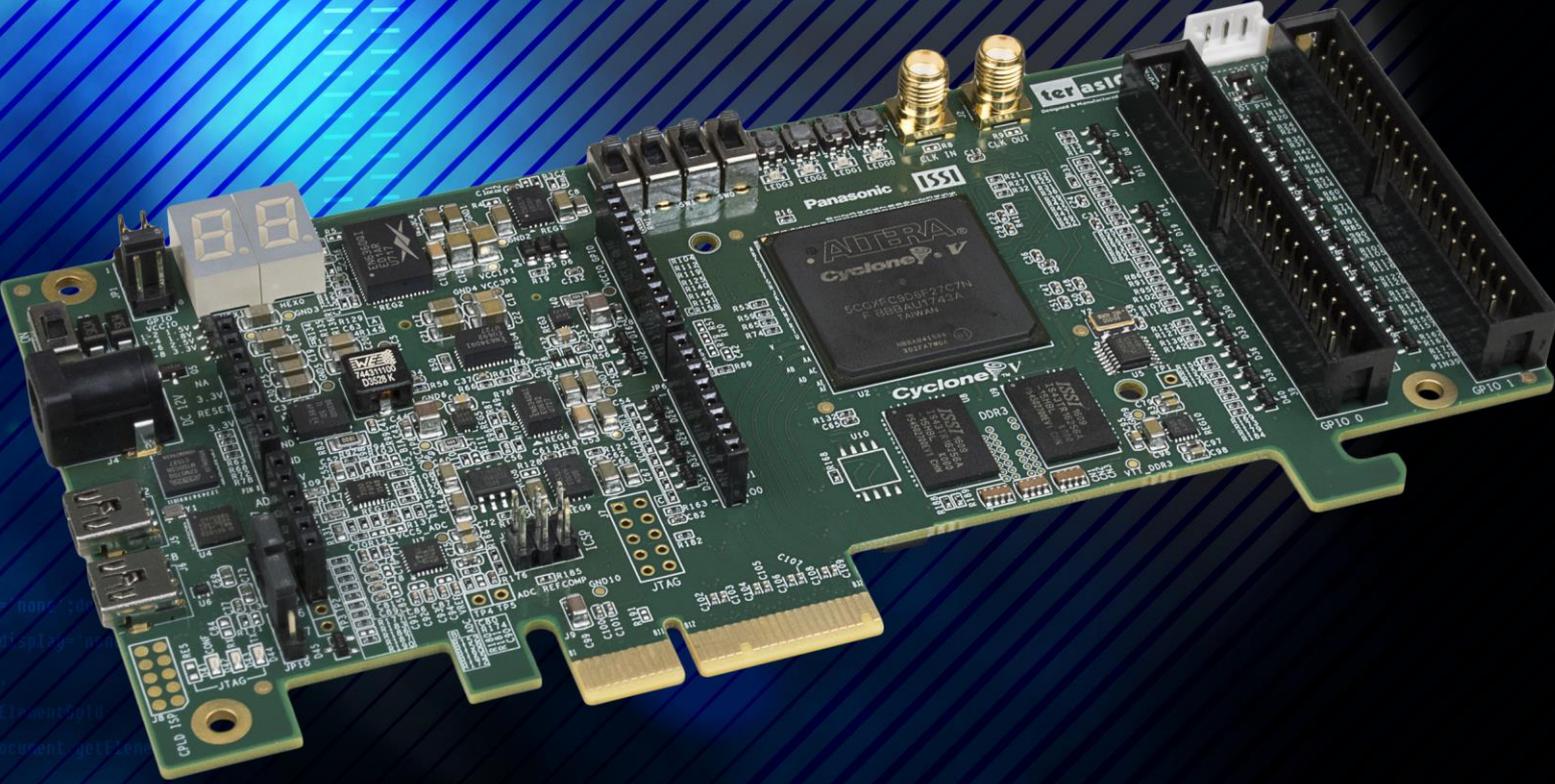


OpenCL

OpenVINO

Starter Kit



terasic
www.terasic.com

CHAPTER 1	<i>OSK OPENCL</i>	3
1.1	SYSTEM REQUIREMENT	4
1.2	OPENCL ARCHITECTURE	5
CHAPTER 2	<i>OPENCL FOR WINDOWS</i>	6
2.1	SOFTWARE INSTALLATION	6
2.2	ENVIRONMENT CONFIGURE	9
2.3	OPENCL ENVIRONMENT VERIFY	12
2.4	INITIALIZING THE FPGA FOR USING WITH OPENCL	14
2.5	OPENCL RUNTIME VERIFY	17
2.6	COMPILE AND TEST OPENCL PROJECT.....	19
CHAPTER 3	<i>OPENCL FOR LINUX</i>	23
3.1	SOFTWARE INSTALLATION	23
3.2	ENVIRONMENT CONFIGURE	27
3.3	OPENCL ENVIRONMENT VERIFY	28
3.4	INITIALIZING THE FPGA FOR USE WITH OPENCL.....	29
3.5	OPENCL RUNTIME VERIFY	32
3.6	COMPILE AND TEST OPENCL PROJECT.....	34
CHAPTER 4	<i>APPENDIX</i>	37
	REVISION HISTORY	47
	COPYRIGHT STATEMENT	47

Chapter 1

OSK OpenCL

OSK (OpenVINO Starter Kit), an unparalleled and powerful platform for high-speed computation, is now an Intel officially certified board for Intel’s Preferred Board Partner Program for OpenCL. It supports both 64-bit Windows and Linux. This document will introduce how to setup OpenCL development environment, and how to compile and execute the example projects for OSK. Note that OpenCL coding instruction is not in the scope of this document, but the user can refer to Intel FPGA SDK for OpenCL Programming Guide for more details.

https://www.altera.com/en_US/pdfs/literature/hb/opencl-sdk/aocl_programming_guide.pdf

This OpenCL BSP and manual can support two devices development (Cyclone V GX or Cyclone V GT device on OpenVINO Starter Kit). In the BSP, we use different board names to distinguish the boards, the correspondence of the *<board name>* and the FPGA Device is as follows:

<i><board name></i>	FPGA Device	PCIe Support
c5gx	5CGXFC9D6F27C7	Gen 1x4
c5gt	5CGTFD9D5F27C7	Gen 2x4

1.1 System Requirement

The following items are required to set up OpenCL for OSK board:

- OpenVINO Starter Kit
- A Host PC with
 - USB Host Port
 - One PCI Express x4/x8/x16 slot
 - 16GB memory is recommended, 8GB is minimal
 - 12V Power for OSK
- An USB Cable(type A to mini-B)
- 64-bit Windows 7/10 or Linux (Redhat 6.5/CentOS 7.0/Ubuntu14.04) Installed
- Quartus Prime Standard Edition 17.1 Installed, **license is required**
- Intel FPGA SDK for OpenCL 17.1 Installed, **license is not required**
- OSK OpenCL BSP 17.1 Installed
- Visual Studio 2012 C/C++ installed for Windows7/10
- GNU development tools for Linux

Note: Intel FPGA OpenCL only supports 64-bit OS and x86 architecture.

1.2 OpenCL Architecture

An OpenCL project is composed of both OpenCL Kernel and Host Program as shown in **Figure 1-1**. OpenCL kernel is compiled with Intel FPGA OpenCL compiler provided by the Intel FPGA OpenCL SDK. The Host Program is compiled by Visual Studio C/C++ on Windows or GCC on Linux.

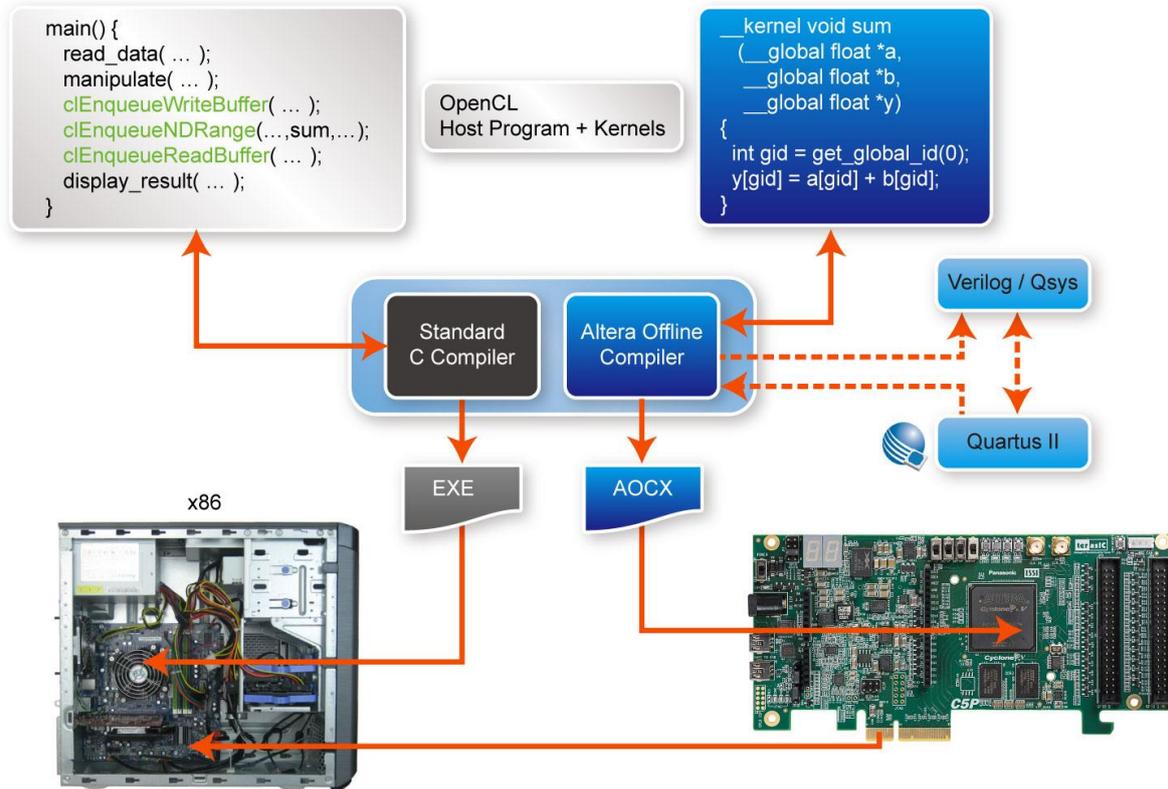


Figure 1-1 Intel FPGA OpenCL Architecture

Chapter 2

OpenCL for Windows

This chapter describes how to set up OSK OpenCL development environment on 64-bit Windows, and how to compile and test the OpenCL examples. For more details about Intel FPGA OpenCL started guide, please refer to:

https://www.altera.com/en_US/pdfs/literature/hb/opencl-sdk/aocl_getting_started.pdf

2.1 Software Installation

This section describes where to get the required software for OpenCL.

■ **Quartus Prime and OpenCL SDK**

Quartus Prime Standard Edition 17.1 and Intel FPGA SDK for OpenCL 17.1 can be download from the web site:

<http://dl.altera.com/opencl/17.1/?edition=standard>

For Quartus Prime installation, please make sure that the Cyclone V device is included.

Open the link and select the **Windows SDK** table as **Figure 2-1** shows.

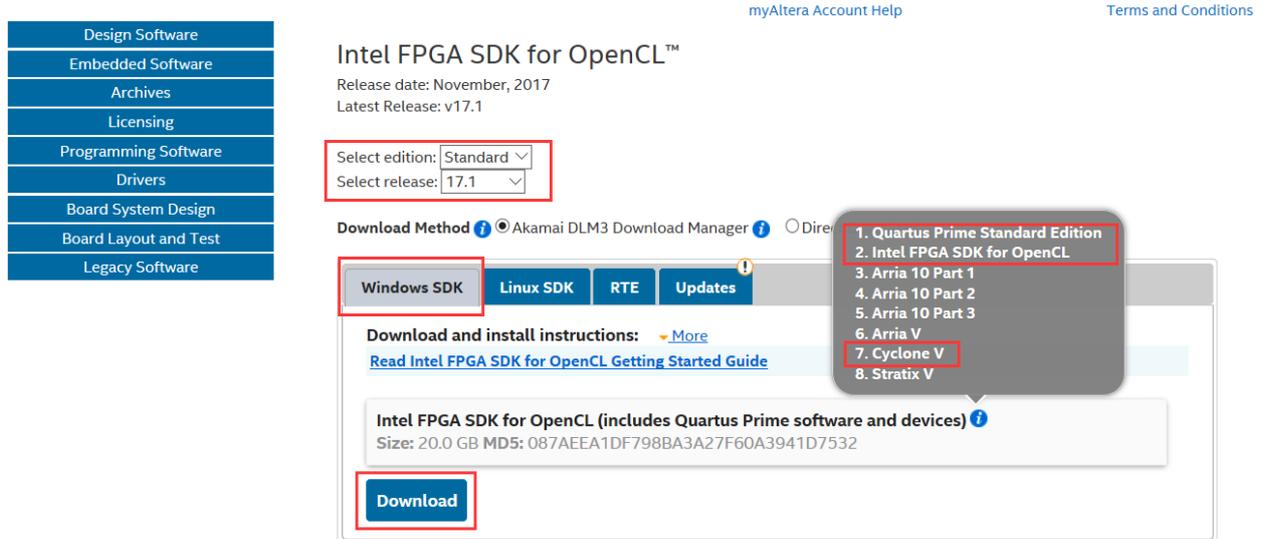


Figure 2-1 OpenCL Windows SDK Files

■ Visual Studio 2012

If developers don't have Visual Studio C/C++ 2012, they can use the trial version of Visual Studio 2012 Express. The software can be downloaded from the web site:

<https://www.visualstudio.com/vs/older-downloads/>

■ OSK OpenCL BSP (Board Support Package)

After Quartus Prime and OpenCL SDK are installed, download the windows BSP for Intel FPGA OpenCL 17.1(OSK_OpenCL_BSP_17.1.zip) from the web:

<http://osk.terasic.com/cd>

Then, decompress OSK_OpenCL_BSP_17.1.zip to the "osk" folder under the folder "C:\intelFPGA\17.1\hld\board", as shown in **Figure 2-2**, where is assumed Quartus Prime is installed on the folder "C:\intelFPGA\17.1".

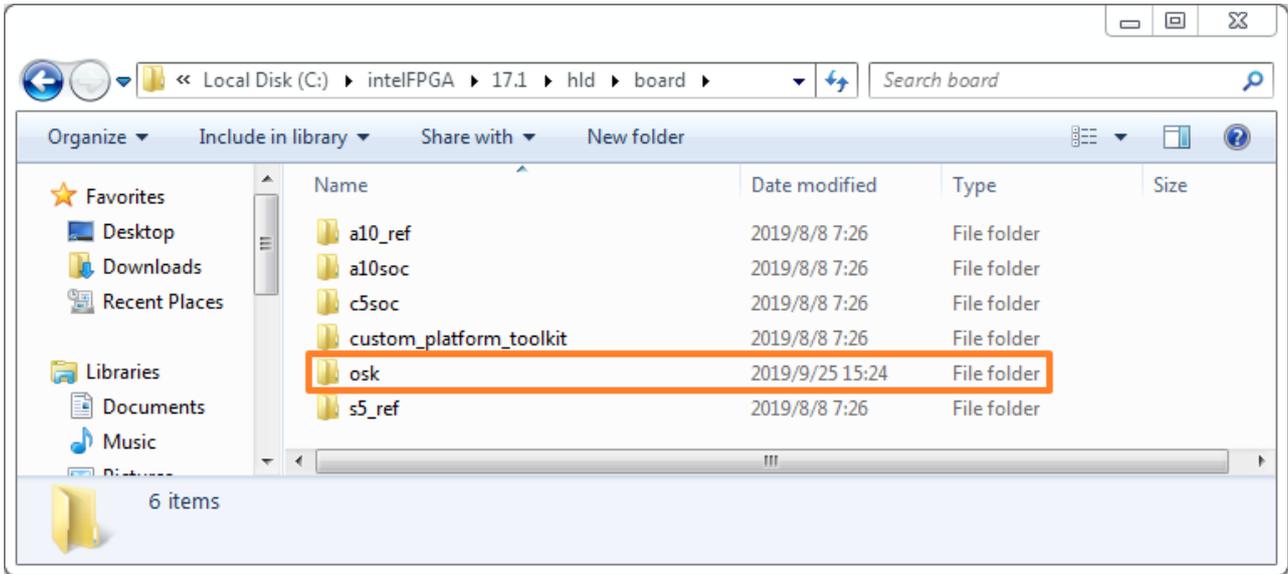


Figure 2-2 OSK OpenCL BSP Content

For more details about OSK OpenCL BSP, please refer to the [Table 1](#).

Table 1 Windows BSP File

File or Folder	Description
board_env.xml	eXtensible Markup Language (XML) file that describes the Reference Platform to the Intel FPGA SDK for OpenCL.
hardware	Contains the Intel Quartus Prime project templates for the OSK board variant.
windows64	Contains the MMD library, kernel mode driver, and executable files of the SDK utilities (that is, install, uninstall, flash, program, diagnose) for your 64-bit operating system
tests	Contains some OpenCL Design Examples. The following examples demonstrate how to describe various applications in OpenCL along with their respective host applications, which you can compile and execute on a host with an FPGA board that supports the Intel FPGA SDK for OpenCL.
bringup	The demo batch files of initializing the OSK or OpenCL User

2.2 Environment Configure

Developers need to create and edit some environment variable that Intel FPGA OpenCL SDK can find the kit location of OSK correctly

Now, here are the procedures to create the required environment variable on Windows 7:

1. Open the **Start** menu and right click on **Computer**. Select **Properties**.
2. Select **Advanced system settings**.
3. In the **Advanced** tab, select **Environment Variables**.
4. Select **New**.
5. In the popup dialog, edit **New User Variable**, type the name in the **Variable name** edit box and type the value in the **Variable value** edit box.

First, edit the environment variable name **ALTERAOCLSDKROOT** to **INTELFPGAOCLSDKROOT**, as shown in [Figure 2-3](#).

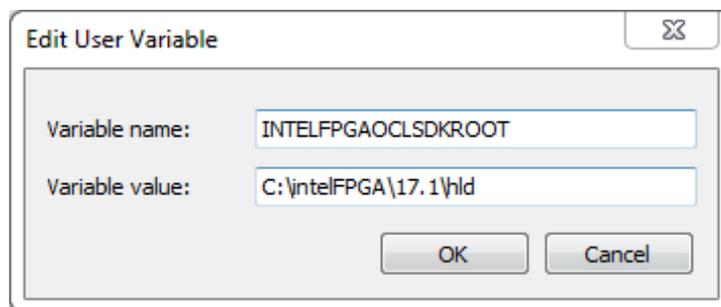


Figure 2-3 Edit ALTERAOCLSDKROOT Environment Variable

Then, create an environment variable **AOCL_BOARD_PACKAGE_ROOT**, and set its value as:

`"%INTELFPGAOCLSDKROOT%\board\osk"`

as shown in [Figure 2-4](#).

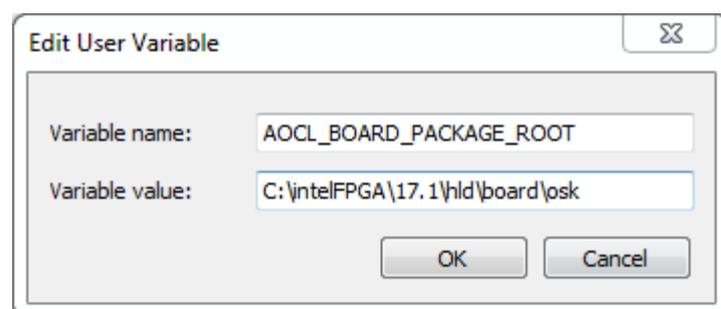


Figure 2-4 Setup AOCL_BOARD_PACKAGE_ROOT Environment Variable

Then, create an environment variable **CL_CONTEXT_COMPILER_MODE_INTELFPGA**, and set its value as "3", as shown in [Figure 2-5](#).

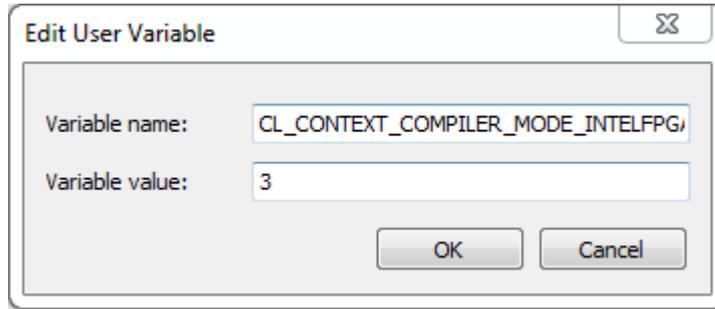


Figure 2-5 Setup CL_CONTEXT_COMPILER_MODE_INTELFPGA Environment Variable

Also, append

"%QUARTUS_ROOTDIR%\bin64" and
 "%INTELFPGAOCLSDKROOT%\bin" and
 "%INTELFPGAOCLSDKROOT%\windows64\bin" and
 "%AOCL_BOARD_PACKAGE_ROOT%\windows64\bin"

into the **PATH** environment variable so the OpenCL SDK can find the binary file provided by OSK BSP as shown in [Figure 2-6](#) and [Figure 2-7](#).

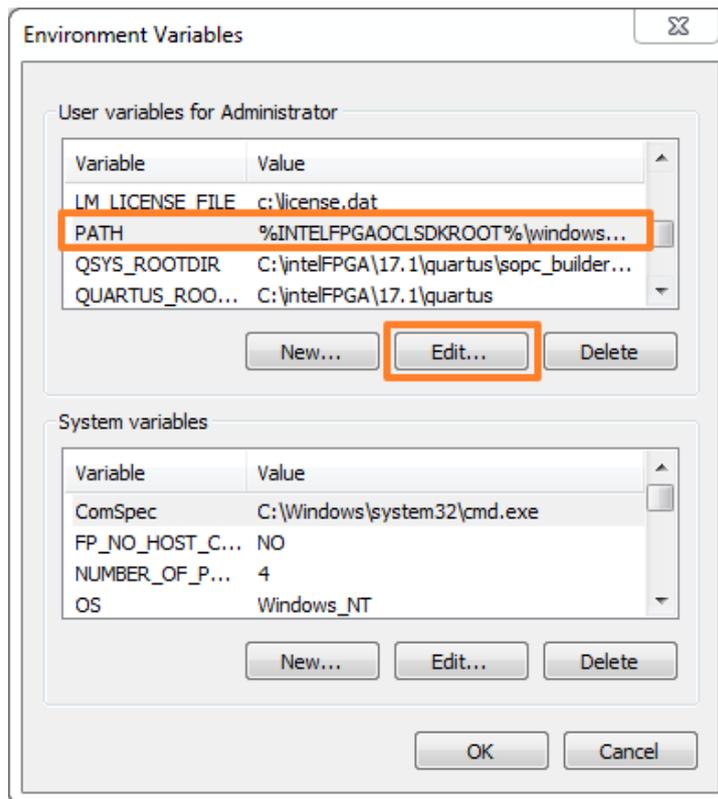


Figure 2-6 Select “PATH” and click “Edit”

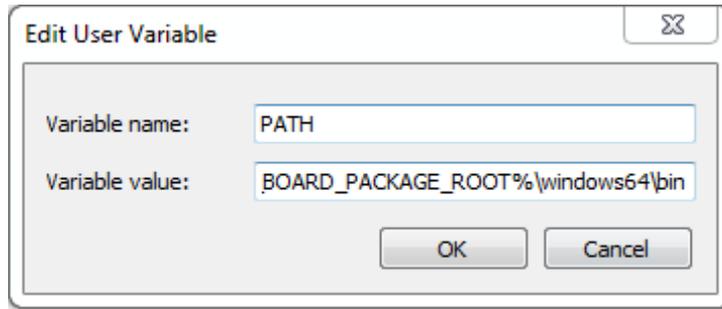


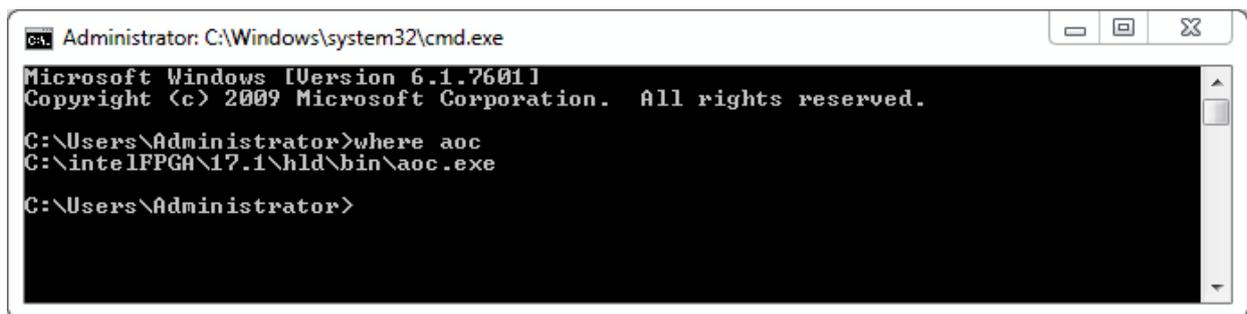
Figure 2-7 Edit PATH environment variable

2.3 OpenCL Environment Verify

This section will show how to make sure the OpenCL environment is setup correctly. Firstly, please open **Command Prompt** windows by clicking Windows **Start** button, clicking **All Programs**, clicking **Accessories**, and then click **Command Prompt**.

■ Target AOCL

In **Command Prompt** window, type “**where aoc**” command, and make sure the path of the “**aoc.exe**” is listed as shown in **Figure 2-8**.



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

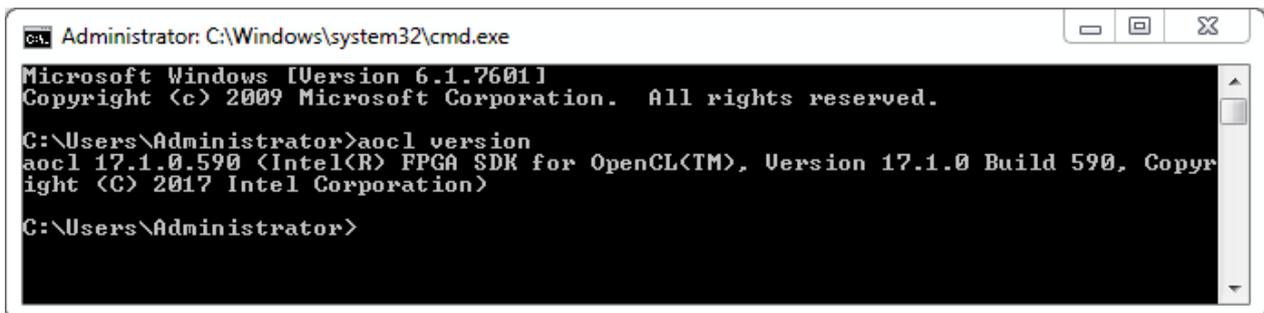
C:\Users\Administrator>where aoc
C:\intelFPGA\17.1\hld\bin\aoc.exe

C:\Users\Administrator>
```

Figure 2-8 Execute “where aoc” command

■ Target SDK Version

In **Command Prompt** window, type “**aocl version**” command, and make sure the version 17.1.0 Build 590 of the OpenCL SDK is listed as shown in **Figure 2-9**.



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

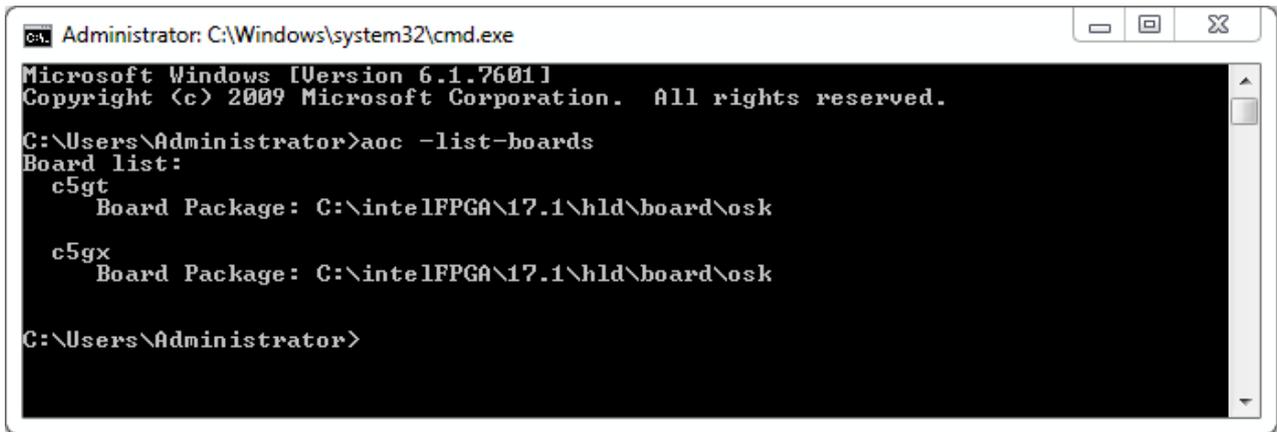
C:\Users\Administrator>aocl version
aocl 17.1.0.590 (Intel(R) FPGA SDK for OpenCL(TM), Version 17.1.0 Build 590, Copyright (C) 2017 Intel Corporation)

C:\Users\Administrator>
```

Figure 2-9 Version of OpenCL SDK

■ Target Board

In **Command Prompt** window, type “**aoc -list-boards**” command, and make sure “**osk**” is listed in **Board list** as shown in **Figure 2-10**.



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>aoc -list-boards
Board list:
  c5gt
    Board Package: C:\intelFPGA\17.1\hld\board\osk

  c5gx
    Board Package: C:\intelFPGA\17.1\hld\board\osk

C:\Users\Administrator>
```

Figure 2-10 'osk' is listed in Board list

For more information about the **aoc** and **aocl**, refer to the '**aoc -h**' and '**aocl help**' command.

2.4 Initializing the FPGA for using with OpenCL

■ Board Setup

Before testing OpenCL on OSK, please follow the below procedures to set up OSK board on your PC as shown in [Figure 2-11](#).

1. Make sure your PC is powered off.
2. Insert OSK board into PCI Express x4/x8 or x16 slot.
3. Connect 12V power source to the OSK
4. Connect PC's USB port to OSK UB2 port using an USB cable.

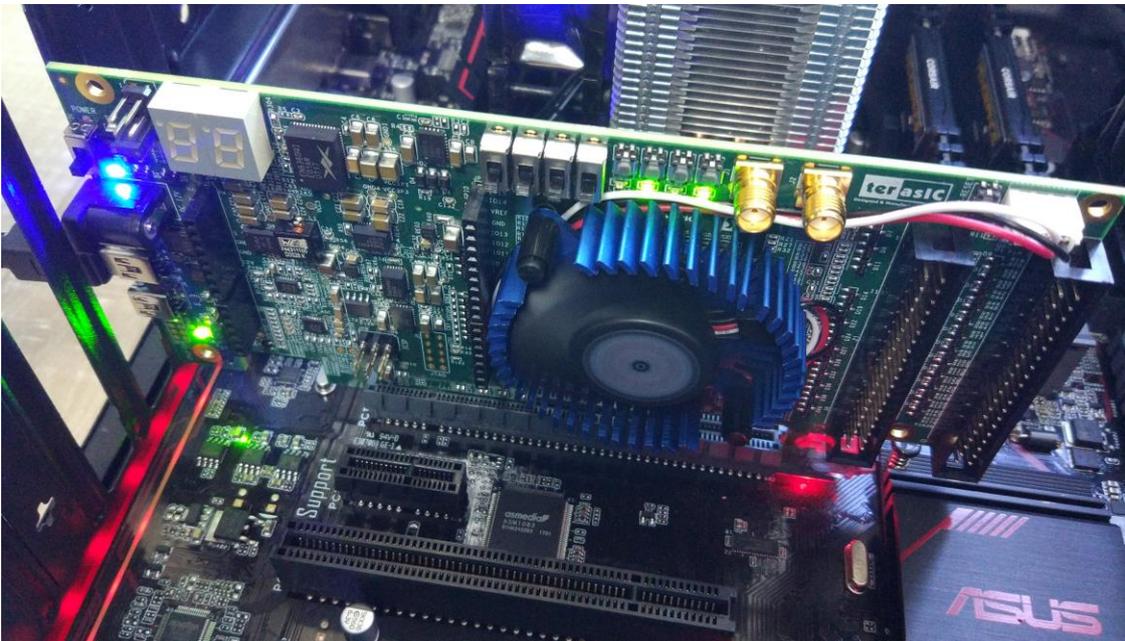


Figure 2-11 Setup OSK board on PC

■ ‘aocl flash’ program the image into FLASH

The **flash** utility in the OSK Development Kit Reference Platform configures the power-on image for the FPGA using the specified .aocx file. Calling into the MMD library implements the flash utility.

In **Command Prompt** window, type “`cd C:\intelFPGA\17.1\hld\board\osk\bringup\<board name>`” to go to bringup folder of the board.

Then type “`aocl flash acl0 hello_world.aocx`” to write **hello_world.aocx** OpenCL image onto the startup configuration flash of OSK. It will take about 5 minutes for flash programming as shown in [Figure 2-12](#).

```

Administrator: C:\Windows\system32\cmd.exe - aocl flash ac10 hello_world.aocx
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd C:\intelFPGA\17.1\hld\board\osk\bringup\c5gt
C:\intelFPGA\17.1\hld\board\osk\bringup\c5gt>aocl flash ac10 hello_world.aocx
aocl flash: Running flash from C:\intelFPGA\17.1\hld\board\osk\windows64\libexec

C:\intelFPGA\17.1\hld\board\osk\bringup\c5gt>"C:/intelFPGA/17.1/quartus"\bin64\perl\bin\perl "C:\intelFPGA\17.1\hld\board\osk"\windows64\libexec\flash.pl ac10 C:\Users\ADMINI~1\AppData\Local\Temp\3172Commandpm820_1569396815_0_fpga_temp.bin
Flash Programming...
Info: *****
Info: Running Quartus Prime Convert_programming_file
Info: Version 17.1.0 Build 590 10/25/2017 SJ Standard Edition
Info: Copyright (C) 2017 Intel Corporation. All rights reserved.
Info: Your use of Intel Corporation's design tools, logic functions
Info: and other software and tools, and its AMPP partner logic
Info: functions, and any output files from any of the foregoing
Info: (including device programming or simulation files), and any
Info: associated documentation or information are expressly subject
Info: to the terms and conditions of the Intel Program License
Info: Subscription Agreement, the Intel Quartus Prime License Agreement,
Info: the Intel FPGA IP License Agreement, or other applicable license
Info: agreement, including, without limitation, that your use is for
Info: the sole purpose of programming logic devices manufactured by
Info: Intel and sold by Intel or its authorized distributors. Please
Info: refer to the applicable agreement for further details.
Info: Processing started: Wed Sep 25 15:34:11 2019

```

Figure 2-12 aocl flash ac10 hello_world.aocx”

After flash programming is done successfully, developers must power off OSK board and PC, then restart the PC.

■ Driver Installation

The `install` utility in the OSK Development Kit is used install the kernel driver on the host computer. Users of the Intel FPGA SDK for OpenCL only need to install the driver once, after that the driver should be automatically loaded each time when the machine reboots.

In **Command Prompt** window, type “`aocl install`” to install the driver as shown in [Figure 2-13](#). Note that users need to have administrator privileges to install the driver.

```

Administrator: C:\Windows\system32\cmd.exe - aocl install
LOG ok: 1, ENTER UpdateDriverForPlugAndPlayDevices...
LOG ok: 0, RETURN UpdateDriverForPlugAndPlayDevices.
LOG ok: 1, Installation was successful.
LOG ok: 0, Install completed
LOG ok: 1, RETURN: DriverPackageInstallW (0x0)
LOG ok: 1, RETURN: DriverPackageInstallA (0x0)
difx_install_preinstall_inf: err 0, last event 0, last error 0. SUCCESS
install: completed successfully

+-----+
+ Installing board drivers... +
+-----+

WDREG utility v11.00. Build Feb 27 2012 14:45:04

Processing HWID PCI\VEN_1172&DEV_D800&SUBSYS_00051172&REV_01
Installing a non-signed driver package for PCI\VEN_1172&DEV_D800&SUBSYS_00051172&REV_01
Device node <hwid:PCI\VEN_1172&DEV_D800&SUBSYS_00051172&REV_01>: exists and is configured. Installing.

```

Figure 2-13 driver installation

For **windows7** x64, If it pops dialog “**Windows Security**” during the installation process, please choose “**Install this driver software anyway**” and go on as shown in **Figure 2-14**.

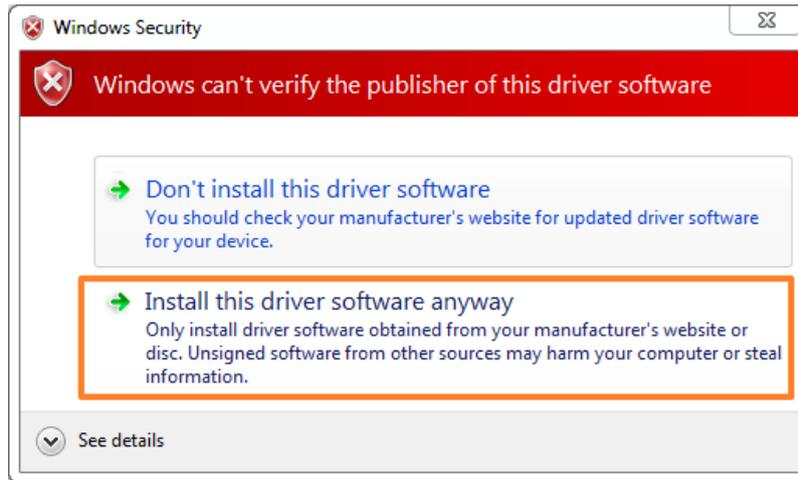


Figure 2-14 windows security

When the installation is successful, **Jungo WinDriver** and **Intel FPGA Accelerator** board can be found in the PC Device Manage as shown in **Figure 2-15**.

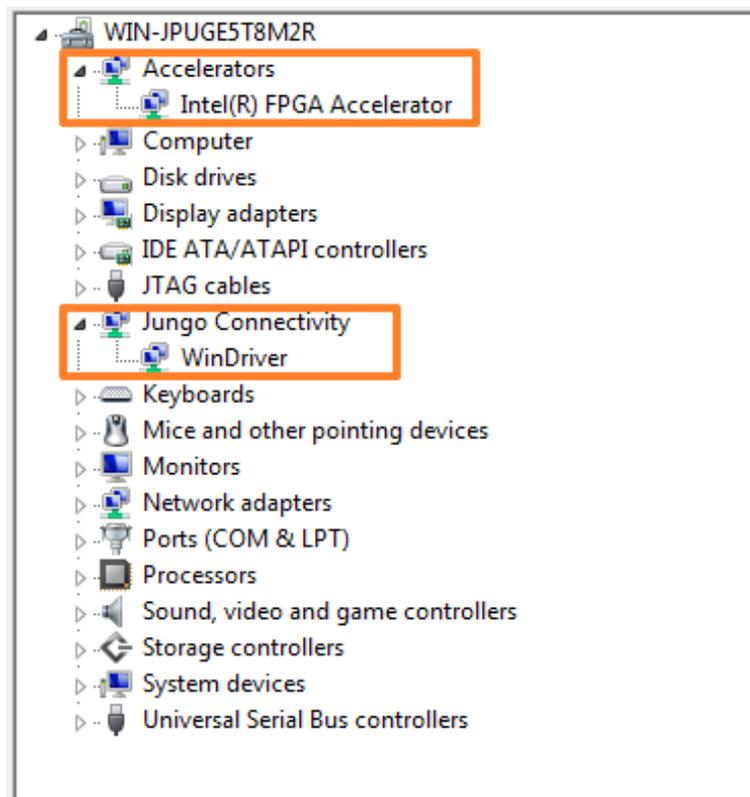


Figure 2-15 driver installation success

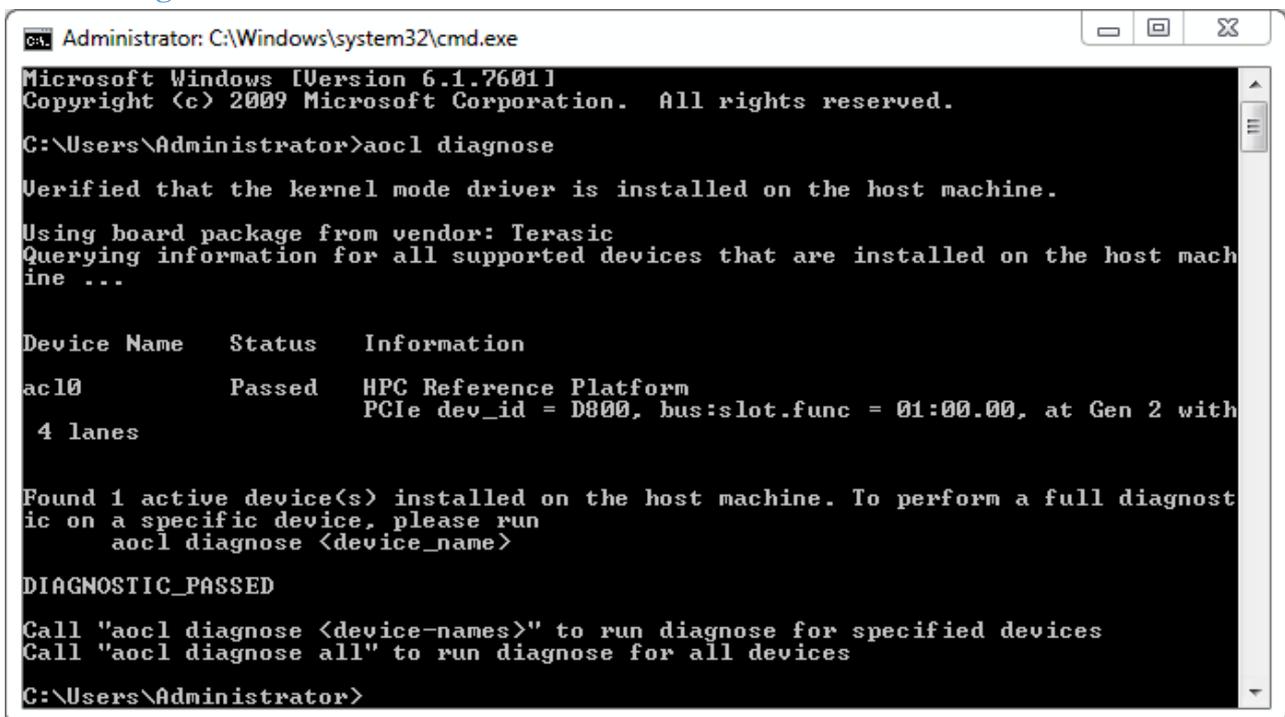
For driver installation on **Windows10** x64 OS, please refer to the **Appendix** of Chapter 4.

2.5 OpenCL Runtime Verify

■ Test ‘aocl diagnose’ Command

The **diagnose** utility in the OSK board reports device information and identifies issues. The diagnose utility first verifies the installation of the kernel driver and returns the overall information of all the devices installed in a host machine.

In **Command Prompt** window, type “**aocl diagnose**” to check if the initialization completed successfully. If successful, the programming message displays “**DIAGNOSTIC_PASSED**” as shown in [Figure 2-16](#).



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>aocl diagnose

Verified that the kernel mode driver is installed on the host machine.

Using board package from vendor: Terasic
Querying information for all supported devices that are installed on the host machine ...

Device Name      Status      Information
-----
acl0             Passed     HPC Reference Platform
                4 lanes   PCIe dev_id = D800, bus:slot.func = 01:00.00, at Gen 2 with

Found 1 active device(s) installed on the host machine. To perform a full diagnostic on a specific device, please run
aocl diagnose <device_name>

DIAGNOSTIC_PASSED

Call "aocl diagnose <device-names>" to run diagnose for specified devices
Call "aocl diagnose all" to run diagnose for all devices

C:\Users\Administrator>
```

Figure 2-16 “aocl diagnose” messages

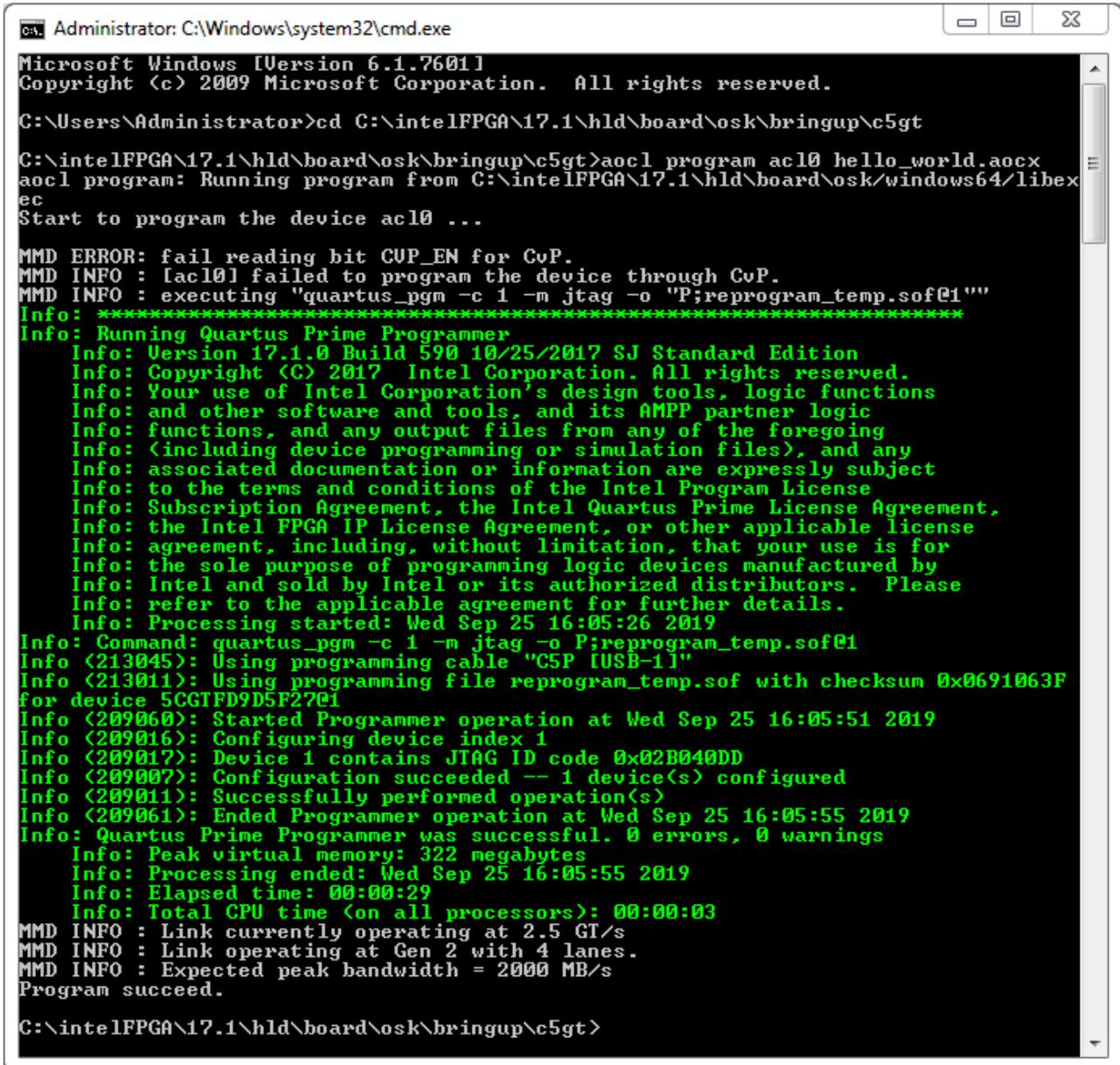
Note: The Cyclone V GT device supports PCIe Gen 2 x4 speed and GX device supports PCIe Gen 1 x4 speed.

■ Test ‘aocl program’ Command

The **program** utility in the OSK Development Kit Reference Platform programs the board with the specified .aocx file use the UB2 port.

Check whether the **hello_world** OpenCL image configures the FPGA successfully. In **Command Prompt** windows, type “**cd C:\intelFPGA\17.1\hld\board\osk\bringup\<board name>**” to go to **bringup** project folder of the board, then type “**aocl program acl0 hello_world.aocx**” to configure

the FPGA with **hello_world.aocx** OpenCL image. If the programming message displays “Program succeed” as shown in **Figure 2-17**, it means the image is programmed into the FPGA correctly.



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd C:\intelFPGA\17.1\hld\board\osk\bringup\c5gt

C:\intelFPGA\17.1\hld\board\osk\bringup\c5gt>aocl program acl0 hello_world.aocx
aocl program: Running program from C:\intelFPGA\17.1\hld\board\osk\windows64\libexec
Start to program the device acl0 ...

MMD ERROR: fail reading bit CUP_EN for CoP.
MMD INFO : [acl0] failed to program the device through CoP.
MMD INFO : executing "quartus_pgm -c 1 -m jtag -o "P;reprogram_temp.sof@1""
Info: *****
Info: Running Quartus Prime Programmer
Info: Version 17.1.0 Build 590 10/25/2017 SJ Standard Edition
Info: Copyright (C) 2017 Intel Corporation. All rights reserved.
Info: Your use of Intel Corporation's design tools, logic functions
Info: and other software and tools, and its AMPP partner logic
Info: functions, and any output files from any of the foregoing
Info: (including device programming or simulation files), and any
Info: associated documentation or information are expressly subject
Info: to the terms and conditions of the Intel Program License
Info: Subscription Agreement, the Intel Quartus Prime License Agreement,
Info: the Intel FPGA IP License Agreement, or other applicable license
Info: agreement, including, without limitation, that your use is for
Info: the sole purpose of programming logic devices manufactured by
Info: Intel and sold by Intel or its authorized distributors. Please
Info: refer to the applicable agreement for further details.
Info: Processing started: Wed Sep 25 16:05:26 2019
Info: Command: quartus_pgm -c 1 -m jtag -o P;reprogram_temp.sof@1
Info <213045>: Using programming cable "C5P USB-1"
Info <213011>: Using programming file reprogram_temp.sof with checksum 0x0691063F
for device 5CGTFD9D5F2701
Info <209060>: Started Programmer operation at Wed Sep 25 16:05:51 2019
Info <209016>: Configuring device index 1
Info <209017>: Device 1 contains JTAG ID code 0x02B040DD
Info <209007>: Configuration succeeded -- 1 device(s) configured
Info <209011>: Successfully performed operation(s)
Info <209061>: Ended Programmer operation at Wed Sep 25 16:05:55 2019
Info: Quartus Prime Programmer was successful. 0 errors, 0 warnings
Info: Peak virtual memory: 322 megabytes
Info: Processing ended: Wed Sep 25 16:05:55 2019
Info: Elapsed time: 00:00:29
Info: Total CPU time (on all processors): 00:00:03
MMD INFO : Link currently operating at 2.5 GT/s
MMD INFO : Link operating at Gen 2 with 4 lanes.
MMD INFO : Expected peak bandwidth = 2000 MB/s
Program succeed.

C:\intelFPGA\17.1\hld\board\osk\bringup\c5gt>
```

Figure 2-17 “aocl program acl0 hello_world.aocx” use UB2

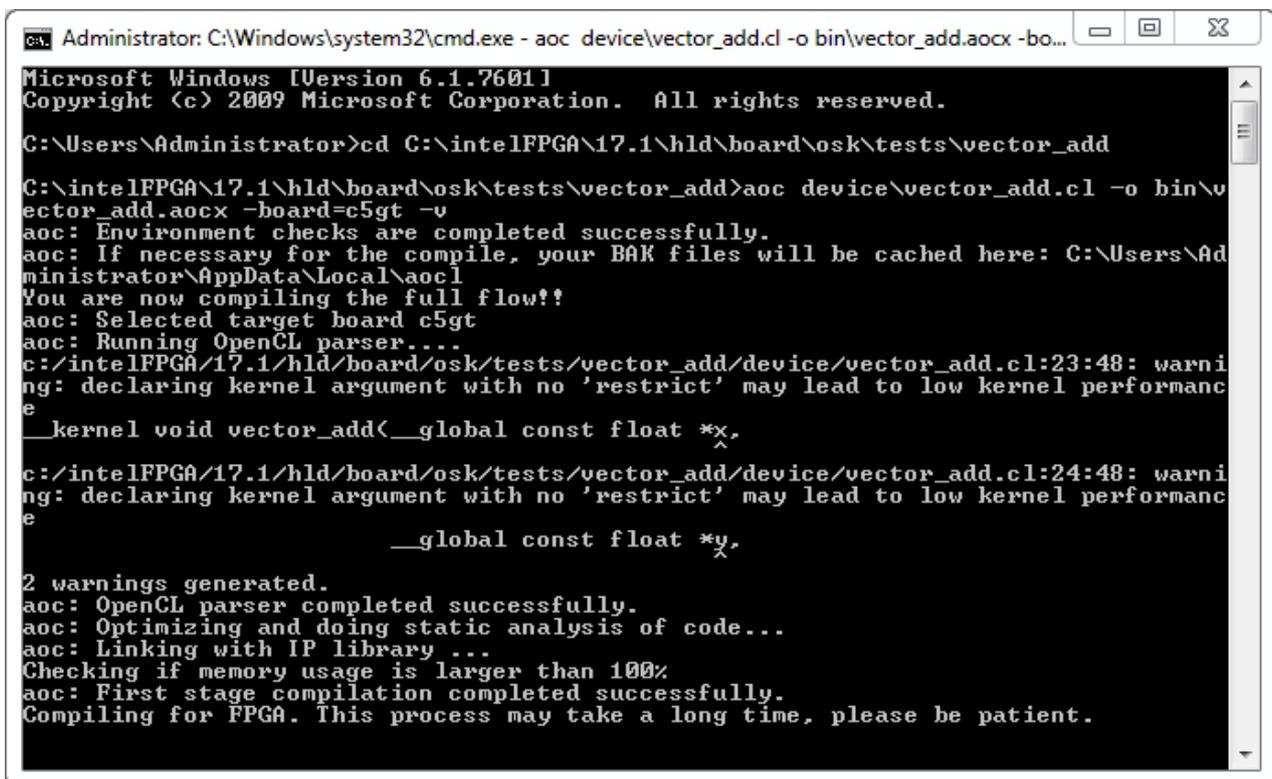
2.6 Compile and Test OpenCL Project

This section will show how to compile and test OpenCL kernel and OpenCL Host Program for the `vector_add` project. Developers can use the same procedures to compile and test other OpenCL examples for OSK.

■ Compile OpenCL Kernel

The utility `aoc` (Intel SDK for OpenCL Kernel Compiler) is used to compile OpenCL kernel. In **Command Prompt** window, type “`cd C:\intelFPGA\17.1\hld\board\osk\tests\vector_add`” to go to `vector_add` project folder, then type “`aoc device\vector_add.cl -o bin\vector_add.aocx -board =<board name> -v`” to compile the OpenCL kernel. It will take about half an hour for compiling. When the compilation process is finished, OpenCL image file `vector_add.aocx` is generated. **Figure 2-18** is the screenshot when OpenCL kernel is compiled successfully. For required parameters to compile `vector_add.cl`, please refer to the `README.txt` that is in the same folder as the `vector_add.cl`. For detailed usage of `aoc`, please refer to the **Intel SDK for OpenCL Programming Guide**:

https://www.altera.com/en_US/pdfs/literature/hb/opencl-sdk/aocl_programming_guide.pdf



```
Administrator: C:\Windows\system32\cmd.exe - aoc device\vector_add.cl -o bin\vector_add.aocx -bo...
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd C:\intelFPGA\17.1\hld\board\osk\tests\vector_add

C:\intelFPGA\17.1\hld\board\osk\tests\vector_add>aoc device\vector_add.cl -o bin\vector_add.aocx -board=c5gt -v
aoc: Environment checks are completed successfully.
aoc: If necessary for the compile, your BAK files will be cached here: C:\Users\Administrator\AppData\Local\aocl
You are now compiling the full flow!!
aoc: Selected target board c5gt
aoc: Running OpenCL parser....
c:/intelFPGA/17.1/hld/board/osk/tests/vector_add/device/vector_add.cl:23:48: warning: declaring kernel argument with no 'restrict' may lead to low kernel performance
    __kernel void vector_add(__global const float *x,
                             ^
c:/intelFPGA/17.1/hld/board/osk/tests/vector_add/device/vector_add.cl:24:48: warning: declaring kernel argument with no 'restrict' may lead to low kernel performance
    __global const float *y,
                             ^
2 warnings generated.
aoc: OpenCL parser completed successfully.
aoc: Optimizing and doing static analysis of code...
aoc: Linking with IP library ...
Checking if memory usage is larger than 100%
aoc: First stage compilation completed successfully.
Compiling for FPGA. This process may take a long time, please be patient.
```

Figure 2-18 “aoc vector_add.cl” OpenCL kernel compile

■ Compile Host Program

Visual Studio C/C++ 2012 is used to compile the Host Program. Launch Visual Studio, and select menu item “FILE→Open Project...”. In the Open Project dialog, go to the folder “C:\intelFPGA\17.1\hld\board\osk\tests\vector_add”, and select “vector_add.sln” as shown **Figure 2-19**.

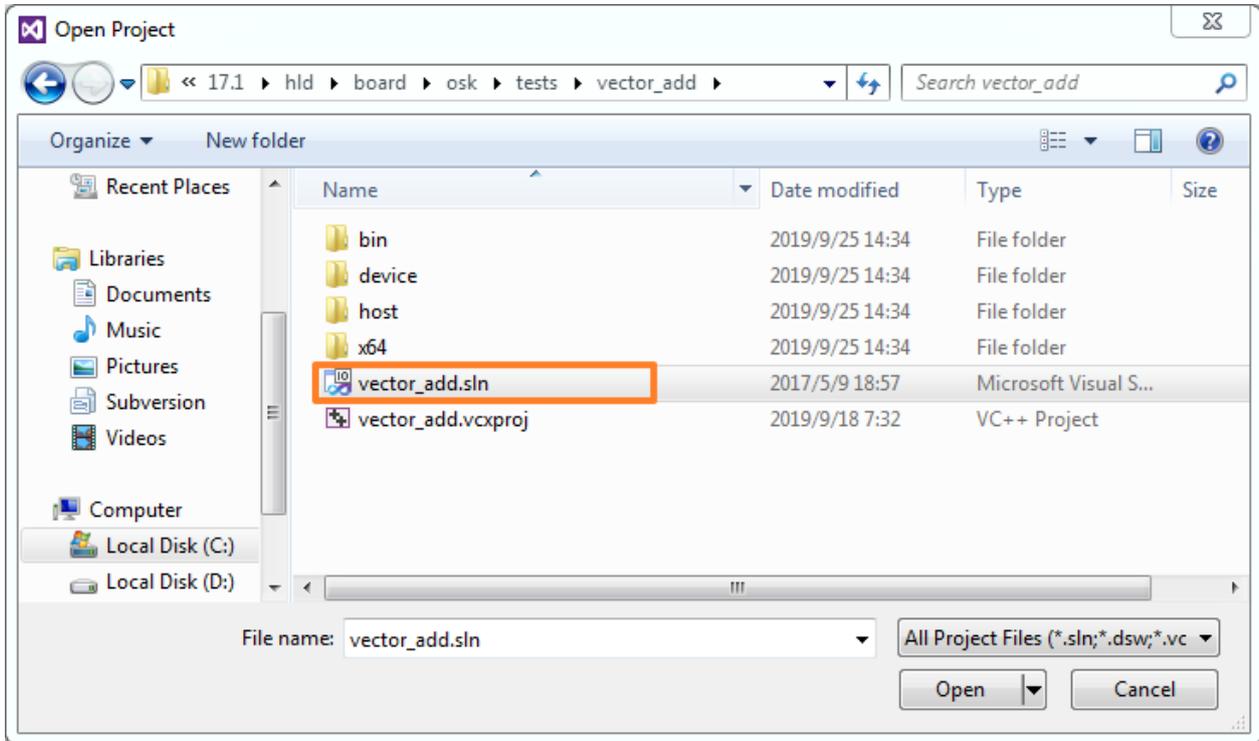


Figure 2-19 Open vector_add.sln Host Program

After vector_add Host Program project is opened successfully, in Visual Studio IDE select menu item “BUILD→Build Solution” to build host program. When build is successfully, you will see successful message as show in **Figure 2-20**. The execute file is generated in:

“C:\intelFPGA\17.1\hld\board\osk\tests\vector_add\bin\host.exe”

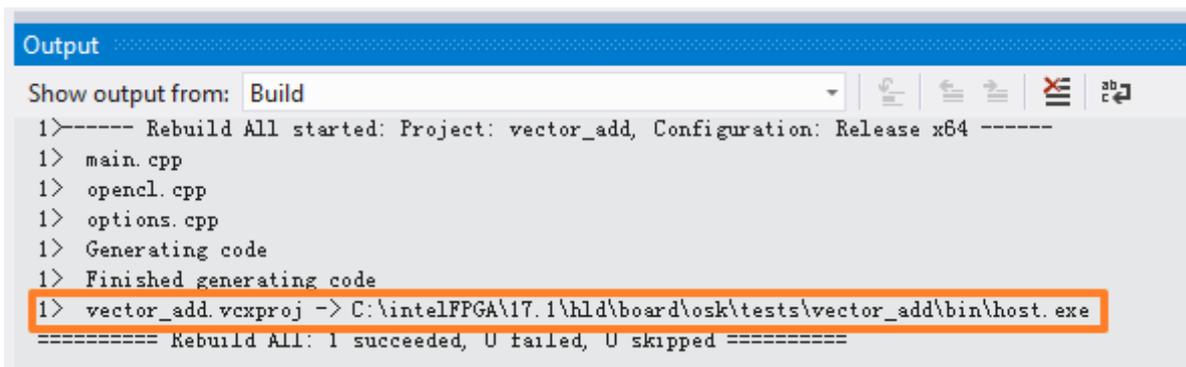
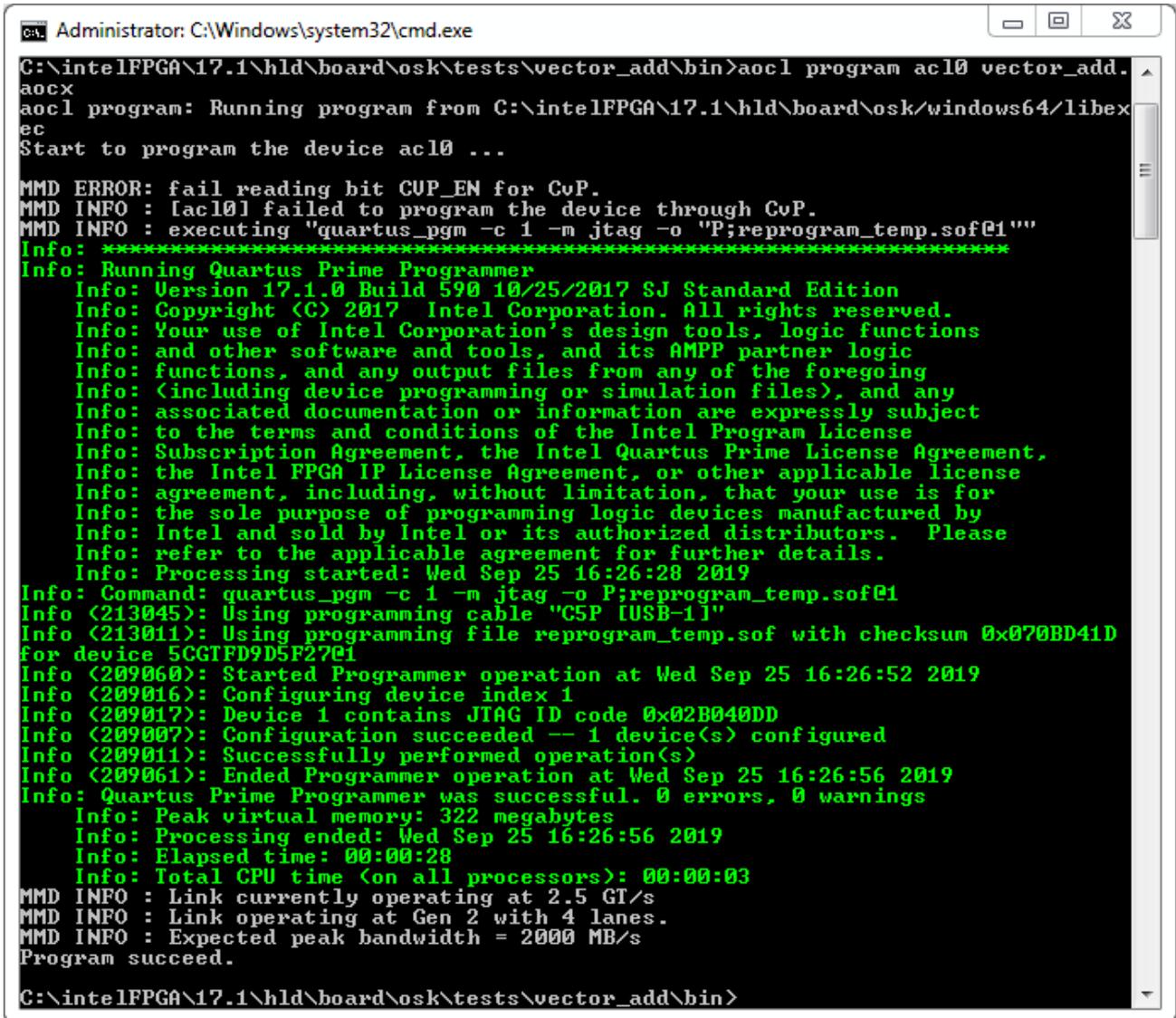


Figure 2-20 Message for vector_add Host Program build successfully

■ Test vector_add project

Firstly, in Command Prompt window, type “cd C:\intelFPGA\17.1\hld\board\osk\tests\vector_add\bin” to go to **vector_add\bin** project folder,

And type “aocl program acl0 vector_add.aocx” to program the bitstream into FPGA board as show in **Figure 2-21**.



```
Administrator: C:\Windows\system32\cmd.exe
C:\intelFPGA\17.1\hld\board\osk\tests\vector_add\bin>aocl program acl0 vector_add.aocx
aocl program: Running program from C:\intelFPGA\17.1\hld\board\osk\windows64\libexec
Start to program the device acl0 ...

MMD ERROR: fail reading bit CUP_EN for CoP.
MMD INFO : [acl0] failed to program the device through CoP.
MMD INFO : executing "quartus_pgm -c 1 -m jtag -o "P;reprogram_temp.sof@1""
Info: *****
Info: Running Quartus Prime Programmer
Info: Version 17.1.0 Build 590 10/25/2017 SJ Standard Edition
Info: Copyright (C) 2017 Intel Corporation. All rights reserved.
Info: Your use of Intel Corporation's design tools, logic functions
Info: and other software and tools, and its AMPP partner logic
Info: functions, and any output files from any of the foregoing
Info: (including device programming or simulation files), and any
Info: associated documentation or information are expressly subject
Info: to the terms and conditions of the Intel Program License
Info: Subscription Agreement, the Intel Quartus Prime License Agreement,
Info: the Intel FPGA IP License Agreement, or other applicable license
Info: agreement, including, without limitation, that your use is for
Info: the sole purpose of programming logic devices manufactured by
Info: Intel and sold by Intel or its authorized distributors. Please
Info: refer to the applicable agreement for further details.
Info: Processing started: Wed Sep 25 16:26:28 2019
Info: Command: quartus_pgm -c 1 -m jtag -o P;reprogram_temp.sof@1
Info (213045): Using programming cable "C5P [USB-1]"
Info (213011): Using programming file reprogram_temp.sof with checksum 0x070BD41D
for device 5CGTFD9D5F27@1
Info (209060): Started Programmer operation at Wed Sep 25 16:26:52 2019
Info (209016): Configuring device index 1
Info (209017): Device 1 contains JTAG ID code 0x02B040DD
Info (209007): Configuration succeeded -- 1 device(s) configured
Info (209011): Successfully performed operation(s)
Info (209061): Ended Programmer operation at Wed Sep 25 16:26:56 2019
Info: Quartus Prime Programmer was successful. 0 errors, 0 warnings
Info: Peak virtual memory: 322 megabytes
Info: Processing ended: Wed Sep 25 16:26:56 2019
Info: Elapsed time: 00:00:28
Info: Total CPU time (on all processors): 00:00:03
MMD INFO : Link currently operating at 2.5 GT/s
MMD INFO : Link operating at Gen 2 with 4 lanes.
MMD INFO : Expected peak bandwidth = 2000 MB/s
Program succeed.

C:\intelFPGA\17.1\hld\board\osk\tests\vector_add\bin>
```

Figure 2-21 Program bitstream into FPGA

Then, execute “host.exe”. **Figure 2-22** is the screen shot when the test is successful.

```
Administrator: C:\Windows\system32\cmd.exe
Info: Quartus Prime Programmer was successful. 0 errors, 0 warnings
Info: Peak virtual memory: 322 megabytes
Info: Processing ended: Wed Sep 25 16:26:56 2019
Info: Elapsed time: 00:00:28
Info: Total CPU time (on all processors): 00:00:03
MMD INFO : Link currently operating at 2.5 GT/s
MMD INFO : Link operating at Gen 2 with 4 lanes.
MMD INFO : Expected peak bandwidth = 2000 MB/s
Program succeed.

C:\intelFPGA\17.1\hld\board\osk\tests\vector_add\bin>host
Initializing OpenCL
Platform: Intel(R) FPGA SDK for OpenCL(TM)
Using 1 device(s)
  c5gt : HPC Reference Platform
Using AOCL: vector_add.aocl
Launching for device 0 (zd elements)

Time: 18.285 ms
Kernel time (device 0): 7.867 ms

Verification: PASS

C:\intelFPGA\17.1\hld\board\osk\tests\vector_add\bin>
```

Figure 2-22 “vector_add” test successfully

Chapter 3

OpenCL for Linux

This chapter describes how to setup OSK OpenCL development environment on 64-bit Linux (Red Hat Enterprise Linux 6.5/CentOS 7.0/Ubuntu14.04 are recommended), and how to compile and test the OpenCL examples for OSK. For more details about Intel OpenCL , please refer to Intel SDK for OpenCL Getting Started document:

https://www.altera.com/en_US/pdfs/literature/hb/opencl-sdk/aocl_getting_started.pdf

3.1 Software Installation

This section describes how to download and install the required software for OpenCL.

■ **Intel Quartus Prime and OpenCL**

Quartus Prime Standard Edition 17.1 and Intel FPGA SDK for OpenCL 17.1 can be downloaded from the web site:

<http://dl.altera.com/opencl/17.1/?edition=standard>

For Quartus Prime installation, please make sure that the Cyclone V device is included.

Open the link and select the **Linux SDK** as **Figure 3-1** shows.

- Design Software
- Embedded Software
- Archives
- Licensing
- Programming Software
- Drivers
- Board System Design
- Board Layout and Test
- Legacy Software

Intel FPGA SDK for OpenCL™

Release date: November, 2017
 Latest Release: v17.1

Select edition: Standard
 Select release: 17.1

Download Method Akamai DLM3 Download Manager Direct

Windows SDK
Linux SDK
RTE
Updates

Download and install instructions: [More](#)
[Read Intel FPGA SDK for OpenCL Getting Started Guide](#)

Intel FPGA SDK for OpenCL (includes Quartus Prime software and devices) ⓘ

Size: 20.6 GB MD5: ED301BE7806917D48FD953026720629A

Download

1. Quartus Prime Standard Edition
2. Intel FPGA SDK for OpenCL
3. Arria 10 Part 1
4. Arria 10 Part 2
5. Arria 10 Part 3
6. Arria V
7. Cyclone V
8. Stratix V

Figure 3-1 Linux SDK table

Quartus Prime software uses the built-in USB-Blaster II driver on Linux to access USB-Blaster II download cable on OSK. But after installed the Quartus Prime software with built-in driver, user needs to change the port permission for USB-Blaster II via issuing

```
'gedit /etc/udev/rules.d/51-usbblaster.rules'
```

to create and add the following lines to the **/etc/udev/rules.d/51-usbblaster.rules** file.

```
# USB-Blaster
ENV{ID_BUS}=="usb" ENV{ID_VENDOR_ID}=="09fb", ENV{ID_MODEL_ID}=="6001", MODE="0666"
ENV{ID_BUS}=="usb" ENV{ID_VENDOR_ID}=="09fb", ENV{ID_MODEL_ID}=="6002", MODE="0666"
ENV{ID_BUS}=="usb" ENV{ID_VENDOR_ID}=="09fb", ENV{ID_MODEL_ID}=="6003", MODE="0666"
# USB-Blaster II
ENV{ID_BUS}=="usb" ENV{ID_VENDOR_ID}=="09fb", ENV{ID_MODEL_ID}=="6010", MODE="0666"
ENV{ID_BUS}=="usb" ENV{ID_VENDOR_ID}=="09fb", ENV{ID_MODEL_ID}=="6810", MODE="0666"
```

Note: You must have system administration (root) privileges to configure the USB-Blaster II download cable driver.

■ GNU development tools

GNU development tools such as **gcc**(include **g++**) and **make** are required to build the driver and application under Linux. And the gcc version must gcc-4.8.0 or later. User can issue ‘yum install gcc compat-gcc-c++ make’ command to download and install them and their dependencies via internet

Note: To install the SDK on Linux, you must install it in a directory that you own (a directory which is not a system directory). You must also have sudo or root privileges.

■ OSK OpenCL BSP (Board Support Package)

After Quartus Prime and OpenCL SDK are installed, please download the OSK_OpenCL_BSP_17.1.tar.gz linux BSP for Intel FPGA OpenCL 17.1 from the web:

<http://osk.terasic.com/cd>

Then, decompress OSK_OpenCL_BSP_17.1.tar.gz to the “**osk**” folder under the folder “/root/intelFPGA/17.1/hld/board”, where assumed Quartus Prime is installed on the folder “/root/intelFPGA /17.1”, as shown in **Figure 3-2**.

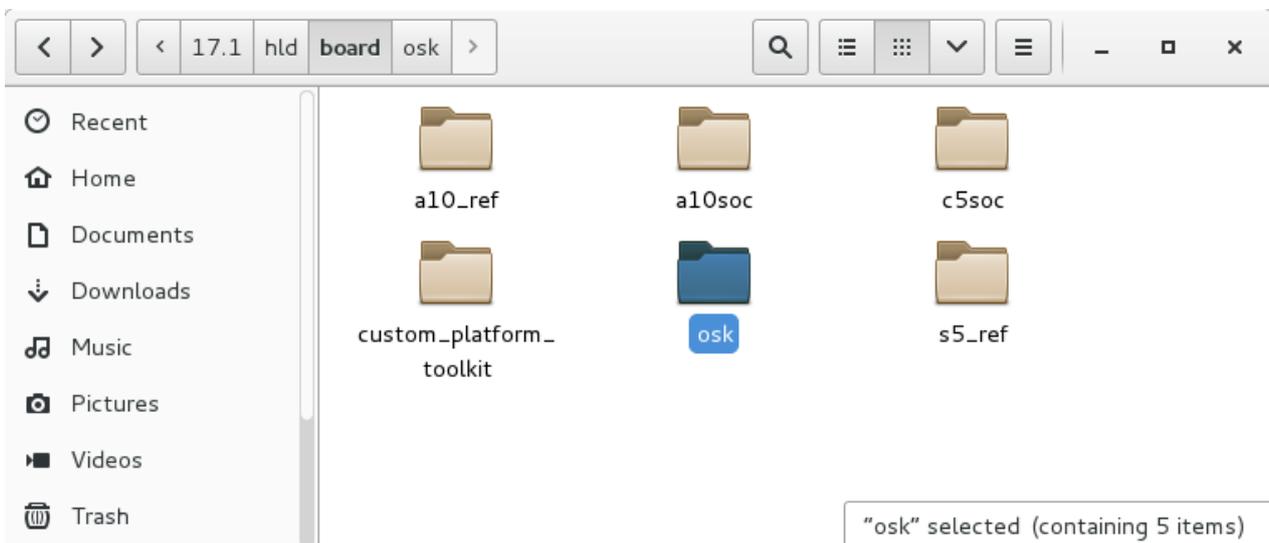


Figure 3-2 OSK OpenCL BSP Content

For more details about OSK OpenCL BSP, please refer to the [Table 2](#).

Table 2 Linux BSP File

File or Folder	Description
board_env.xml	eXtensible Markup Language (XML) file that describes the Reference Platform to the Intel FPGA SDK for OpenCL.
Hardware	Contains the Intel Quartus Prime project templates for the OSK board variant.
Linux64	Contains the MMD library, kernel mode driver, and executable files of the SDK utilities (that is, install, uninstall, flash, program, diagnose) for your 64-bit operating system
Tests	Contains some OpenCL Design Examples. The following examples demonstrate how to describe various applications in OpenCL along with their respective host applications, which you can compile and execute on a host with an FPGA board that supports the Intel FPGA SDK for OpenCL.
Bringup	The demo batch files of initializing the OSK for OpenCL Use

3.2 Environment Configure

If you install the Intel FPGA development software and OpenCL SDK on a system that does not contain any `.cshrc` or Bash Resource file (`.bashrc`) in your directory, you must set the `INTELFPGAOCLSDKROOT` and `PATH` environment variables manually. And for Intel FPGA OpenCL SDK able to find the kit location of OSK correctly, the developers need to create an environment variable for the OSK board `AOCL_BOARD_PACKAGE_ROOT`, and set its value as:

```
"$INTELFPGAOCLSDKROOT"/board/osk"
```

Alternatively, you can edit the `"/etc/profile"` **profile** file, and append the environment variables to it. To do this type `"gedit /etc/profile"` command on Linux Terminal to open the **profile** file by the **gedit** editor tool, and append the following setting to the **profile** file. Then, save the file and type `"source /etc/profile"` command in Linux Terminal to make the settings effect.

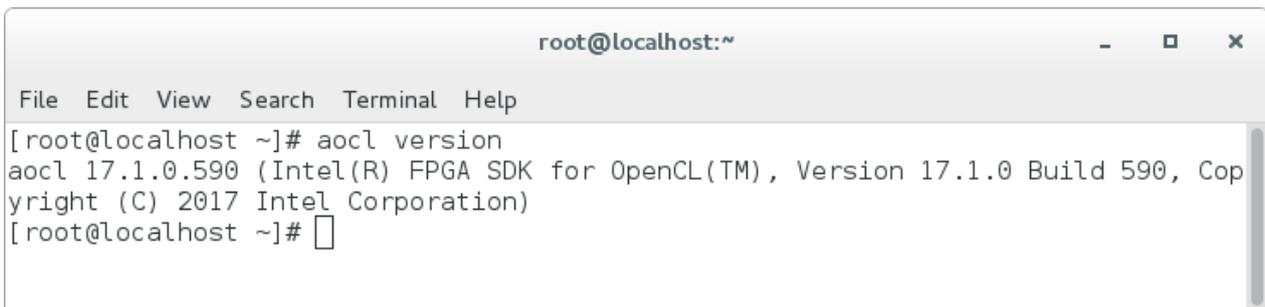
```
export QUARTUS_ROOTDIR=/root/intelFPGA/17.1/quartus
export INTELFPGAOCLSDKROOT=/root/ intelFPGA/17.1/hld
export AOCL_BOARD_PACKAGE_ROOT=/root/intelFPGA/17.1/hld/board/osk
export PATH=$PATH:$INTELFPGAOCLSDKROOT/linux64/bin:$INTELFPGAOCLSDKROOT/bin:$INTELFPGA
AOCLSDKROOT/host/linux64/bin:$QUARTUS_ROOTDIR/bin
export LD_LIBRARY_PATH=$AOCL_BOARD_PACKAGE_ROOT/tests/extlibs/lib:$INTELFPGAOCLSDKRO
OT/host/linux64/lib:$AOCL_BOARD_PACKAGE_ROOT/linux64/lib
export CL_CONTEXT_COMPILER_MODE_INTELFPGA=3
export QUARTUS_64BIT=1
export LM_LICENSE_FILE=/root/intelFPGA/17.1/hld/license.dat
```

3.3 OpenCL Environment Verify

This section will show how to make sure the OpenCL environment is setup correctly. Firstly, please open the Linux system terminal window by right click the Mouse on system desktop, then clicking on Open Terminal.

■ Target SDK Version

In the Linux **terminal**, type “**aocl version**” command, and make sure the version of the OpenCL SDK is listed as shown in **Figure 3-3**.

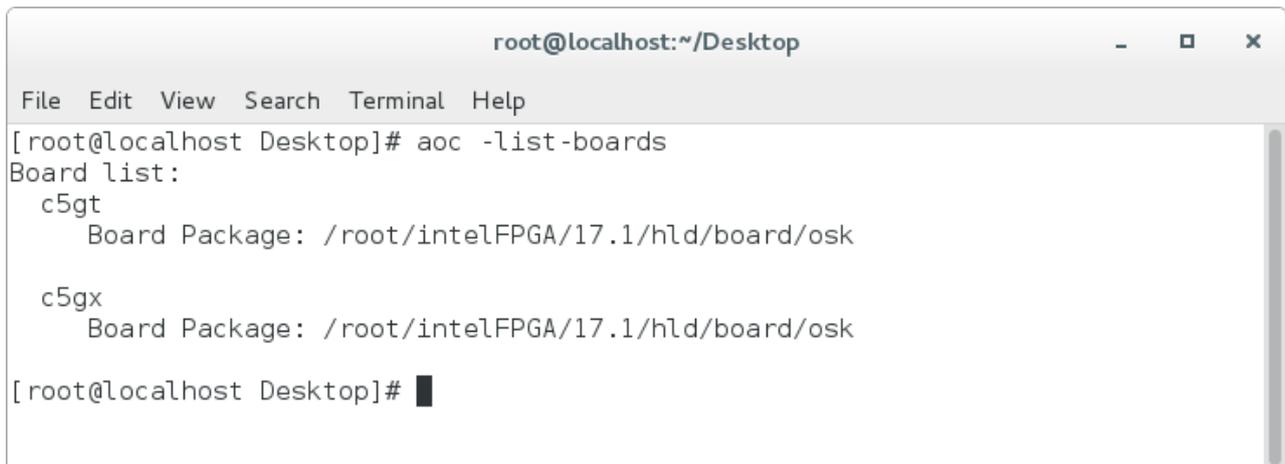
A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command '[root@localhost ~]# aocl version' has been executed, resulting in the output: 'aocl 17.1.0.590 (Intel(R) FPGA SDK for OpenCL(TM), Version 17.1.0 Build 590, Copyright (C) 2017 Intel Corporation)'. The prompt '[root@localhost ~]#' is visible again.

```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# aocl version  
aocl 17.1.0.590 (Intel(R) FPGA SDK for OpenCL(TM), Version 17.1.0 Build 590, Cop  
yright (C) 2017 Intel Corporation)  
[root@localhost ~]#
```

Figure 3-3 Version of OpenCL SDK

■ Target Board

In the Linux **terminal**, type “**aoc -list-boards**” command, and make sure “**osk**” is listed in **Board list** as shown in **Figure 3-4**.

A terminal window titled 'root@localhost:~/Desktop' with a menu bar (File, Edit, View, Search, Terminal, Help). The command '[root@localhost Desktop]# aoc -list-boards' has been executed, resulting in the output: 'Board list: c5gt Board Package: /root/intelFPGA/17.1/hld/board/osk c5gx Board Package: /root/intelFPGA/17.1/hld/board/osk'. The prompt '[root@localhost Desktop]#' is visible again.

```
root@localhost:~/Desktop  
File Edit View Search Terminal Help  
[root@localhost Desktop]# aoc -list-boards  
Board list:  
  c5gt  
    Board Package: /root/intelFPGA/17.1/hld/board/osk  
  
  c5gx  
    Board Package: /root/intelFPGA/17.1/hld/board/osk  
[root@localhost Desktop]#
```

Figure 3-4 ‘osk’ is Listed in Board List

For more information about the aoc and aocl, refer to the ‘aoc -h’ and ‘aocl help’ command.

3.4 Initializing the FPGA for use with OpenCL

■ Board Setup

Before testing OpenCL on OSK, please follow the below procedures to set up the board on your PC as shown in [Figure 3-5](#).

1. Make sure your PC is powered off.
2. Insert OSK board into PCI Express x4/x8 or x16 slot.
3. Connect 12V power source to the OSK
4. Connect PC's USB port to OSK UB2 port using a USB cable.

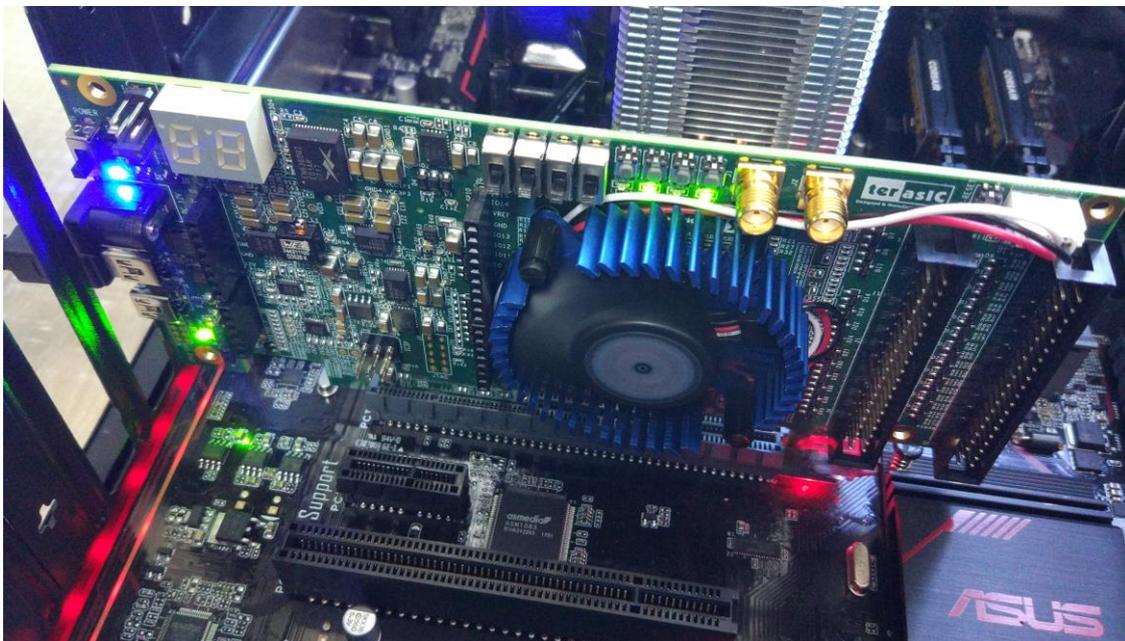


Figure 3-5 Setup OSK board on PC

■ ‘aocl flash’ program

The **flash** utility in the OSK Development Kit Reference Platform configures the power-on image for the FPGA using the specified .aocx file. Calling into the MMD library implements the flash utility.

In the terminal, type “`cd /root/intelFPGA/17.1/hld/board/osk/bringup/<board name>`” to go to bringup folder of the board, then type “`aocl flash acl0 hello_world.aocx`” to program **hello_world.aocx** OpenCL image into the startup configuration flash of OSK. It will take about 5 minutes for flash programming as shown in [Figure 3-6](#).

```
root@localhost:~/intelFPGA/17.1/hld/board/osk/bringup/c5gt
File Edit View Search Terminal Help
[root@localhost c5gt]# aocl flash acl0 hello_world.aocx
aocl flash: Running flash from /root/intelFPGA/17.1/hld/board/osk/linux64/libexec
Flash Programming of c5gt ...
Info: *****
Info: Running Quartus Prime Convert_programming_file
Info: Version 17.1.0 Build 590 10/25/2017 SJ Standard Edition
Info: Copyright (C) 2017 Intel Corporation. All rights reserved.
Info: Your use of Intel Corporation's design tools, logic functions
Info: and other software and tools, and its AMPP partner logic
Info: functions, and any output files from any of the foregoing
Info: (including device programming or simulation files), and any
Info: associated documentation or information are expressly subject
Info: to the terms and conditions of the Intel Program License
Info: Subscription Agreement, the Intel Quartus Prime License Agreement,
Info: the Intel FPGA IP License Agreement, or other applicable license
Info: agreement, including, without limitation, that your use is for
Info: the sole purpose of programming logic devices manufactured by
Info: Intel and sold by Intel or its authorized distributors. Please
Info: refer to the applicable agreement for further details.
Info: Processing started: Thu Sep 26 14:21:10 2019
Info: Command: quartus_cpf -c -d epcq256 -s 5cgtfd9d5 -m ASx4 flash.sof flash.ji
c
```

Figure 3-6 ‘aocl flash acl0 hello_world.aocx’

After flash programming is done successfully, **developers must power off OSK board and PC, then restart the PC.**

■ Driver Installation

Your system must recognize the card so that the Intel FPGA SDK for OpenCL driver can be loaded. The **install** utility is used to install the kernel driver on the host computer. Users of the Intel FPGA SDK for OpenCL only need to install the driver once, after that the driver should be automatically loaded each time when the machine reboots.

First, in the Linux **terminal**, type ‘**lspci |grep Altera**’ to make sure the system recognizes the PCIe card as shown in **Figure 3-7**.

```
root@localhost:~/Desktop
File Edit View Search Terminal Help
[root@localhost Desktop]# lspci | grep Altera
01:00.0 Unassigned class [ff00]: Altera Corporation Device d800 (rev 01)
[root@localhost Desktop]# █
```

Figure 3-7 PCIE Message

Type ‘**aocl install**’ to install the driver as shown in **Figure 3-8**. Note that users need to have **root privileges** to install the driver.

```
root@localhost:~
File Edit View Search Terminal Help
[root@localhost ~]# aocl install
Do you want to install /root/intelFPGA/17.1/hld/board/osk? [y/n] y
aocl install: Running install from /root/intelFPGA/17.1/hld/board/osk/linux64/libexec
Using kernel source files from /lib/modules/3.10.0-327.el7.x86_64/build
make: Entering directory `/usr/src/kernels/3.10.0-327.el7.x86_64'
  CC [M] /tmp/opencv_driver_LkWhYA/aclpci_queue.o
  CC [M] /tmp/opencv_driver_LkWhYA/aclpci.o
  CC [M] /tmp/opencv_driver_LkWhYA/aclpci_fileio.o
  CC [M] /tmp/opencv_driver_LkWhYA/aclpci_dma.o
  CC [M] /tmp/opencv_driver_LkWhYA/aclpci_cvp.o
  CC [M] /tmp/opencv_driver_LkWhYA/aclpci_cmd.o
  LD [M] /tmp/opencv_driver_LkWhYA/aclpci_drv.o
Building modules, stage 2.
MODPOST 1 modules
  CC /tmp/opencv_driver_LkWhYA/aclpci_drv.mod.o
  LD [M] /tmp/opencv_driver_LkWhYA/aclpci_drv.ko
make: Leaving directory `/usr/src/kernels/3.10.0-327.el7.x86_64'
is not an object at /root/intelFPGA/17.1/hld/share/lib/perl/acl/Command.pm line
1290, <F> line 34.
[root@localhost ~]#
```

Figure 3-8 driver installation

Note: if user don't use the recommended Linux system or version, recompiling the driver is needed. You can compile it by typing `"cd root/intelFPGA/17.1/hld/board/osk/linux64/driver"` (there are source code, makefile and readme.txt) to *locate at the* driver source code directory and type `"/make_all"` to compile and generate the new driver. Before recompiling the driver, user need to install the kernel package (which version should be matched the current Linux system) via issuing `'yum install kernel-devel'` command.

3.5 OpenCL Runtime Verify

This section will show how to make sure the OpenCL runtime environment is setup correctly.

Firstly, please open the Linux system **terminal** window by right click the Mouse on system desktop, then clicking on **Open Terminal**.

■ Test ‘aocl diagnose’ Command

The **diagnose** utility in the OSK board reports device information and identifies issues. The diagnose utility first verifies the installation of the kernel driver and returns the overall information of all the devices installed in a host machine.

The diagnose utility in the OSK board reports device information and identifies issues. The diagnose utility first verifies the installation of the kernel driver and returns the overall information of all the devices installed in a host machine.

In the Linux terminal, type “**aocl diagnose**” to check if the initialization completed successfully. If successful, the programming message displays “**DIAGNOSTIC_PASSED**” as shown in [Figure 3-9](#).

```
root@localhost:~
File Edit View Search Terminal Help
[root@localhost ~]# aocl diagnose

Verified that the kernel mode driver is installed on the host machine.

Using board package from vendor: Terasic
Querying information for all supported devices that are installed on the host machine ...

Device Name    Status    Information
aocl0          Passed    Cyclone V HPC Reference Platform
              PCIe dev_id = D800, bus:slot.func = 01:00.00, at Gen 2 with 4 lanes

Found 1 active device(s) installed on the host machine. To perform a full diagnostic on a specific device, please run
    aocl diagnose <device_name>

DIAGNOSTIC_PASSED

Call "aocl diagnose <device-names>" to run diagnose for specified devices
Call "aocl diagnose all" to run diagnose for all devices
[root@localhost ~]#
```

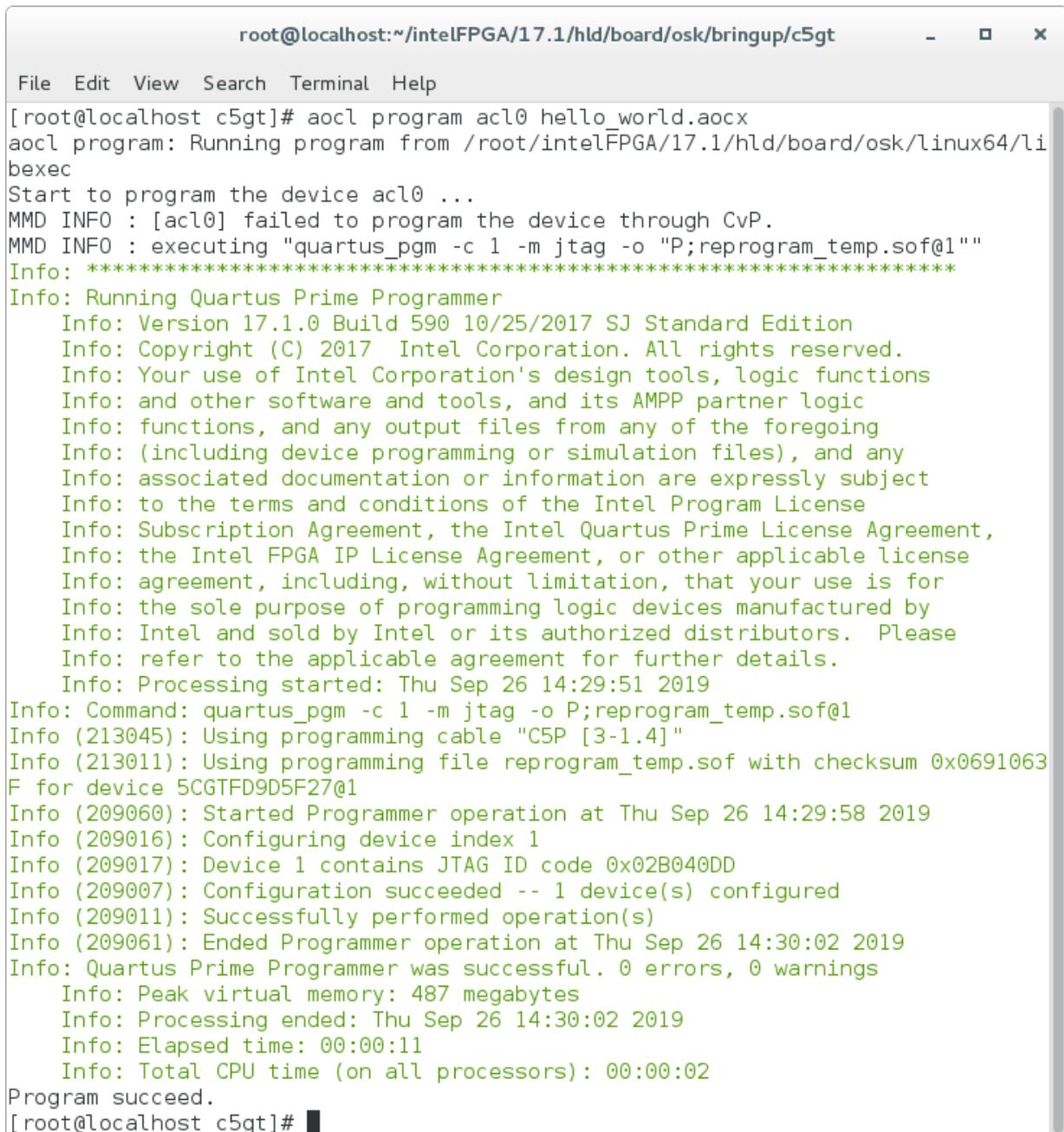
Figure 3-9 “aocl diagnose” messages

Note: The Cyclone V GT device supports PCIe Gen 2 x4 speed and GX device supports PCIe Gen 1 x4 speed.

■ Test ‘aocl program’ Command

The **program** utility in the OSK Development Kit Reference Platform programs the board with the specified .aocx file use the UB2 port.

Check whether the **hello_world** OpenCL image configures the FPGA successfully when OSK is power on. In the Linux terminal, type “*cd /root/intelFPGA/17.1/hld/board/osk/bringup/<board name>*” to go to bringup folder of the board, then type “*aocl program acl0 hello_world.aocx*” to configure the FPGA with **hello_world.aocx** OpenCL image. If the programming message display “Program succeed” as shown in **Figure 3-10**, it means the **hello_world** OpenCL image is programmed into the flash correctly.



```
root@localhost:~/intelFPGA/17.1/hld/board/osk/bringup/c5gt
File Edit View Search Terminal Help
[root@localhost c5gt]# aocl program acl0 hello_world.aocx
aocl program: Running program from /root/intelFPGA/17.1/hld/board/osk/linux64/libexec
Start to program the device acl0 ...
MMD INFO : [acl0] failed to program the device through CvP.
MMD INFO : executing "quartus_pgm -c 1 -m jtag -o "P;reprogram_temp.sof@1""
Info: *****
Info: Running Quartus Prime Programmer
Info: Version 17.1.0 Build 590 10/25/2017 SJ Standard Edition
Info: Copyright (C) 2017 Intel Corporation. All rights reserved.
Info: Your use of Intel Corporation's design tools, logic functions
Info: and other software and tools, and its AMPP partner logic
Info: functions, and any output files from any of the foregoing
Info: (including device programming or simulation files), and any
Info: associated documentation or information are expressly subject
Info: to the terms and conditions of the Intel Program License
Info: Subscription Agreement, the Intel Quartus Prime License Agreement,
Info: the Intel FPGA IP License Agreement, or other applicable license
Info: agreement, including, without limitation, that your use is for
Info: the sole purpose of programming logic devices manufactured by
Info: Intel and sold by Intel or its authorized distributors. Please
Info: refer to the applicable agreement for further details.
Info: Processing started: Thu Sep 26 14:29:51 2019
Info: Command: quartus_pgm -c 1 -m jtag -o P;reprogram_temp.sof@1
Info (213045): Using programming cable "C5P [3-1.4]"
Info (213011): Using programming file reprogram_temp.sof with checksum 0x0691063
F for device 5CGTFD9D5F27@1
Info (209060): Started Programmer operation at Thu Sep 26 14:29:58 2019
Info (209016): Configuring device index 1
Info (209017): Device 1 contains JTAG ID code 0x02B040DD
Info (209007): Configuration succeeded -- 1 device(s) configured
Info (209011): Successfully performed operation(s)
Info (209061): Ended Programmer operation at Thu Sep 26 14:30:02 2019
Info: Quartus Prime Programmer was successful. 0 errors, 0 warnings
Info: Peak virtual memory: 487 megabytes
Info: Processing ended: Thu Sep 26 14:30:02 2019
Info: Elapsed time: 00:00:11
Info: Total CPU time (on all processors): 00:00:02
Program succeed.
[root@localhost c5gt]#
```

Figure 3-10 “aocl program acl0 hello_world.aocx” use UB2

3.6 Compile and Test OpenCL Project

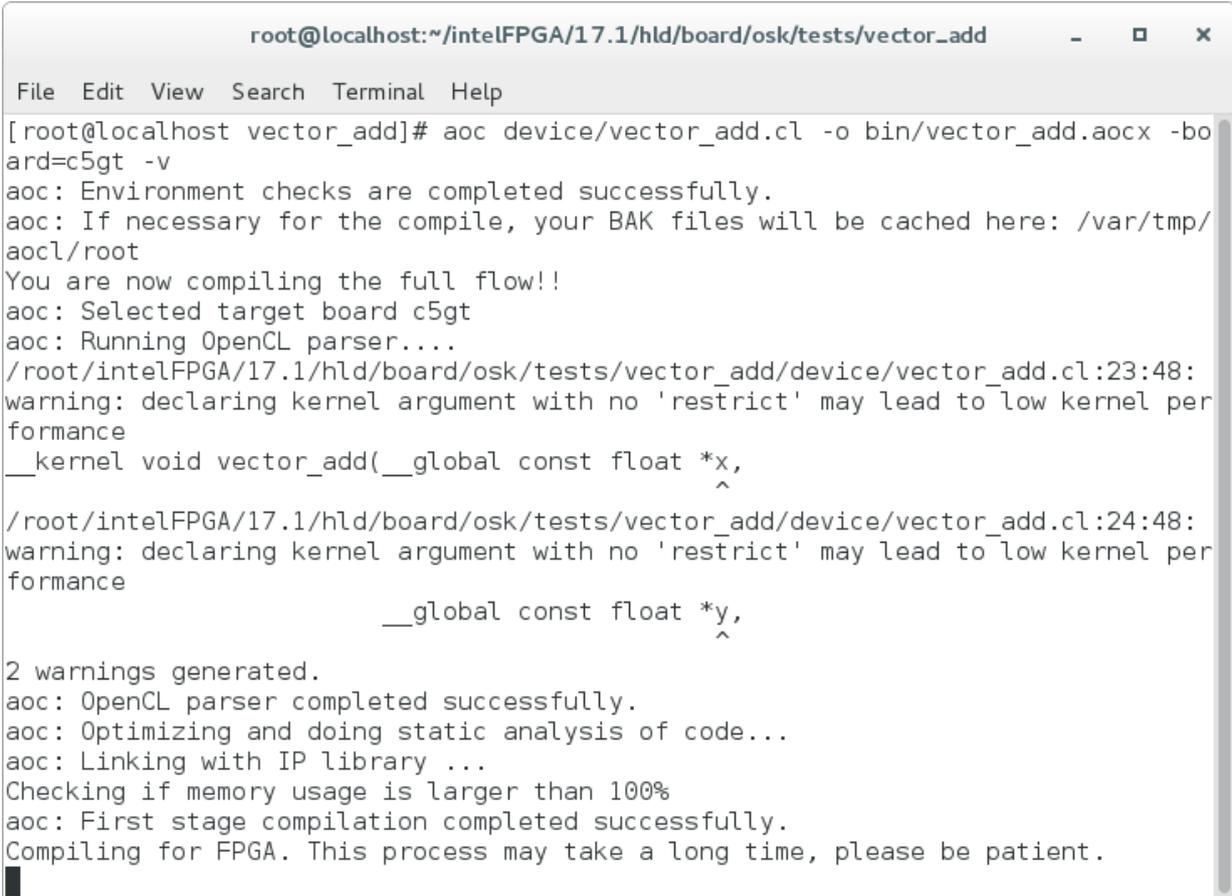
This section will show how to compile and run the OpenCL kernel and OpenCL Host Program for the `vector_add` example project. Developers can use the same procedures to compile and test other OpenCL examples (included in the kit) for OSK.

■ Compile OpenCL Kernel

In the terminal, type “`cd /root/intelFPGA/17.1/hld/board/osk/tests/vector_add`” to go to `vector_add` project folder, then type “`aoc device/vector_add.cl -o bin/vector_add.aocx -board=<board name> -report -v`” to compile the OpenCL kernel. It will takes about one hour for compiling. After that, the OpenCL image file `vector_add.aocx` is generated. [Figure 3-11](#) is the screen shot when OpenCL kernel is compiled successfully. For required parameters to compile `vector_add.cl`, please refer to the `README.html` that is in the same directory.

The utility `aoc` is used to compile OpenCL kernel. For detailed usage of `aoc`, please refer to the **Intel FPGA SDK for OpenCL Programming Guide**:

http://www.altera.com/literature/hb/opencl-sdk/aocl_programming_guide.pdf



```
root@localhost:~/intelFPGA/17.1/hld/board/osk/tests/vector_add
File Edit View Search Terminal Help
[root@localhost vector_add]# aoc device/vector_add.cl -o bin/vector_add.aocx -board=c5gt -v
aoc: Environment checks are completed successfully.
aoc: If necessary for the compile, your BAK files will be cached here: /var/tmp/aocl/root
You are now compiling the full flow!!
aoc: Selected target board c5gt
aoc: Running OpenCL parser...
/root/intelFPGA/17.1/hld/board/osk/tests/vector_add/device/vector_add.cl:23:48:
warning: declaring kernel argument with no 'restrict' may lead to low kernel performance
__kernel void vector_add(__global const float *x,
                        ^
/root/intelFPGA/17.1/hld/board/osk/tests/vector_add/device/vector_add.cl:24:48:
warning: declaring kernel argument with no 'restrict' may lead to low kernel performance
                        __global const float *y,
                        ^
2 warnings generated.
aoc: OpenCL parser completed successfully.
aoc: Optimizing and doing static analysis of code...
aoc: Linking with IP library ...
Checking if memory usage is larger than 100%
aoc: First stage compilation completed successfully.
Compiling for FPGA. This process may take a long time, please be patient.
```

Figure 3-11 ‘aoc vector_add.cl’ OpenCL kernel compile successfully

■ Compile Host Program

In the terminal, type “`cd /root/intelFPGA/17.1/hld/board/osk/tests/vector_add`” and then type “`make`” to compile the host program.

When compiling is successfully, you will see successful message as show in [Figure 3-12](#). The execute file is generate in the same directory which named bin.



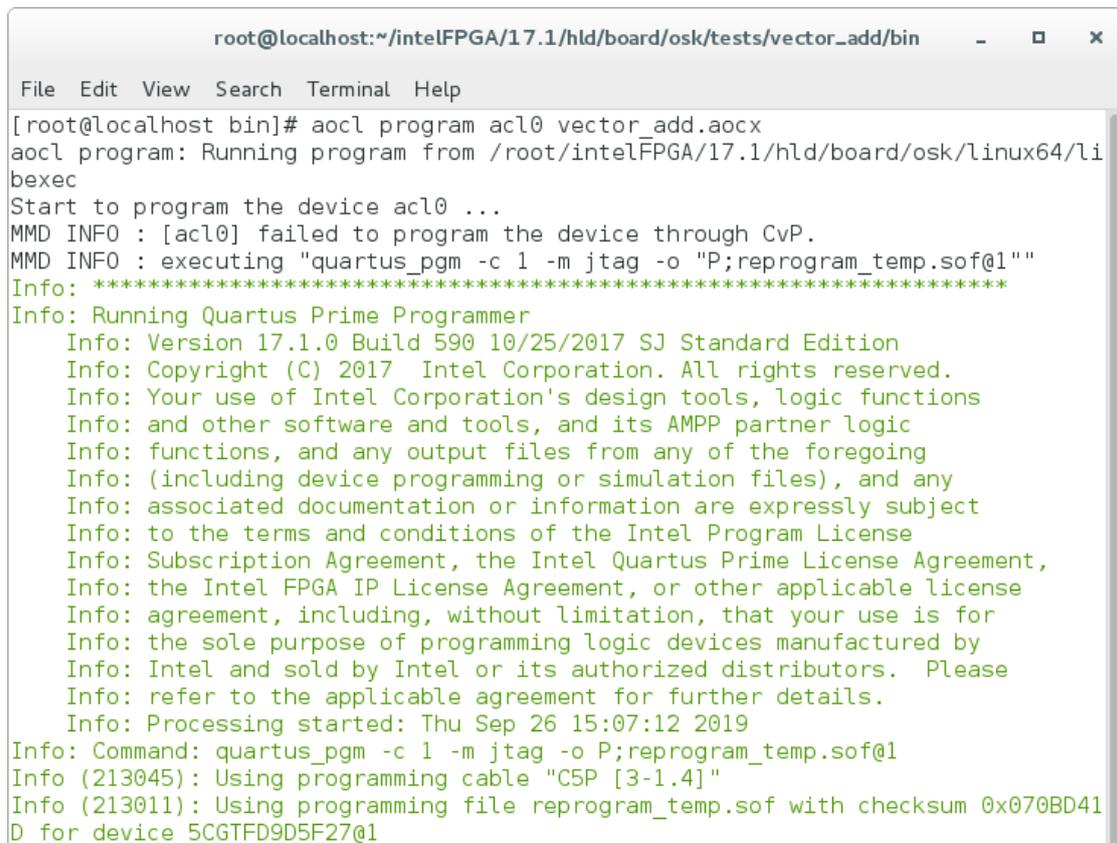
```
root@localhost:~/intelFPGA/17.1/hld/board/osk/tests/vector_add - □ x
File Edit View Search Terminal Help
[root@localhost vector_add]# make
[root@localhost vector_add]#
```

Figure 3-12 successful Message for vector_add Host Program build

■ Test vector_add project

Firstly, in the terminal, type “`cd /root/intelFPGA/17.1/hld/board/osk/tests/vector_add/bin`” to goto the **vector_add** project folder.

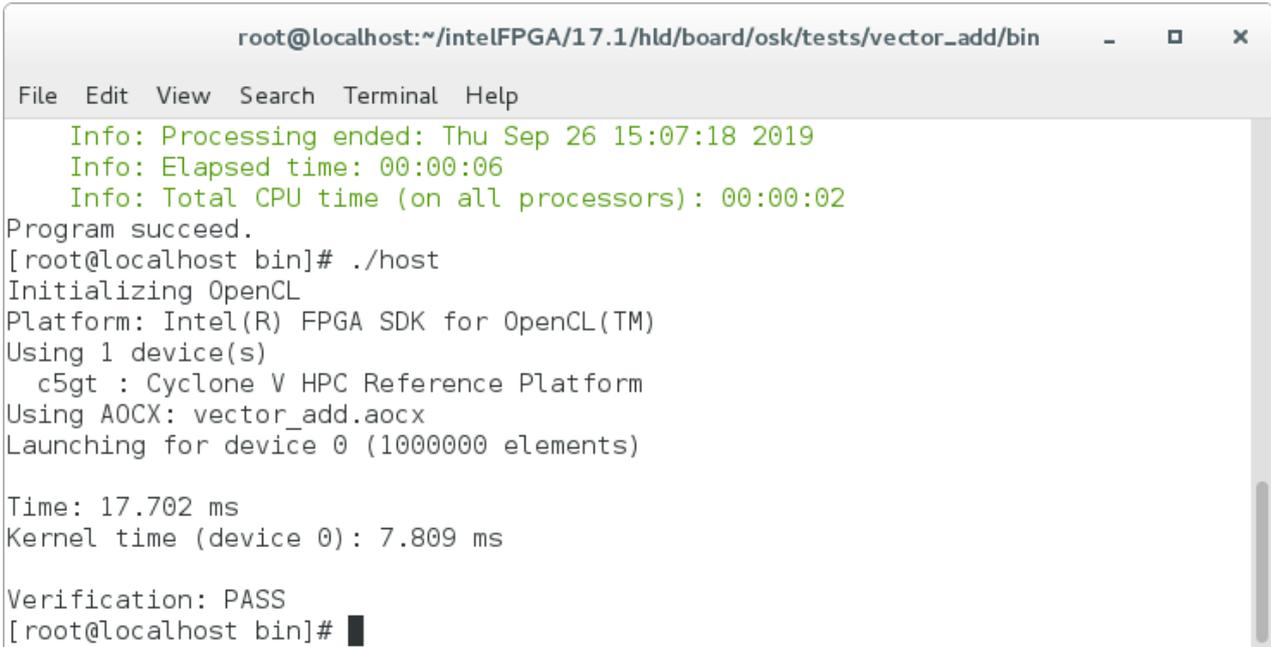
And type “`aocl program acl0 vector_add.aocx`” to program the bitstream into FPGA board as show in [Figure 3-13](#).



```
root@localhost:~/intelFPGA/17.1/hld/board/osk/tests/vector_add/bin - □ x
File Edit View Search Terminal Help
[root@localhost bin]# aocl program acl0 vector_add.aocx
aocl program: Running program from /root/intelFPGA/17.1/hld/board/osk/linux64/li
bexec
Start to program the device acl0 ...
MMD INFO : [acl0] failed to program the device through CvP.
MMD INFO : executing "quartus_pgm -c 1 -m jtag -o "P;reprogram_temp.sof@1"
Info: *****
Info: Running Quartus Prime Programmer
Info: Version 17.1.0 Build 590 10/25/2017 SJ Standard Edition
Info: Copyright (C) 2017 Intel Corporation. All rights reserved.
Info: Your use of Intel Corporation's design tools, logic functions
Info: and other software and tools, and its AMPP partner logic
Info: functions, and any output files from any of the foregoing
Info: (including device programming or simulation files), and any
Info: associated documentation or information are expressly subject
Info: to the terms and conditions of the Intel Program License
Info: Subscription Agreement, the Intel Quartus Prime License Agreement,
Info: the Intel FPGA IP License Agreement, or other applicable license
Info: agreement, including, without limitation, that your use is for
Info: the sole purpose of programming logic devices manufactured by
Info: Intel and sold by Intel or its authorized distributors. Please
Info: refer to the applicable agreement for further details.
Info: Processing started: Thu Sep 26 15:07:12 2019
Info: Command: quartus_pgm -c 1 -m jtag -o P;reprogram_temp.sof@1
Info (213045): Using programming cable "C5P [3-1.4]"
Info (213011): Using programming file reprogram_temp.sof with checksum 0x070BD41
D for device 5CGTFD9D5F27@1
```

Figure 3-13 Program bitstream into FPGA

Then, launch the compiled Host Program to start vector_add execute file for the test. In the terminal type “./host”. **Figure 3-14** shows the execution is successful.

A terminal window titled "root@localhost:~/intelFPGA/17.1/hld/board/osk/tests/vector_add/bin" with standard window controls. The terminal output shows the following text:

```
Info: Processing ended: Thu Sep 26 15:07:18 2019
Info: Elapsed time: 00:00:06
Info: Total CPU time (on all processors): 00:00:02
Program succeed.
[root@localhost bin]# ./host
Initializing OpenCL
Platform: Intel(R) FPGA SDK for OpenCL(TM)
Using 1 device(s)
  c5gt : Cyclone V HPC Reference Platform
Using AOCX: vector_add.aocx
Launching for device 0 (1000000 elements)

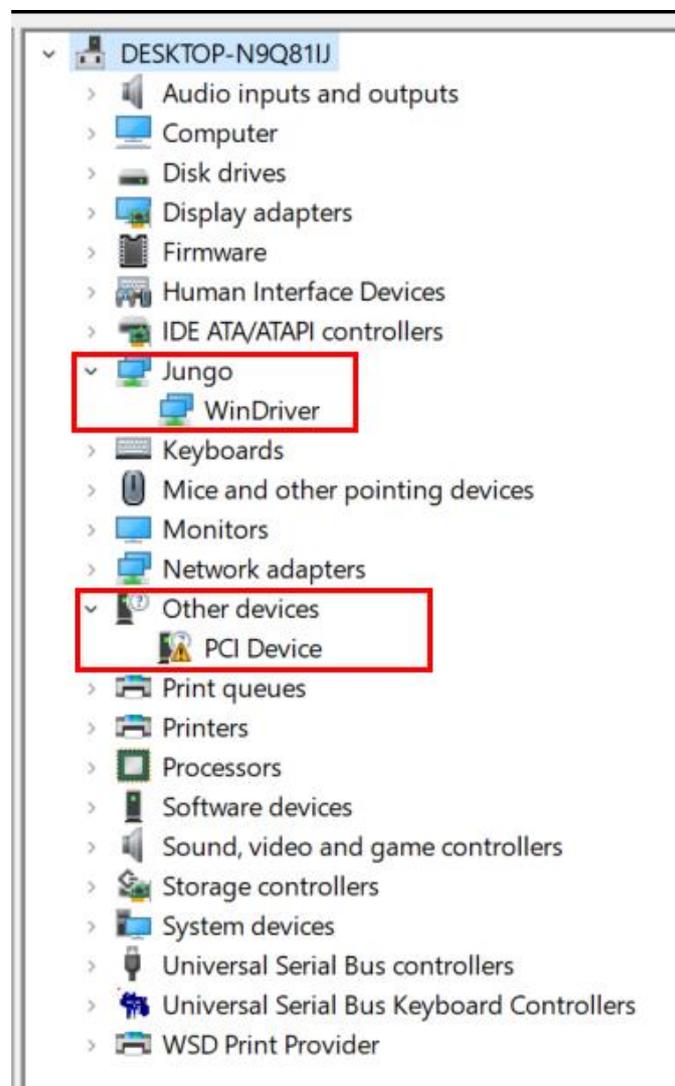
Time: 17.702 ms
Kernel time (device 0): 7.809 ms

Verification: PASS
[root@localhost bin]#
```

Figure 3-14 Successful Message for “vector_add” test

OSK Windows10 x64 OpenCL driver install

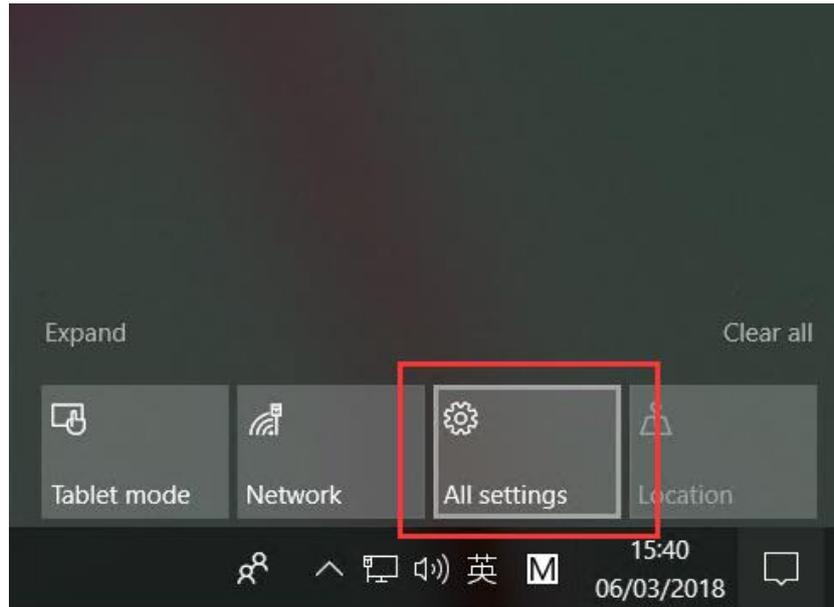
Because the Win10 system requires the signature of the inf file, sometimes, the driver of the PCIe (without signature) fails to be installed after running aocl install.



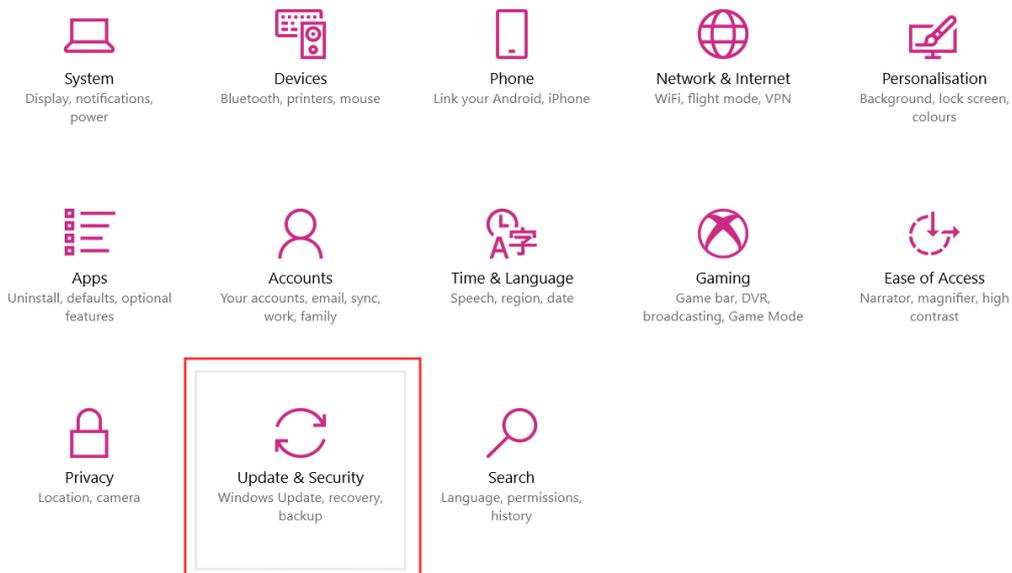
As a solution, it needs to disable the driver signature, then manually install the PCIe driver. The steps are as following:

A. Disable the driver signature in the Win10 system

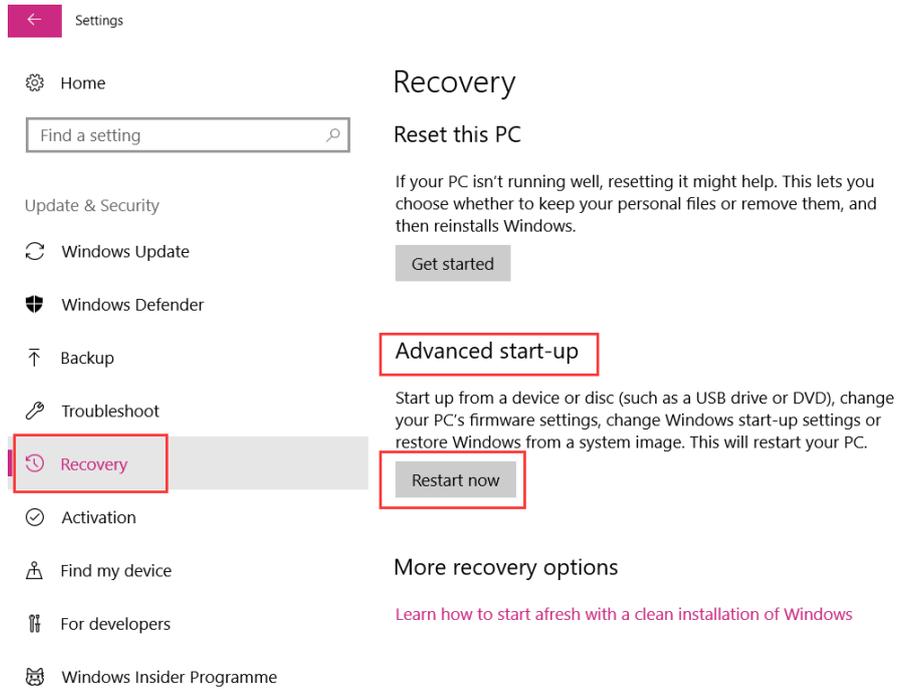
1. Click **Home**, enter “**All settings**”.



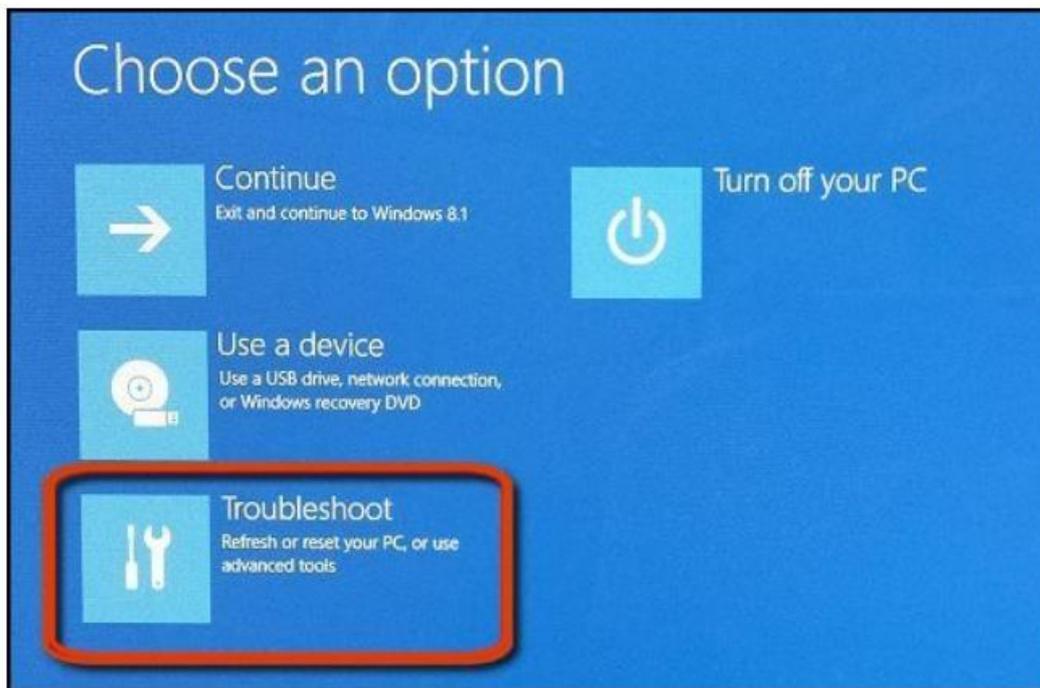
2. Access “**Update & Security**”.



3. Find **Recovery**. Click “**Restart now**” below “**Advanced start-up**”, restart the PC.



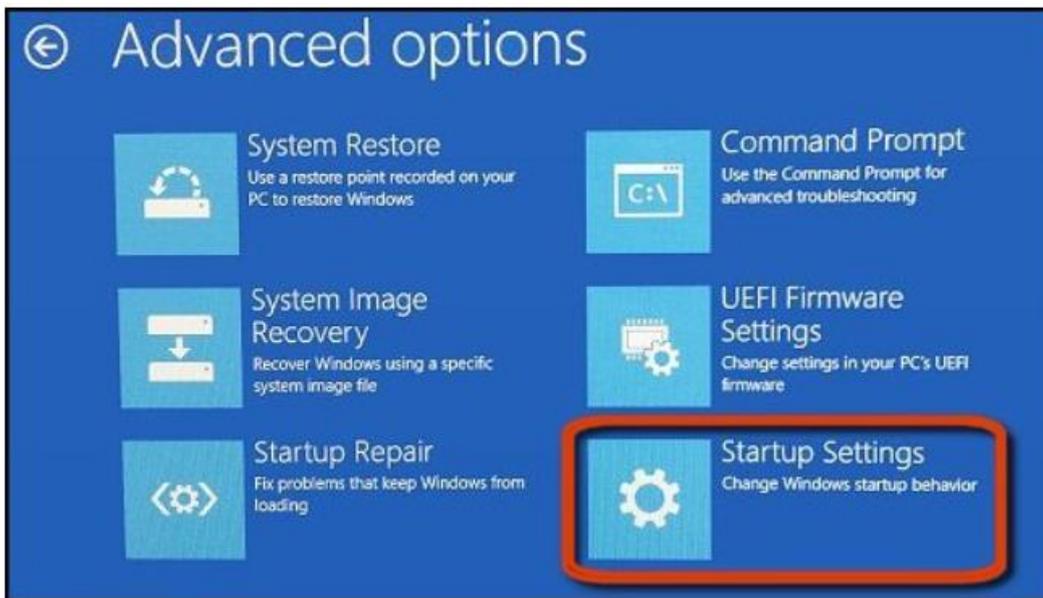
4. After restarting, choose “**Troubleshoot**”.



5. Choose “**Advanced options**”.



6. Choose “**Start-up Settings**”.



7. Click “**Restart**”.



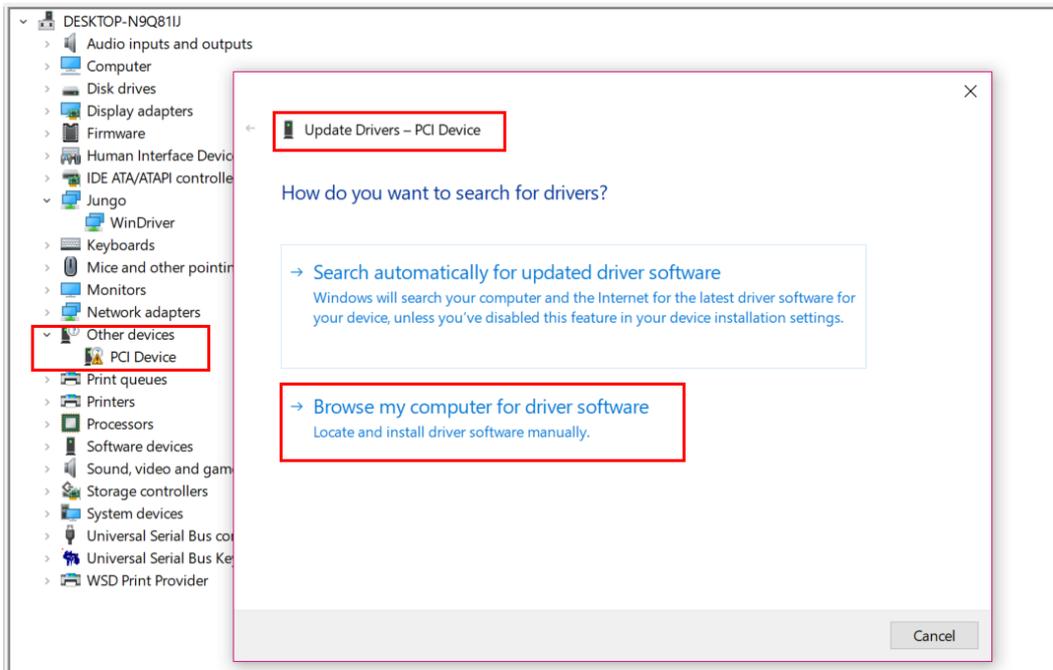
8. Enter “F7” to disable driver signature.



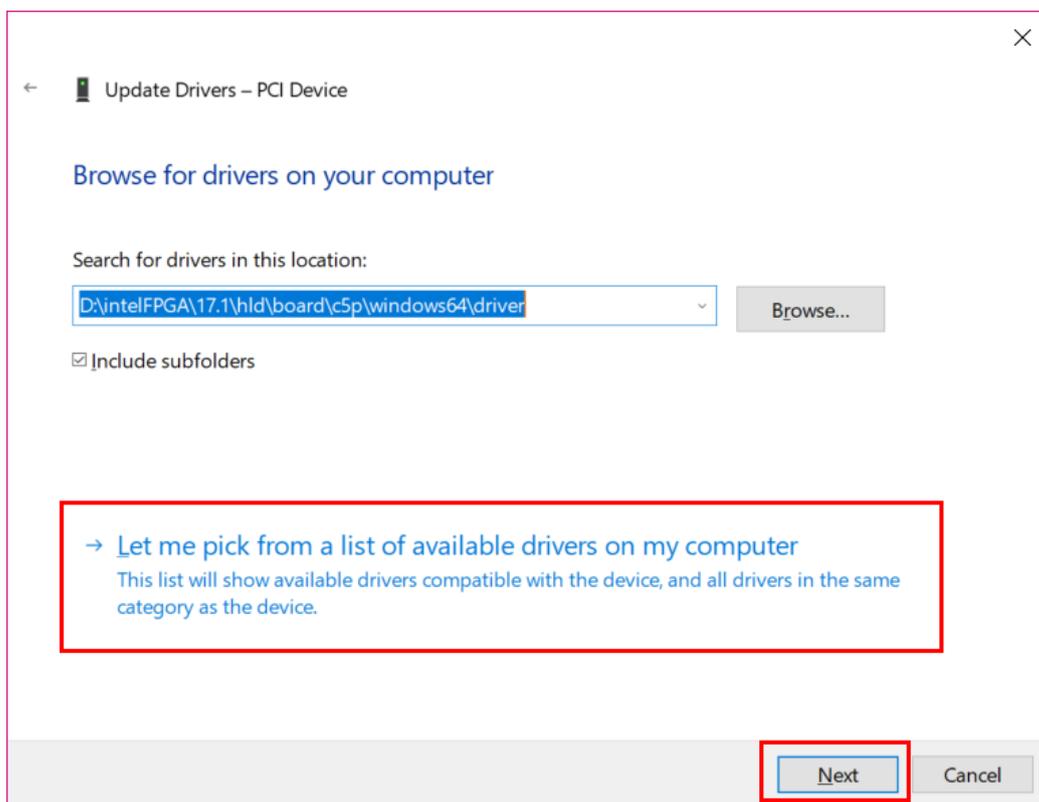
B. Install the PCI driver manually

1. After disable driver signature enforcement and restarting the system. Open the **Device Manager**, you can see a **PCI Device** with a yellow exclamation mark.

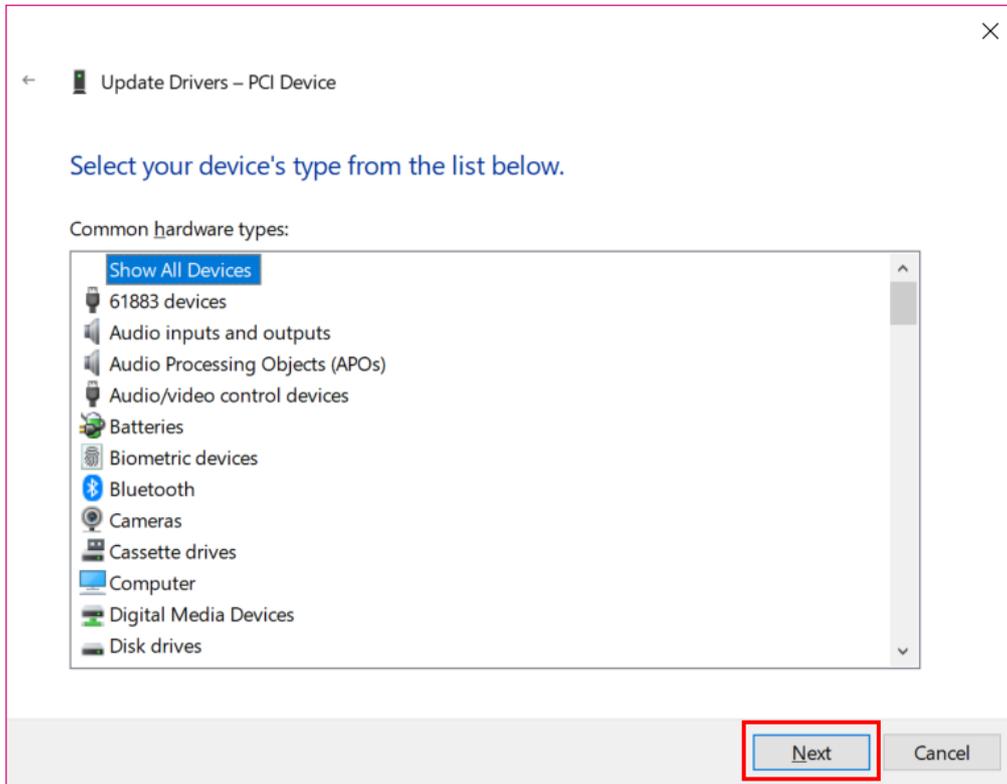
Right Click --> Update Drivers - PCI Device --> Browse my computer for driver software



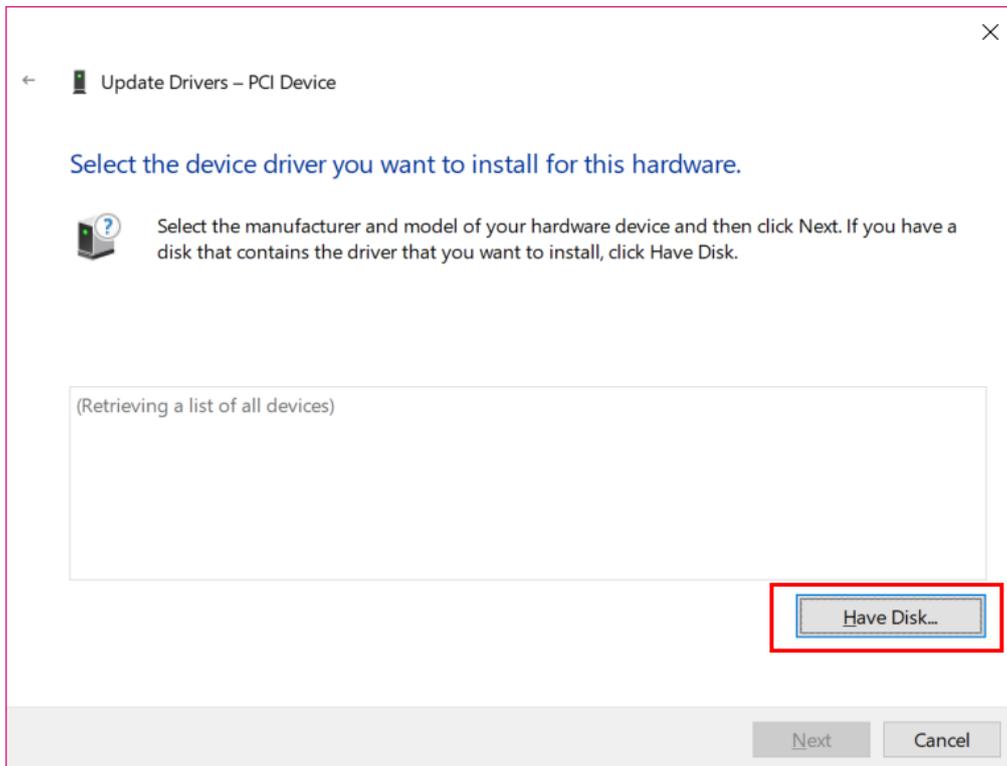
2. Choose “Let me pick from a list of available drivers on my computer”, Click “Next”.



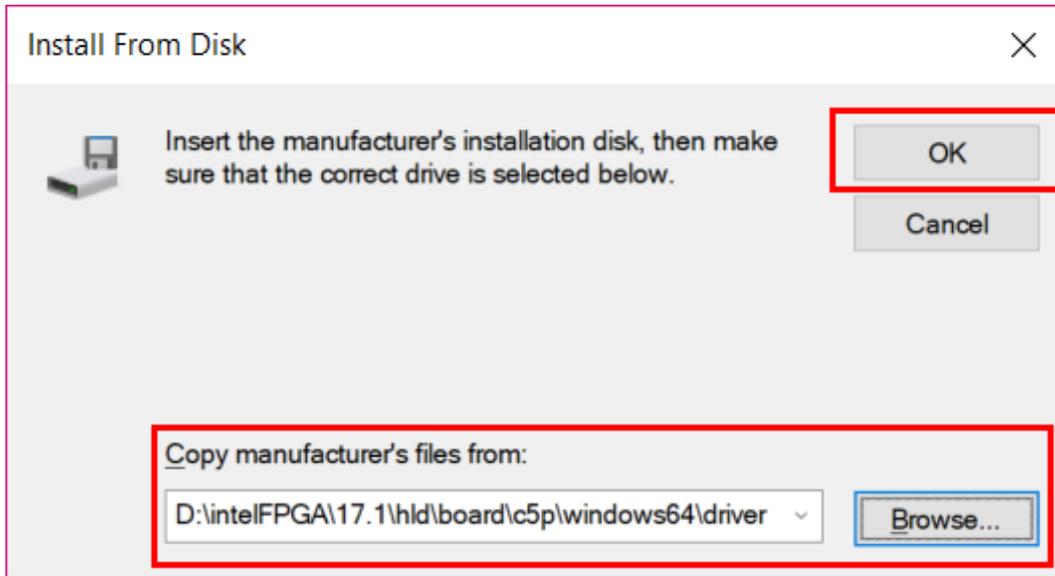
3. Continue choose “Next”.



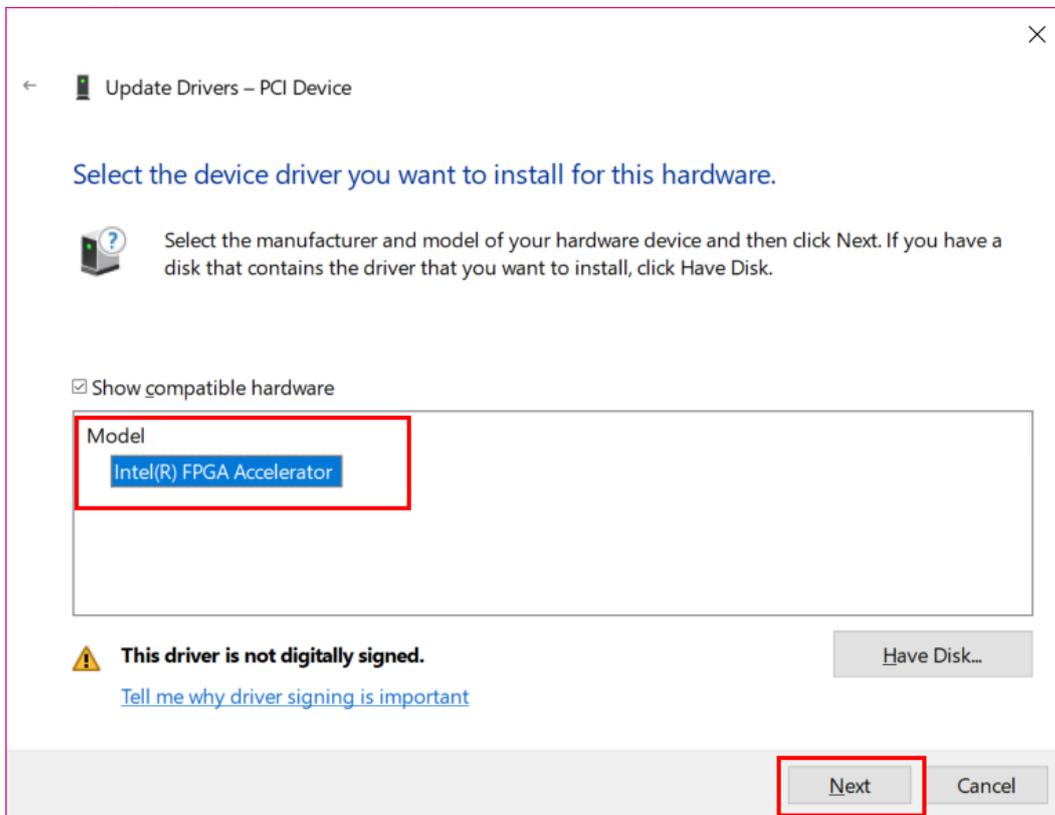
4. Choose **“Have Disk ...”**



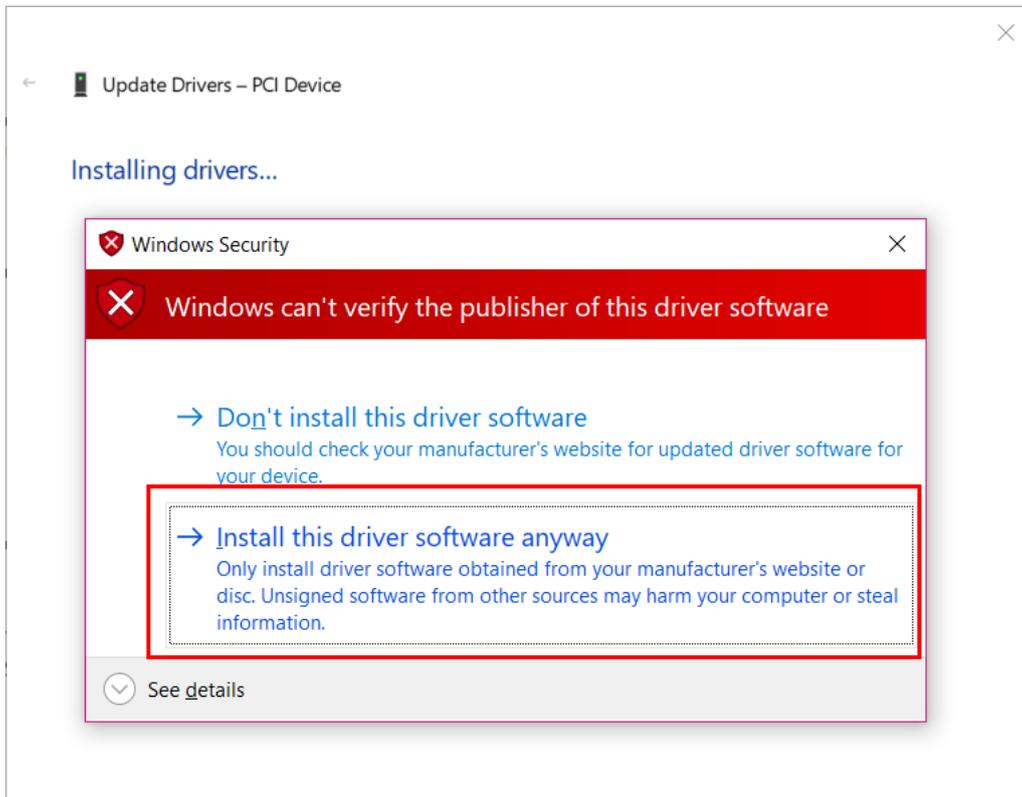
5. Click **Browse**, locate BSP to `osk/windows64/driver/acl_boards.inf`, Click **“OK”**.



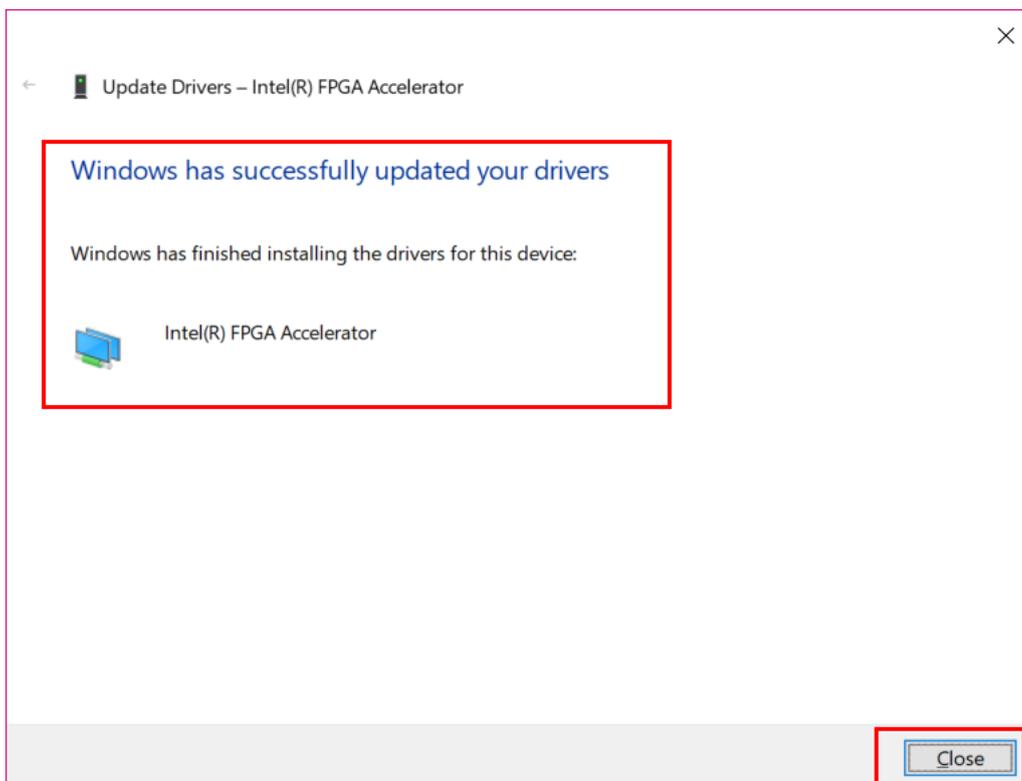
6. Select “**Intel(R) FPGA Accelerator**”, Click “**Next**” to continue the installation.



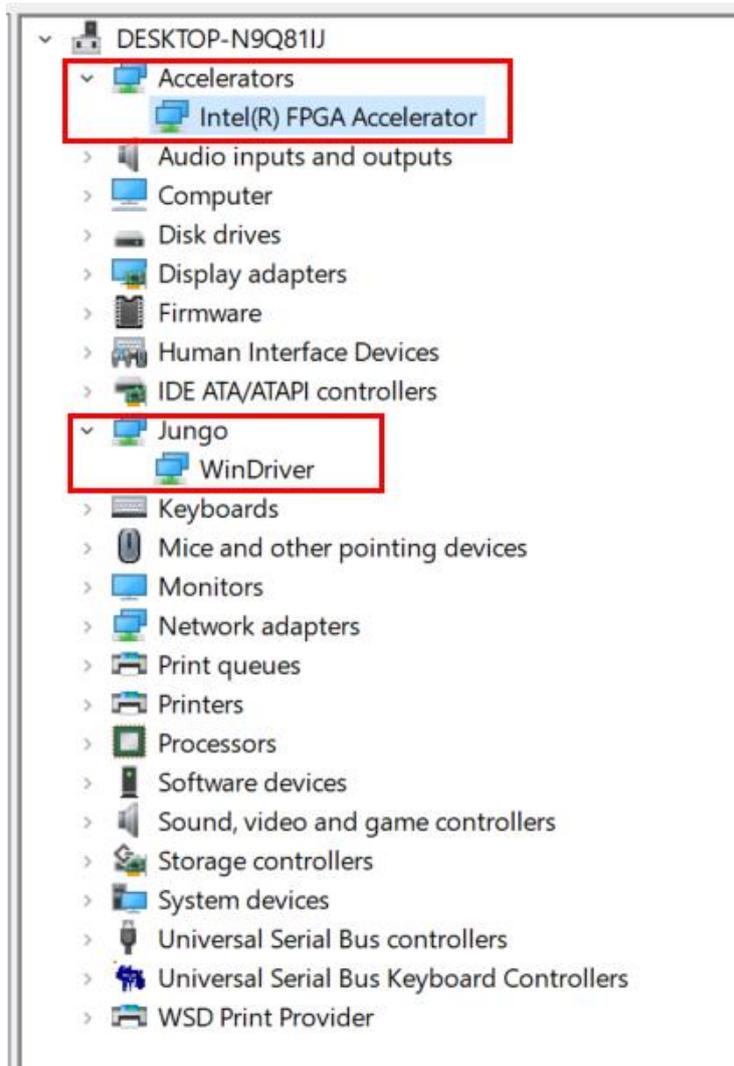
7. Select “**Install this driver software anyway**” in the pop-up “**Windows Security**” window.



8. Installation complete.



9. In the device manager, the Jungo Windriver and the OSK PCIE driver are both installed successfully.



Revision History

<i>Version</i>	<i>Change Log</i>
V1.0	Initial Version

Copyright Statement

Copyright © 2018 Terasic Inc. All Rights Reserved.