM		10	1	1
	STM				
	Triggers				
1	PTM		8	1	1
	STM				
	Triggers	2			
2	PTM		8	1	1
	STM		1	-	1
	Triggers				
3	PTM		3	1	1
	STM		4	-	1
	Triggers	2			

31.4 Memory Mapping

Table 31-25 summarizes the memory mapping of the debug modules.

Table 31-25. Debug Modules Memory Mapping

Memory Space	Module Name	Start Address (hex)	End Address (hex)	Size
L3_EMU	MIPI-STM (256 × 4K channels) (address space 0)	0x4B00_0000	0x4B0F_FFFF	1MB
	MIPI-STM (256 × 1K channels) (address space 1)	0x4B10_0000	0x4B13_FFFF	256KB
	Cortex-A9 PTM	0x4B14_0000	0x4B00_0FFF	4KB
	Cortex-A9 Debug	0x4B14_1000	0x4B00_1FFF	4KB
	Cortex-A9 CTI	0x4B14_2000	0x4B00_2FFF	4KB
	DRM	0x4B16_0000	0x4B16_0FFF	4KB
	MIPI-STM	0x4B16_1000	0x4B16_1FFF	4KB
	CS-ETB	0x4B16_2000	0x4B16_3FFF	4KB
	CS-TF	0x4B16_4000	0x4B16_2FFF	4KB
L4_PER	OCP-WP	0x4818_C000	0x4818_CFFF	4KB
	PRU-ICSS	0x5440_0000	0x547F_FFFF	512KB
L4_WAKEUP	INSTRUMENTATION for Wakeup Processor	0x44E8_0000	0x44E8_0FFF	4KB
L3 configuration	SC_LAT0	0x4400_2000	0x4400_2FFF	4KB
	SC_LAT1	0x4400_3000	0x4400_3FFF	4KB
	SC_LAT2	0x4400_4000	0x4400_4FFF	4KB
	SC_LAT3	0x4480_4000	0x4480_4FFF	4KB
PRU-ICSS	PRU_0 debug(1)	0x0002_2400	0x0002_3FFF	7KB
	PRU_1 debug(1)	0x0002_4400	0x0002_5FFF	7KB

(1) Private memory access per PRU-ICSS

Table 31-26. Debug Modules Memory Mapping (APB-AP View)

Module Name	State Address (hex)	End Address (hex)	Size
Cortex-A9 Debug	0x8000_0000	0x8000_0FFF	4KB
Cortex-A9 CTI	0x8000_8000	0x8000_8FFF	4KB
Cortex-A9 PTM	0x8000_C000	0x8000_CFFF	4KB

31.5 DRM Registers

Table 31-27 lists the memory-mapped registers for the DRM. All register offset addresses not listed in Table 31-27 should be considered as reserved locations and the register contents should not be modified.

Table 31-27. DRM Registers

Offset	Acronym	Register Name	Section
200h	DEBUGSS_DRM_SUSPEND_CTRL0	Watchdog Timer 1 (WDT1) Suspend Control	Section 31.5.1
204h	DEBUGSS_DRM_SUSPEND_CTRL1	DMTimer0 Suspend Control	Section 31.5.2
208h	DEBUGSS_DRM_SUSPEND_CTRL2	DMTimer1 Suspend Control	Section 31.5.3
20Ch	DEBUGSS_DRM_SUSPEND_CTRL3	DMTimer2 Suspend Control	Section 31.5.4
210h	DEBUGSS_DRM_SUSPEND_CTRL4	DMTimer3 Suspend Control	Section 31.5.5
214h	DEBUGSS_DRM_SUSPEND_CTRL5	DMTimer4 Suspend Control	Section 31.5.6
218h	DEBUGSS_DRM_SUSPEND_CTRL6	DMTimer5 Suspend Control	Section 31.5.7
21Ch	DEBUGSS_DRM_SUSPEND_CTRL7	DMTimer6 Suspend Control	Section 31.5.8
220h	DEBUGSS_DRM_SUSPEND_CTRL8	EMAC Suspend Control	Section 31.5.9
228h	DEBUGSS_DRM_SUSPEND_CTRL10	I2C0 Suspend Control	Section 31.5.10

Table 31-27. DRM Registers (continued)

Offset	Acronym	Register Name	Section
22Ch	DEBUGSS_DRM_SUSPEND_CTRL11	I2C1 Suspend Control	Section 31.5.11
230h	DEBUGSS_DRM_SUSPEND_CTRL12	I2C2 Suspend Control	Section 31.5.12
234h	DEBUGSS_DRM_SUSPEND_CTRL13	eHRPWM0 Suspend Control	Section 31.5.13
238h	DEBUGSS_DRM_SUSPEND_CTRL14	eHRPWM1 Suspend Control	Section 31.5.14
23Ch	DEBUGSS_DRM_SUSPEND_CTRL15	eHRPWM2 Suspend Control	Section 31.5.15
240h	DEBUGSS_DRM_SUSPEND_CTRL16	DCAN0 Suspend Control	Section 31.5.16
244h	DEBUGSS_DRM_SUSPEND_CTRL17	DCAN1 Suspend Control	Section 31.5.17
248h	DEBUGSS_DRM_SUSPEND_CTRL18	PRU-ICSS Suspend Control	Section 31.5.18
24Ch	DEBUGSS_DRM_SUSPEND_CTRL19	SyncTimer Suspend Control	Section 31.5.19
260h	DEBUGSS_DRM_SUSPEND_CTRL24	DMTimer7-11 Suspend Control	Section 31.5.20
26Ch	DEBUGSS_DRM_SUSPEND_CTRL27	PWM3 Suspend Control	Section 31.5.21
270h	DEBUGSS_DRM_SUSPEND_CTRL28	PWM4 Suspend Control	Section 31.5.22
274h	DEBUGSS_DRM_SUSPEND_CTRL29	PWM5 Suspend Control	Section 31.5.23

31.5.1 DEBUGSS_DRM_SUSPEND_CTRL0 Register (Offset = 200h) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL0 is shown in Figure 31-5 and described in Table 31-28.

Return to Summary Table.

Figure 31-5. DEBUGSS_DRM_SUSPEND_CTRL0 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>SUSPEND_SEL</td></tr>
<tr><td colspan="7">R-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">SUSPEND_SEL</td><td>SUSPEND_DEFAULT_OVERRIDE</td><td colspan="2">RESERVED</td><td>SENSCTRL</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R-0h</td><td colspan="2">R-0h</td><td>R/W-0h</td></tr>
</table>

Table 31-28. DEBUGSS_DRM_SUSPEND_CTRL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.2 DEBUGSS_DRM_SUSPEND_CTRL1 Register (Offset = 204h) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL1 is shown in Figure 31-6 and described in Table 31-29.

Return to Summary Table.

Figure 31-6. DEBUGSS_DRM_SUSPEND_CTRL1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							SUSPEND_SEL
R-0h							R/W-0h
7	6	5	4	3	2	1	0
SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL
R/W-0h				R-0h	R-0h		R/W-0h

Table 31-29. DEBUGSS_DRM_SUSPEND_CTRL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.3 DEBUGSS_DRM_SUSPEND_CTRL2 Register (Offset = 208h) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL2 is shown in Figure 31-7 and described in Table 31-30.

Return to Summary Table.

Figure 31-7. DEBUGSS_DRM_SUSPEND_CTRL2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							SUSPEND_SEL
R-0h							R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL
R/W-0h				R-0h	R-0h		R/W-0h

Table 31-30. DEBUGSS_DRM_SUSPEND_CTRL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.4 DEBUGSS_DRM_SUSPEND_CTRL3 Register (Offset = 20Ch) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL3 is shown in Figure 31-8 and described in Table 31-31.

Return to Summary Table.

Figure 31-8. DEBUGSS_DRM_SUSPEND_CTRL3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							SUSPEND_SEL
R-0h							R/W-0h
7	6	5	4	3	2	1	0
SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL
R/W-0h				R-0h	R-0h		R/W-0h

Table 31-31. DEBUGSS_DRM_SUSPEND_CTRL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.5 DEBUGSS_DRM_SUSPEND_CTRL4 Register (Offset = 210h) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL4 is shown in Figure 31-9 and described in Table 31-32.

Return to Summary Table.

Figure 31-9. DEBUGSS_DRM_SUSPEND_CTRL4 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							SUSPEND_SEL
R-0h							R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL
R/W-0h				R-0h	R-0h		R/W-0h

Table 31-32. DEBUGSS_DRM_SUSPEND_CTRL4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.6 DEBUGSS_DRM_SUSPEND_CTRL5 Register (Offset = 214h) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL5 is shown in Figure 31-10 and described in Table 31-33.

Return to Summary Table.

Figure 31-10. DEBUGSS_DRM_SUSPEND_CTRL5 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							SUSPEND_SEL
R-0h							R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL
R/W-0h				R-0h	R-0h		R/W-0h

Table 31-33. DEBUGSS_DRM_SUSPEND_CTRL5 Register Field Descriptions

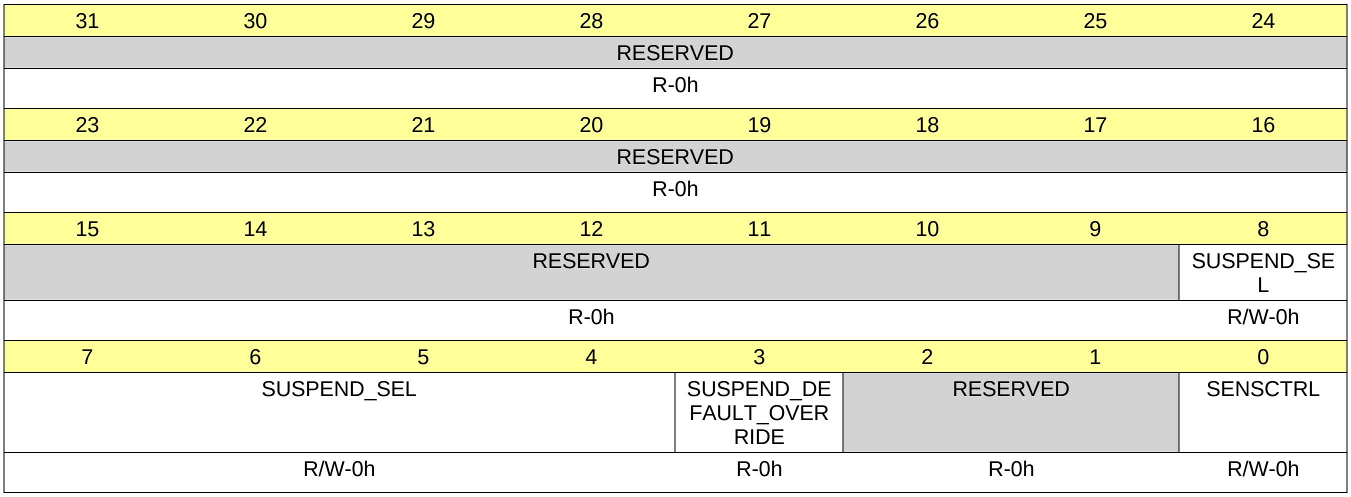
Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.7 DEBUGSS_DRM_SUSPEND_CTRL6 Register (Offset = 218h) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL6 is shown in Figure 31-11 and described in Table 31-34.

Return to Summary Table.

Figure 31-11. DEBUGSS_DRM_SUSPEND_CTRL6 Register



<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>SUSPEND_SEL</td></tr>
<tr><td colspan="7">R-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">SUSPEND_SEL</td><td>SUSPEND_DEFAULT_OVERRIDE</td><td colspan="2">RESERVED</td><td>SENSCTRL</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R-0h</td><td colspan="2">R-0h</td><td>R/W-0h</td></tr>
</table>

Table 31-34. DEBUGSS_DRM_SUSPEND_CTRL6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.8 DEBUGSS_DRM_SUSPEND_CTRL7 Register (Offset = 21Ch) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL7 is shown in Figure 31-12 and described in Table 31-35.

Return to Summary Table.

Figure 31-12. DEBUGSS_DRM_SUSPEND_CTRL7 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							SUSPEND_SEL
R-0h							R/W-0h
7	6	5	4	3	2	1	0
SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL
R/W-0h				R-0h	R-0h		R/W-0h

Table 31-35. DEBUGSS_DRM_SUSPEND_CTRL7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.9 DEBUGSS_DRM_SUSPEND_CTRL8 Register (Offset = 220h) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL8 is shown in Figure 31-13 and described in Table 31-36.

Return to Summary Table.

Figure 31-13. DEBUGSS_DRM_SUSPEND_CTRL8 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							SUSPEND_SEL
R-0h							R/W-0h
7	6	5	4	3	2	1	0
SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL
R/W-0h				R-0h	R-0h		R/W-0h

Table 31-36. DEBUGSS_DRM_SUSPEND_CTRL8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.10 DEBUGSS_DRM_SUSPEND_CTRL10 Register (Offset = 228h) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL10 is shown in Figure 31-14 and described in Table 31-37.

Return to Summary Table.

Figure 31-14. DEBUGSS_DRM_SUSPEND_CTRL10 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							SUSPEND_SEL
R-0h							R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL
R/W-0h				R-0h	R-0h		R/W-0h

Table 31-37. DEBUGSS_DRM_SUSPEND_CTRL10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.11 DEBUGSS_DRM_SUSPEND_CTRL11 Register (Offset = 22Ch) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL11 is shown in Figure 31-15 and described in Table 31-38.

Return to Summary Table.

Figure 31-15. DEBUGSS_DRM_SUSPEND_CTRL11 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>SUSPEND_SEL</td></tr>
<tr><td colspan="7">R-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">SUSPEND_SEL</td><td>SUSPEND_DEFAULT_OVERRIDE</td><td colspan="2">RESERVED</td><td>SENSCTRL</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R-0h</td><td colspan="2">R-0h</td><td>R/W-0h</td></tr>
</table>

Table 31-38. DEBUGSS_DRM_SUSPEND_CTRL11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.12 DEBUGSS_DRM_SUSPEND_CTRL12 Register (Offset = 230h) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL12 is shown in Figure 31-16 and described in Table 31-39.

Return to Summary Table.

Figure 31-16. DEBUGSS_DRM_SUSPEND_CTRL12 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							SUSPEND_SEL
R-0h							R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL
R/W-0h				R-0h	R-0h		R/W-0h

Table 31-39. DEBUGSS_DRM_SUSPEND_CTRL12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.13 DEBUGSS_DRM_SUSPEND_CTRL13 Register (Offset = 234h) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL13 is shown in Figure 31-17 and described in Table 31-40.

Return to Summary Table.

Figure 31-17. DEBUGSS_DRM_SUSPEND_CTRL13 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>SUSPEND_SEL</td></tr>
<tr><td colspan="7">R-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">SUSPEND_SEL</td><td>SUSPEND_DEFAULT_OVERRIDE</td><td colspan="2">RESERVED</td><td>SENSCTRL</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R-0h</td><td colspan="2">R-0h</td><td>R/W-0h</td></tr>
</table>

Table 31-40. DEBUGSS_DRM_SUSPEND_CTRL13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.14 DEBUGSS_DRM_SUSPEND_CTRL14 Register (Offset = 238h) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL14 is shown in Figure 31-18 and described in Table 31-41.

Return to Summary Table.

Figure 31-18. DEBUGSS_DRM_SUSPEND_CTRL14 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							SUSPEND_SEL
R-0h							R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL
R/W-0h				R-0h	R-0h		R/W-0h

Table 31-41. DEBUGSS_DRM_SUSPEND_CTRL14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.15 DEBUGSS_DRM_SUSPEND_CTRL15 Register (Offset = 23Ch) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL15 is shown in Figure 31-19 and described in Table 31-42.

Return to Summary Table.

Figure 31-19. DEBUGSS_DRM_SUSPEND_CTRL15 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>SUSPEND_SEL</td></tr>
<tr><td colspan="7">R-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">SUSPEND_SEL</td><td>SUSPEND_DEFAULT_OVERRIDE</td><td colspan="2">RESERVED</td><td>SENSCTRL</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R-0h</td><td colspan="2">R-0h</td><td>R/W-0h</td></tr>
</table>

Table 31-42. DEBUGSS_DRM_SUSPEND_CTRL15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.16 DEBUGSS_DRM_SUSPEND_CTRL16 Register (Offset = 240h) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL16 is shown in Figure 31-20 and described in Table 31-43.

Return to Summary Table.

Figure 31-20. DEBUGSS_DRM_SUSPEND_CTRL16 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>SUSPEND_SEL</td></tr>
<tr><td colspan="7">R-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">SUSPEND_SEL</td><td>SUSPEND_DEFAULT_OVERRIDE</td><td colspan="2">RESERVED</td><td>SENSCTRL</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R-0h</td><td colspan="2">R-0h</td><td>R/W-0h</td></tr>
</table>

Table 31-43. DEBUGSS_DRM_SUSPEND_CTRL16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.17 DEBUGSS_DRM_SUSPEND_CTRL17 Register (Offset = 244h) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL17 is shown in Figure 31-21 and described in Table 31-44.

Return to Summary Table.

Figure 31-21. DEBUGSS_DRM_SUSPEND_CTRL17 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							SUSPEND_SEL
R-0h							R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL
R/W-0h				R-0h	R-0h		R/W-0h

Table 31-44. DEBUGSS_DRM_SUSPEND_CTRL17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.18 DEBUGSS_DRM_SUSPEND_CTRL18 Register (Offset = 248h) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL18 is shown in Figure 31-22 and described in Table 31-45.

Return to Summary Table.

Figure 31-22. DEBUGSS_DRM_SUSPEND_CTRL18 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>SUSPEND_SEL</td></tr>
<tr><td colspan="7">R-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">SUSPEND_SEL</td><td>SUSPEND_DEFAULT_OVERRIDE</td><td colspan="2">RESERVED</td><td>SENSCTRL</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R-0h</td><td colspan="2">R-0h</td><td>R/W-0h</td></tr>
</table>

Table 31-45. DEBUGSS_DRM_SUSPEND_CTRL18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.19 DEBUGSS_DRM_SUSPEND_CTRL19 Register (Offset = 24Ch) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL19 is shown in Figure 31-23 and described in Table 31-46.

Return to Summary Table.

Figure 31-23. DEBUGSS_DRM_SUSPEND_CTRL19 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							SUSPEND_SEL
R-0h							R/W-0h
7	6	5	4	3	2	1	0
SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL
R/W-0h				R-0h	R-0h		R/W-0h

Table 31-46. DEBUGSS_DRM_SUSPEND_CTRL19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.20 DEBUGSS_DRM_SUSPEND_CTRL24 Register (Offset = 260h) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL24 is shown in Figure 31-24 and described in Table 31-47.

Return to Summary Table.

Figure 31-24. DEBUGSS_DRM_SUSPEND_CTRL24 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							SUSPEND_SEL
R-0h							R/W-0h
7	6	5	4	3	2	1	0
SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL
R/W-0h				R-0h	R-0h		R/W-0h

Table 31-47. DEBUGSS_DRM_SUSPEND_CTRL24 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.21 DEBUGSS_DRM_SUSPEND_CTRL27 Register (Offset = 26Ch) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL27 is shown in Figure 31-25 and described in Table 31-48.

Return to Summary Table.

Figure 31-25. DEBUGSS_DRM_SUSPEND_CTRL27 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							SUSPEND_SEL
R-0h							R/W-0h
7	6	5	4	3	2	1	0
SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL
R/W-0h				R-0h	R-0h		R/W-0h

Table 31-48. DEBUGSS_DRM_SUSPEND_CTRL27 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.22 DEBUGSS_DRM_SUSPEND_CTRL28 Register (Offset = 270h) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL28 is shown in Figure 31-26 and described in Table 31-49.

Return to Summary Table.

Figure 31-26. DEBUGSS_DRM_SUSPEND_CTRL28 Register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RESERVED</td></tr>
<tr><td colspan="8">R-0h</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="7">RESERVED</td><td>SUSPEND_SEL</td></tr>
<tr><td colspan="7">R-0h</td><td>R/W-0h</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">SUSPEND_SEL</td><td>SUSPEND_DEFAULT_OVERRIDE</td><td colspan="2">RESERVED</td><td>SENSCTRL</td></tr>
<tr><td colspan="4">R/W-0h</td><td>R-0h</td><td colspan="2">R-0h</td><td>R/W-0h</td></tr>
</table>

Table 31-49. DEBUGSS_DRM_SUSPEND_CTRL28 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

31.5.23 DEBUGSS_DRM_SUSPEND_CTRL29 Register (Offset = 274h) [reset = 0h]

DEBUGSS_DRM_SUSPEND_CTRL29 is shown in Figure 31-27 and described in Table 31-50.

Return to Summary Table.

Figure 31-27. DEBUGSS_DRM_SUSPEND_CTRL29 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
R-0h							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							SUSPEND_SEL
R-0h							R/W-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL
R/W-0h				R-0h	R-0h		R/W-0h

Table 31-50. DEBUGSS_DRM_SUSPEND_CTRL29 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-4	SUSPEND_SEL	R/W	0h	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when Suspend_Default_Override=0 and SensCtrl=1. When read, these bits reflect the default suspend signal. 0000b: Cortex-A9 suspend signal. All other values are reserved.
3	SUSPEND_DEFAULT_OVERRIDE	R	0h	Enable or disable the override value in Suspend_Sel. 0: Suspend_Sel field will select which suspend signal reaches the peripheral. 1: Suspend_Sel field ignored. Default suspend signal will reach the peripheral.
2-1	RESERVED	R	0h	
0	SENSCTRL	R/W	0h	Sensitivity Control for suspend signals. When Suspend_Default_Override=1, this bit is ignored and read as a 1. When Suspend_Default_Override=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.

Appendix A

SPRUHL7H–April 2014–Revised January 2018

Glossary

Numerical

2D— Two dimensional

3D— Three dimensional

3GPP— 3rd Generation Partnership Project

A

ABS— Absolute value

ACA —Accessory Charger Adaptor

ADC— Analog-to-digital converter/conversion

A-dev —A USB OTG device with an A-plug connected to its port (ID pin grounded). Default host

ADP —Attach Detection Protocol: detects USB OTG attach/detach events.

AEW— Adaptive multirate

AHB— Advanced high-performance bus

AMR— Adaptive multirate; also Audio modem riser: An Intel specification that defines a new architecture for the design of motherboards.

APB— Advanced peripheral bus

APE— Applicative processor engine

API— Application programming interface; also ARM port interface

ARAU— Auxiliary register arithmetic unit

ARGB— Alpha, red, green, blue

ASIC— Application-specific integrated circuit: A chip built for a specific application. In the context of this document, ASIC refers to the FPGA that resides on the EVM board.

ASCII— American standard code for information

ATR— Answer to reset

AVC— Advanced video coding (MPEG4 - Part 10 also known as H264)

AXI— Advanced extensible interface

B

BB— Busy bus

BCD— Binary-coded decimal: A representation of decimal digits (0–9) using a nibble that uses a certain number of bits. For example, using 4 bits, two BCDs can be packed into 1 byte.

B-dev —A USB OTG device with a B-plug connected to its port (ID pin floating). Default peripheral.

BER — Bit Error Rate.

BGA— Ball grid array

BGAPTS— Band gap voltage and temperature sensor

BGT— Block guard time

BIOS— Built-in operating system

BIST— Built-in self-test

Bluetooth®— A short-range radio technology designed to simplify communications among network devices and between devices and the Internet. It also simplifies data synchronization between network devices and other computers.

BPP— Bits per pixel

BTA— Bus turn around

C

CALU— Central arithmetic logic unit

CAVLC— Context-adaptive variable length coder

CAVLD— Context-adaptive variable length decoder

CBUFF— Circular buffer

CCP— Compact camera port

CDC — Communication Device Class: USB device class.

CDL— Controlled delay line

CE— Chip enable

CH— Configuration header. To use settings other than ROM code defaults, (that is, clock frequencies, SDRAM/DDRAM settings, or GPMC settings).

CID— Card identification number

CIF— Common intermediate format. A video format used in video conferencing systems that easily supports both NTSC and PAL signals. CIF is part of the ITU H.261 video conferencing standard. It specifies a data rate of 30 fps, with each frame containing 288 lines and 352 pixels per line.

CLE— Command latch enable

CLK— Clock

CLUT— Color look-up table

CMOS— Complementary metal oxide semiconductor

CMT— Cellular mobile telephone

Codec— Coder/decoder or compression/decompression: A device that codes in one direction of transmission and decodes in another direction of transmission.

ConnID— Connection identifier: An initiator module identifier. A ConnID transmitted in-band with the request and is used for protection and error logging mechanism.

CP15— Coprocessor 15: This coprocessor controls the operation and configuration of the TI925T.

CPSR— Current program status register

CPU— Central processing unit: The CPU is the portion of the processor involved in arithmetic, shifting, and Boolean logic operations, as well as the generation of data and program memory addresses. The CPU includes the CALU, the multiplier, and the ARAU.

CRC— Cyclic redundancy check

CS— Chip-select

CTRL— Control

CVBS— Composite video broadcast signal

CWT— Command wire tracer

D

D2D— Die-to-die

DAC— Digital-to-analog converter

DBB— Digital baseband

DBI— Display date interface

DCS— Display command set

DCO— Digitally controlled oscillator

DCT— Discrete cosine transform: A fast Fourier transform used to manipulate compressed still and moving picture data.

DDR— Dual data rate

DE— Date enable

DFF— Digital flip-flop

DFT— Design for test

DI— Data identifier

DISPC— Display controller

DLL— Delay-locked loop

DMA— Direct memory access: A mechanism whereby a device other than the host processor contends for and receives mastery of the memory bus so that data transfers can occur independent of the host.

DMC— Data memory controller

DPCM— Differential pulse code modulation

DPD, DPDM— Deep power-down mode

DPF— Dynamic power framework

DPI— Display parallel interface. Digital implementation of PLL

DPLL— Digital phase-locked loop

DRD— Dual role device: USB Host and Peripheral capable.

DRDY— Data ready

DRM— Digital rights management

DS — Down Stream (A USB port facing from a Host or Hub to a Device/Peripheral).

DSI— Display serial interface

DSP— Digital signal processor: A semiconductor that manipulates discrete or discontinuous electrical impulses in a manner that implements a desired algorithm.

DSS— Display subsystem

DT— Data type

DVFS— Dynamic voltage and frequency scaling

E

EAV— End of active video

ECC— Error checking and correction; also error correction code.

ED— Endpoint descriptor

EDC— Error detection code

eFuse— Electrical fuse: A one-time programmable memory location usually set at the factory

EHCI— Enhanced host controller interface

EMC— External memory controller

EMI— Electromagnetic interference

EMV— Initial letters of Europay, MasterCard, and VISA, the three companies which originally cooperated to develop the EMV standard

EOF— End of frame

EOI —End of interrupt

EOP —End of packet

EOT— End of transfer

EP — Endpoint: USB communication channel between the USB link partners carrying a single transfer type (BULK, ISOCH, INT, or CONTROL) and bus sharing arbitration scheme.

ES— Erase status

ETB— Embedded trace buffer

ETSI— European Telecommunications Standard Institute

ETM— Embedded trace macrocell

F

FC— Frame counter

FE— Framing error: An error that occurs when the asynchronous serial port receives a data character that does not have a valid stop-bit.

FIFO— First in first out: A queue; a data structure or hardware buffer from which items are removed in the same order they were put in. A FIFO is useful for buffering a stream of data between a sender and receiver which are not synchronized; that is, the sender and receiver are not sending and receiving at exactly the same rate. If the rates differ by too much in one direction for too long, the FIFO becomes either full (blocking the sender) or empty (blocking the receiver).

FIQ— Fast interrupt request

FlatLink™ 3G— A Texas Instruments display interface protocol

FS— Frame synchronization or Full-Speed USB data rate (12 Mbps).

FSM— Finite state-machine: A model of computation consisting of a set of states, a start state, an input alphabet, and a transition function that maps input symbols and current states to a next state.

FSR— Fault status register

FT— Feed through

G

GDD— Generic distributed DMA

GP device— General-purpose device

GPIO— General-purpose input/output: Pins that can accept input signals and/or send output signals but are not linked to specific uses.

GPMC— General-purpose memory controller

GSM— Global system for mobile communications

H

HC— Host controller

HCD — Host Controller Driver, designates the USB SW.

HDTV— High-definition television

HFP— Horizontal Front Porch

HNP — Host Negotiation Protocol, OTG extension to swap USB host and peripheral roles.

HPI— Host port interface

HS — High-Speed USB data rate (480 Mbps).

HSEC— Horizontal sync end code

HSSC— Horizontal sync start code

HSYNC— Horizontal synchronization: A bidirectional horizontal timing signal occurring once per line with a pulse width defined as an integral number of FCLK periods. Synchronization signals can be used to enable retrace of the electron beam of a display screen. Also HS.

HSW— Horizontal Synchronization Pulse Width

HVGA— Half-size video graphics array. One-half the resolution of VGA

HW, H/W— Hardware

HWA— Hardware accelerators

I

I^{2}C— Inter-integrated circuit: A multimaster bus where multiple chips can be connected. Each chip can act as a master by initiating a data transfer.

I2S— Inter-IC sound: A digital audio interface standard.

IA— Initiator agent, also Identifier address

IC— Integrated circuit

ICR— Intersystem communication registers

IF— Interface

INT— Interrupt: A signal sent by hardware or software to a processor requesting attention. An interrupt tells the processor to suspend its current operation, save the current task status, and perform a particular set of instructions. Interrupts communicate with the operating system and prioritize tasks to be performed.

INTC— Interrupt controller

I/O— Input/output

IP— Intellectual property

IPC— Interprocessor communication. Also referred to as *mailbox.*

IrDA— Infrared Data Association: Represents the group of device manufacturers that developed a standard for transmitting data via infrared light waves.

IRQ— Interrupt request: IRQs are hardware lines over which devices can send interrupt signals to the microprocessor.

ISO— Isochronous: This refers to processes where data must be delivered within certain time constraints. For example, multimedia streams require an isochronous transport mechanism to ensure that data is delivered as fast as it is displayed and to ensure that the audio is synchronized with the video. Also, *International Standards Organization.*

ISOCH — Isochronous transfer type.

ISP— Image sensing product; also image signal processor

ISR— Interrupt service routine: A function or set of functions that are called when an interrupt is encountered.

ITP — Isochronous Timestamp Packet: USB SS micro-frame boundary packets.

ITU— International Telecommunications Union

IVA— Image and video accelerator

J

J — Logical USB2 line level: Diff1 in HS/FS, Diff0 in LS. Idle FS/LS bus state.

JPEG— Joint Photographics Experts Group

JTAG— Joint Test Action Group: The JTAG was formed in 1985 to develop economical test methodologies for systems designed around complex integrated circuits and assembled with surface-mount technologies. The group drafted a standard that was subsequently adopted by IEEE as IEEE Standard 1149.1-1990, IEEE Standard Test Access Port, and Boundary-Scan Architecture.

K

K — Logical USB2 line level: Diff0 in HS/FS, Diff1 in LS. Resume bus state.

Kb— Kilobits

KB— Kilobyte, 1024 B

Kbps— Kilobits per second

L

L1— Level 1 cache/memory

L2— Level 2 cache/memory

L3— First level of interconnect in OMAP platform

L4— Second level of interconnect in OMAP platform

LCD— Liquid crystal display: A display that uses two sheets of polarizing material with a liquid crystal solution between them.

LCh— Logical DMA channel

LDM— Load multiple

LDO— Low dropout

LFPS —Low-Frequency Periodic Signal.

LH— Local host

LINK— Link layer device

LLP— Low-level protocol

LMP —Link Management Packet.

LP— Low power, operation mode for PHY

LPDDR— Low-power, double-data rate (SDRAM)

LPM— Low-power mode

LPP— Lines per panel

LPSDR— Low-power, single-data rate (SDRAM)

LRC— Longitudinal redundancy check

LRU— Least recently used

LS— Level shifter; also low speed or Low-Speed USB data rate (1.5 Mbps).

LSB— Least-significant bit

LUT— Look-up table

LVDS— Low-voltage differential signaling

M

Mailbox— See *IPC*

Mb— Megabit

Mbps— Megabits per second

McBSP— Multichannel buffered serial port: An enhanced buffered serial port that includes the following standard features: buffered data registers, full duplex communication, and independent clocking and framing for receive and transmit. In addition, the McBSP includes the following enhanced features: internal programmable clock and frame generation, multichannel mode, and general-purpose I/O.

MCSPI— Multichannel serial port interface

MCP— Multi-chip package

MCU— Microcontroller unit (refers to the MPU)

MDDR— Mobile double-data-rate SDRAM, dedicated to mobile applications

MHz —Megahertz.

MIPI®— Mobile industry processor interface

MMC— Multimedia card

MMC/SD— Multimedia card/secure data

MMR —Memory Mapped Register.

MMU— Memory management unit: The MMU performs virtual-to-physical address translations, performs access permission checks for access to the system memory, and provides the flexibility and protection required for the OS to manage a shared physical memory space between the two processors.

MPEG— Motion Pictures Expert Group: A compression scheme for full-motion video.

MPEG1— The first MPEG compression scheme specification

MPEG4— The most current MPEG compression scheme specification, intended for very narrow bandwidths.

MPU— Microprocessor unit

MS— Mobile station

MSB— Most-significant bit: The highest order bit in a word. The plural form (MSBs) refers to a specified number of high-order bits, beginning with the highest order bit and counting to the right. For example, the 8 MSBs of a 16-bit value are bits 15 through 8.

MSC —Mass Storage Class.

MUX— Multiplex/multiplexer

Muxed pin— A pin is muxed when its pin control register field can be reconfigured by software to change the function associated with the pin.

MuxMode— A 3-bit field of the pin control register field which enables to change the mode. Mode programming is assumed by software and selects a function on the device external interface.

N

N/A— Not applicable

NAK— Not acknowledged

NAND— NAND flash memory that is a high-capacity, low-cost, embedded, permanent data storage solution

NF— Noise filter

NMI— Nonmaskable interrupt: An interrupt that cannot be masked or disabled.

NOP— No operation (DSP/CPU instruction)

NR— Noise reduction

NSC— National Semiconductor Corporation

NTSC— National Television System Committee (television broadcast system)

O

OCM— On-chip memory

OCO— One change only

OCP— Open-core protocol

OCPI— Open-core protocol interface

OEM— Original equipment manufacturer

OGL— Open GL (programming API) enable

OHCI— Open host controller interface: This is an industry standard USB host controller interface.

OMAP— An open software and hardware platform targeted at second- and third-generation cellular phones with multimedia capabilities.

OneNand™ — A memory chip based on NAND architecture integrating SRAM buffers and NOR logic interface. It combines the advanced data storage function of NAND flash with fast read speed function of NOR flash.

OTG— On-the-go (USB 2.0 specification)

P

PBIAS— PMOS bias transistor to provide the bias voltage to extended drain I/Os

PCB— Printed circuit board

PCLK— Pixel clock

PCM— Pulse code modulation: A technique for digitizing speech by sampling the sound waves and converting each sample into a binary number.

PDA— Personal digital assistant

PDE— Personal down enable

PE— Packet end

PF— Packet footer

PH— Packet header

PG— Protection group

PGA— Pin grid array

PHY— Physical layer device

PI— Packet identifier

PID— Protocol identifier: The PID register is used in Windows CE mode only.

PIPE —PHY Interface for PCI Express, also used in USB3 SS PHY interface.

PIPE3 —PIPE Interface for USB3 SS PHY interface.

PLD— Programmable logic device

PLL— Phase-locked loop: A closed loop frequency control system whose function is based on the phase-sensitive detection of the phase difference between the input signal and the output signal of the controlled oscillator (CO).

PMC— Program memory controller

POP— Package-on-package technology

PMP— Power management port

POR— Power-on reset

PPI— Physical layer protocol interface

PPS— Protocol and parameter selection

PRCM— Power, reset, and clock management

PRM— Power and reset manager

PS— Packet start

PSA— Parallel signature analyzer

PT— Packet type

PVT— Process, voltage, and temperature (that is, PVT dispersion)

PWB— Printed wiring board

PWM— Pulse width modulation

PWR— Power

Q

QCIF— Quarter common intermediate format: A video conferencing format that specifies data rates of 30 fps, with each frame containing 144 lines and 176 pixels per line. This is one-fourth the resolution of CIF. QCIF support is required by the ITU H.261 video conferencing standard.

QVGA— Quarter video graphics array (one-fourth the resolution of VGA).

R

R— Read only

RAM —Random Access Memory.

RBL— Read buffer logic

RC— Read-Clear

RC/W— Read-Clear / Write

RC/W1C— Read-Clear / Write 1 to Clear

RC/W1CP— Read-Clear / Write 1 to Clear (Privilege Only)

RC/W1S— Read-Clear / Write 1 to Set

RC/W1SP— Read-Clear / Write 1 to Set (Privilege Only)

RC/WP— Read-Clear / Write (Privilege Only)

RCP— Read-Clear (Requires Privilege)

RCP/W— Read-Clear (Privilege Only) / Write

RCP/W1C— Read-Clear (Privilege Only) / Write 1 to Clear

RCP/W1CP— Read-Clear (Privilege Only) / Write 1 to Clear (Privilege Only)

RCP/W1S— Read-Clear (Privilege Only) / Write 1 to Set

RCP/W1SP— Read-Clear (Privilege Only) / Write 1 to Set (Privilege Only)

RCP/WP— Read-Clear (Privilege Only) / Write (Privilege Only)

RFBI— Remote frame buffer interface

RFU— Reserved for future use

RGB— Red, green, blue

RGBA— Red, green, blue, alpha

ROM— Read only memory: A semiconductor storage element containing permanent data that cannot be changed.

RP— Read (Requires Privilege)

RP/W— Read (Privilege Only) / Write

RP/W1C— Read (Privilege Only) / Write 1 to Clear

RP/W1CP— Read (Privilege Only) / Write 1 to Clear (Privilege Only)

RP/W1S— Read (Privilege Only) / Write 1 to Set

RP/W1SP— Read (Privilege Only) / Write 1 to Set (Privilege Only)

RP/WP— Read (Privilege Only) / Write (Privilege Only)

RST— Reset

RT— Real-time

RVLC— Reversible variable length coder

RVLD— Reversible variable length decoder

R/W— Read / Write

R/W1C— Read / Write 1 to Clear

R/W1CP— Read / Write 1 to Clear (Privilege Only)

R/W1S— Read / Write 1 to Set

R/W1SP— Read / Write 1 to Set (Privilege Only)

R/WP— Read / Write (Privilege Only)

RX— Receive/receiver

S

S3220— Abbreviation of Sonics3220. It refers to the L4 interconnect.

SAM— Shared access mode: The mode that allows both the DSP and the host to access host port interface (HPI) memory. In this mode, asynchronous host accesses are synchronized internally, and, in case of conflict, the host has access priority and the DSP waits one cycle.

SAR— Save and restore: Hardware context saving for power saving

SAV— Start of active video

SCCB— Serial camera control interface: 3-wire and 2-wire serial bus defined and deployed by Omnivision Technologies, Inc.

SCL— Serial clock: Programmable serial clock used in the I^{2}C interface. Also SCLK.

SCM— Scan combiner module; also statistic collection module

SCP— Serial configuration port

SD— Starting delimiter

SDA— Serial data: Serial data bus in the I^{2}C interface.

SDI— Serial display interface

SDIO— Secure digital input/output

sDMA— System direct memory access

SDR— Single data rate

SDRC— SDRAM controller

SDRAM— Synchronous dynamic random access memory

SDTI— System debug trace interface

SDTV— Standard digital television

SE — Single-Ended: USB state based on individual lines’ state (D+/D-).

SE0 — Single-Ended zero line state where D+=D-=0. Used for reset, FS/LS EOP.

SER — Soft Error Rate.

SGX— Abbreviation for graphic accelerator (GFX in ES1.0)

SIM— Subscriber identity module

SIMD— Single instruction multiple data

SILVS— Scalable low-voltage signaling

SMS— SDRAM memory scheduler

SMX— Abbreviation of SonicsMX. It refers to the L3 interconnect.

SNR— Signal-to-noise ratio

SOC — System On a Chip.

SOF— Start of frame

SOT— Start of transmission

SRAM— Static random access memory. Fast memory that does not require refreshing, as DRAM does. It is more expensive than DRAM, though, and is not available in as high a density as DRAM.

SRG— Sample rate generator

SRP — Session Request Protocol. OTG extension to wake-up a system, allowing a B-device to request an A-device to turn on the VBUS power and start session.

SRRP— Session RAM restoration pointer

SS— Subsystem or Super Speed

SSC — Spread Spectrum Clocking.

SSR— Serial synchronous receiver

SST— Serial synchronous transmitter

ST— Start timer

STM— Synchronous transfer mode; also store multiple

STN— Super-Twist Nematic: A technique for improving LCD display screens by twisting light rays.

SVGA— Super video graphic adapter

SW— Software

SWI— Software interrupt

T

TC— Traffic controller. Allows asynchronous operation among the external memory interface, the MPU, and the DSP.

TCK— Test clock

TD— Transfer descriptor

TDM— Time division multiplex/multiplexing: The process by which a single serial bus is shared by multiple devices with each device taking turns to communicate on the bus. The total number of time slots (channels) depends on the number of devices connected. During a time slot, a given device may talk to any combination of devices on the bus.

TFT— Thin film transistor. A type of LCD flat panel display screen in which each pixel is controlled by one to four transistors.

TLB— Translation lookaside buffer: A cache that contains entries for virtual-to-physical address translation and access permission checking.

TLL— Transceiverless link: This is logic that lets the user connect two USB transceiver interfaces together directly without the use of differential transceivers.

TM— Target module: A target module cannot generate read/write requests to the chip interconnects, but it responds to these requests. However, it may generate interrupts or a DMA request to the system (typically: peripherals, memory controllers).

TOC— Table of contents

TRB —Transfer Request Block.

TRM— Technical reference manual

TRX— USB transceiver: The USB analog driver/receiver.

TTB— Translation table base: It points to the base of a table in physical memory that contains section and page table descriptors.

TWL— Table walking logic

TX— Transmit/transmitter

U

UART— Universal asynchronous receiver/transmitter: Another name for the asynchronous serial port.

UAS —USB-Attached SCSI: ANSI standard for USB-attached storage.

UICC— Universal integrated circuit card

ULPM— Ultra-low power mode

ULPI— UTMI+ low pincount interface

ULPS— Ultra-low power state

UMC— United memory controller

UNP— Unpredictable

US — Upstream facing from a device (or hub) to the host (or a hub).

USAR— Universal synchronous/asynchronous receiver

USART— Universal synchronous/asynchronous receiver/transmitter

USB— Universal serial bus: An external bus standard that supports data transfer rates of 12M bps (12 million bits per second). A single USB port can be used to connect up to 127 peripheral devices.

USB IF —USB Implementers Forum. The governing organization that develops and maintains the USB specifications and compliance standard. www.usb.org

UTMI —USB 2.0 PHY Transceiver Macrocell Interface specification.

UTMI+ —UTMI plus extensions to the UTMI spec. Among other improvements, it defines the PHY interface for OTG 2.0.

V

VA— Volt-amps: A form of power management. A VA rating is the volts rating multiplied by the amps (current) rating, used to indicate the output capacity of an uninterruptible power supply (UPS) or other power source.

VC— Virtual channel

VENC— Video encoder

VESA— Video Electronics Standards Association

VFP— Vertical front porch

VGA— Video graphics array: An industry standard for video cards.

VLC— Variable length decoder

VLCD— Variable length coding and decoding coprocessor

VLD— Variable length coder

VMODE— Bi-level voltage control interface

VPBE— Video processing back end

VSEC— Vertical synchronization end code

VSSC— Vertical synchronization start code

VSYNC, VS— Vertical synchronization: A bidirectional vertical timing signal occurring once per frame with a pulse-width defined as an integral number of lines (half-lines for interlaced mode).

W

W— Write

W1C— Write 1 to Clear

W1CP— Write 1 to Clear (Requires Privilege)

W1S— Write 1 to Set

W1SP— Write 1 to Set (Requires Privilege)

WBL— Write buffer logic

WC— Word count

WD— Watchdog: A timer that requires the user program or OS periodically write to the count register before the counter underflows.

WDT— Watchdog timer

WLAN— Wireless local area network

WMV— Windows media video

WMA— Windows media audio

Word16— 16-bit word

Word32— 32-bit word

WP— Write (Requires Privilege)

WSS— Wide-screen signaling

WWT— Work waiting time

X

XGA, XVGA— Extended graphics array

xHC — Host Controller, designates the USB HW that implements the xHCI specification.

xHCI — eXtensible Host Controller Interface. The specification that defines the register level interface for host controller.

XIP— Execution in place

Y

YUV— Luminance-Bandwidth-Chrominance

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265