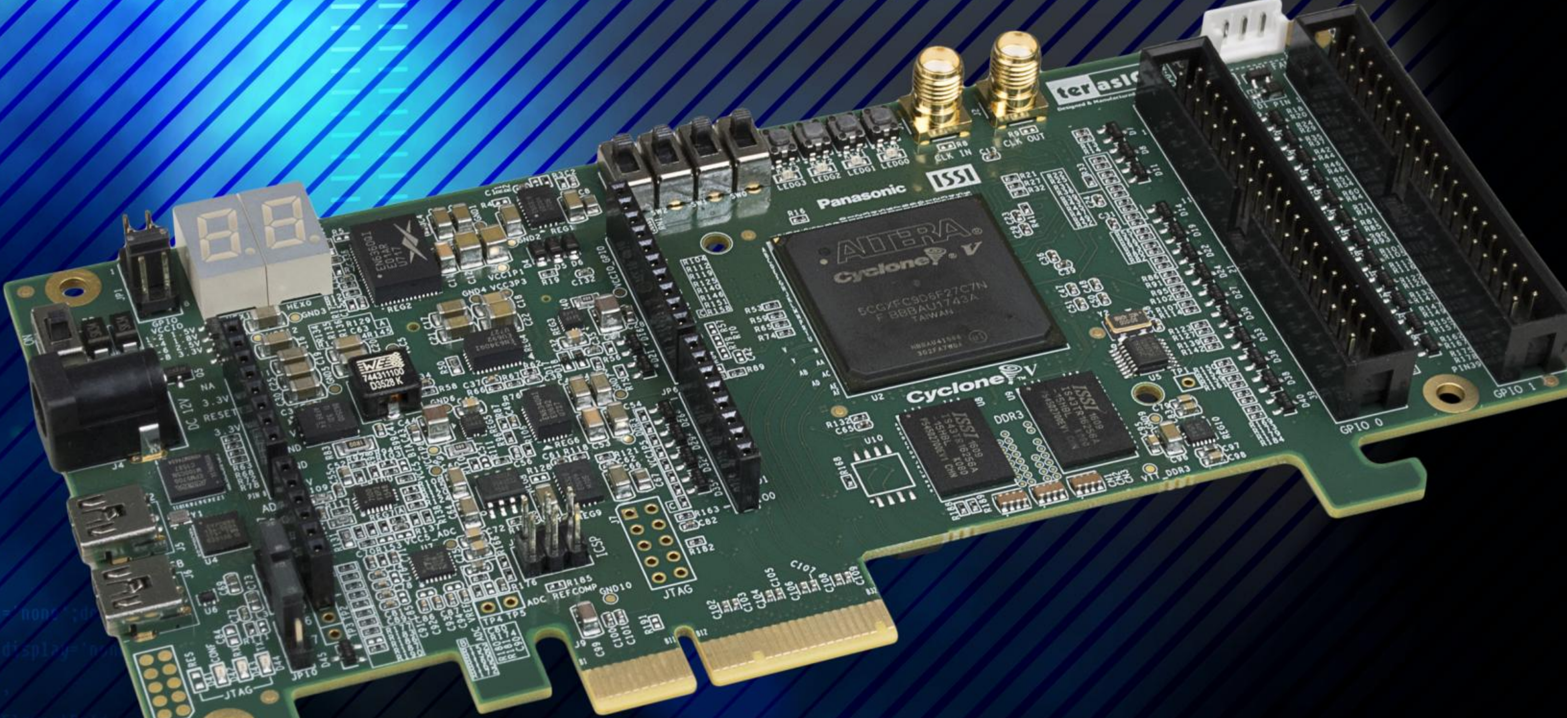
OpenCL

Starter Platform for OpenVINO™ Toolkit



Flash

PCI EXPRESS

DDR3

terasIC
www.terasic.com

CONTENTS

Chapter 1

TSP OpenCL

Terasic Starter Platform for OpenVINO™ Toolkit (TSP), 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 TSP. 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:

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

1.1 System Requirement

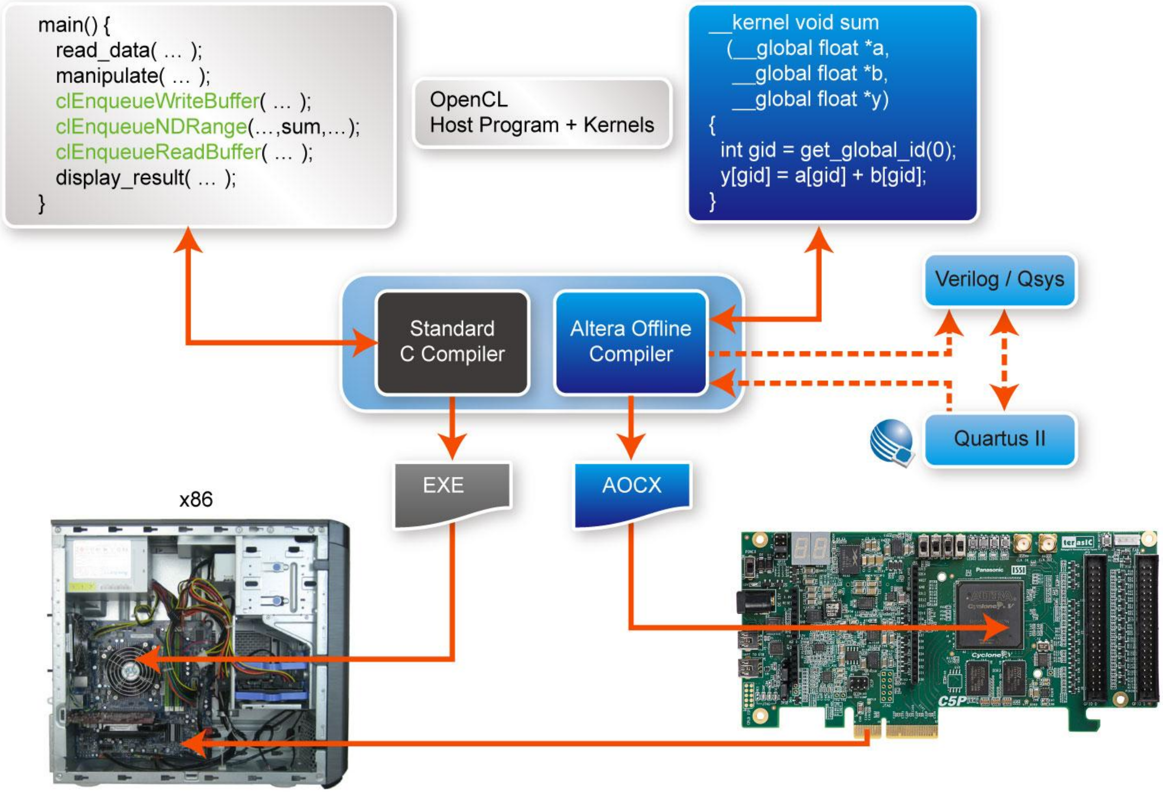
The following items are required to set up OpenCL for TSP board:

- Terasic Starter Platform for OpenVINO™ Toolkit (TSP)
- A Host PC with
  - USB Host Port
  - One PCI Express x4/x8/x16 slot
  - 16GB memory is recommended, 8GB is minimal
  - 12V Power for TSP
- 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**
- TSP 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.



```
main() {
   read_data( … );
   manipulate( … );
   clEnqueueWriteBuffer( … );
   clEnqueueNDRange(…,sum,…);
   clEnqueueReadBuffer( … );
   display_result( … );
}
```

```
__kernel void sum
   (__global float *a,
    __global float *b,
    __global float *y)
{
  int gid = get_global_id(0);
  y[gid] = a[gid] + b[gid];
}
```

Figure 1-1 Intel FPGA OpenCL Architecture

Chapter 2

OpenCL for Windows

This chapter describes how to set up TSP 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.

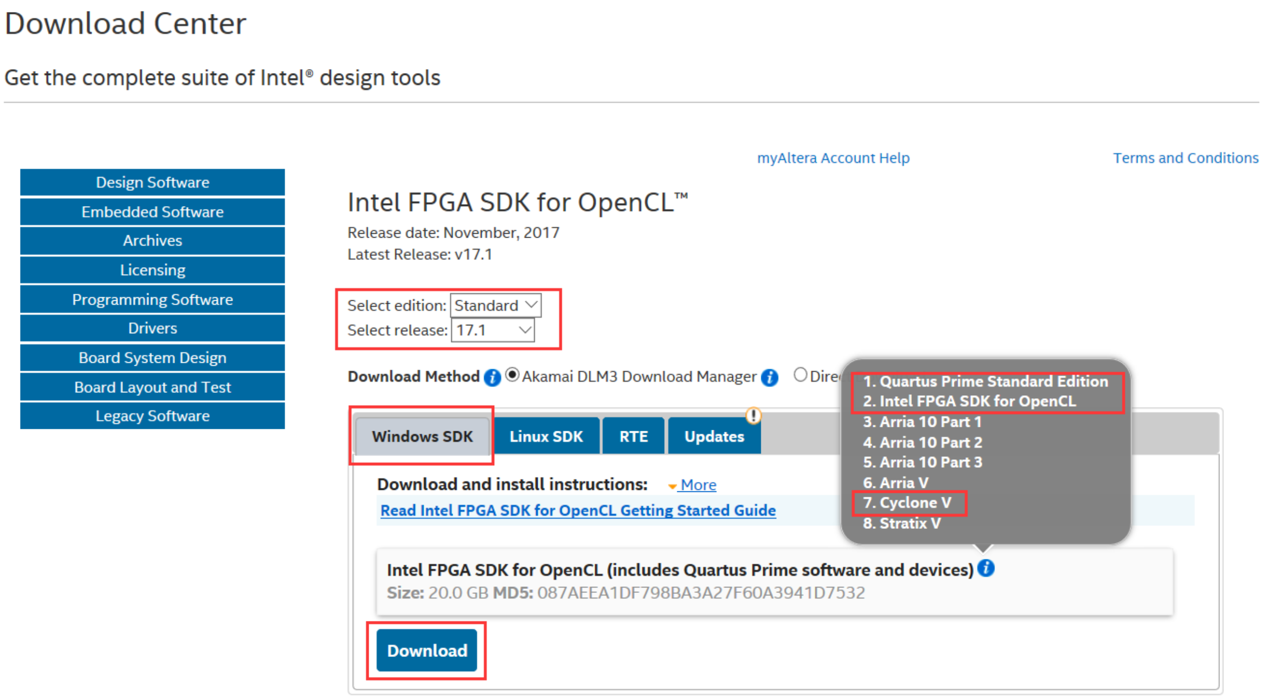
■ Quartus Prime and OpenCL SDK

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.



Download Center

Get the complete suite of Intel® design tools

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 ○ Dire

Download and install instructions: More

Read Intel FPGA SDK for OpenCL Getting Started Guide

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

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

Size: 20.0 GB MD5: 087AEEA1DF798BA3A27F60A3941D7532

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:

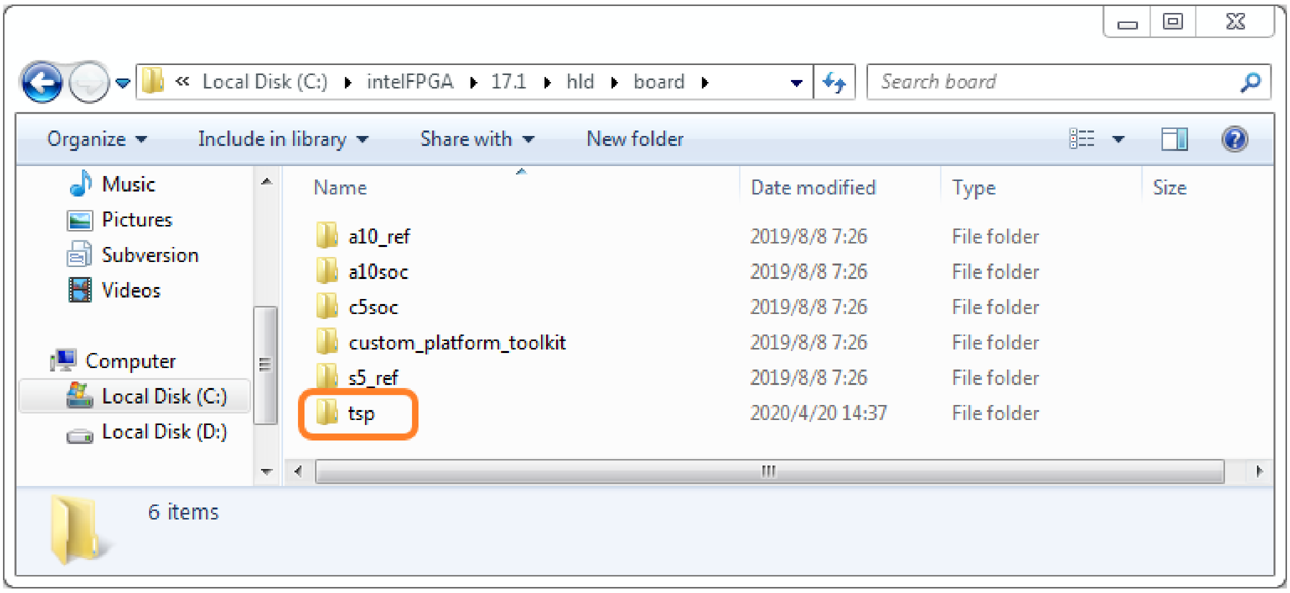
https://www.visualstudio.com/vs/older-downloads/

■ TSP OpenCL BSP (Board Support Package)

After Quartus Prime and OpenCL SDK are installed, download the windows BSP for Intel FPGA OpenCL 17.1(TSP _OpenCL_BSP_17.1.zip) from the web:

http://tsp.terasic.com/cd

Then, decompress TSP_OpenCL_BSP_17.1.zip to the "**tsp**" 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".



Name	Date modified	Type	Size
a10_ref	2019/8/8 7:26	File folder	
a10soc	2019/8/8 7:26	File folder	
c5soc	2019/8/8 7:26	File folder	
custom_platform_toolkit	2019/8/8 7:26	File folder	
s5_ref	2019/8/8 7:26	File folder	
tsp	2020/4/20 14:37	File folder	

Figure 2-2 TSP OpenCL BSP Content

For more details about TSP 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 TSP 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 TSP or OpenCL User

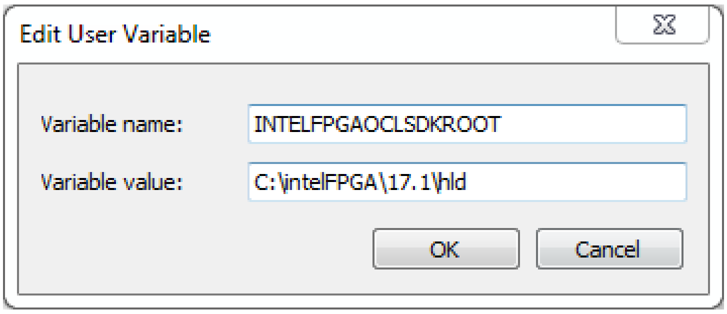
2.2 Environment Configure

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


Edit User Variable

Variable name: INTELFPGAOCLSDKROOT

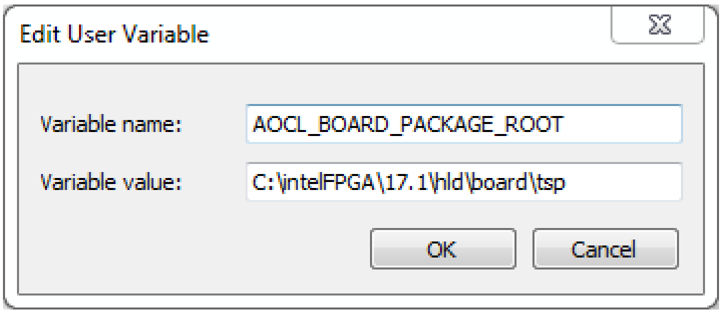
Variable value: C:\intelFPGA\17.1\hld

Figure 2-3 Edit ALTERAOCLSDKROOT Environment Variable

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

"%INTELFPGAOCLSDKROOT%\board\tsp"

as shown in **Figure 2-4**.

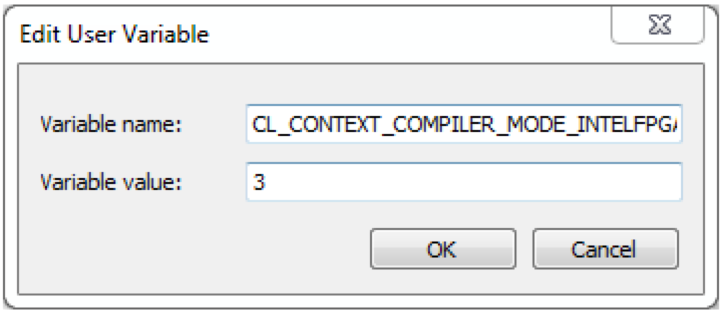

Edit User Variable

Variable name: AOCL_BOARD_PACKAGE_ROOT

Variable value: C:\intelFPGA\17.1\hld\board\tsp

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.**



Variable name: CL_CONTEXT_COMPILER_MODE_INTELFPG/

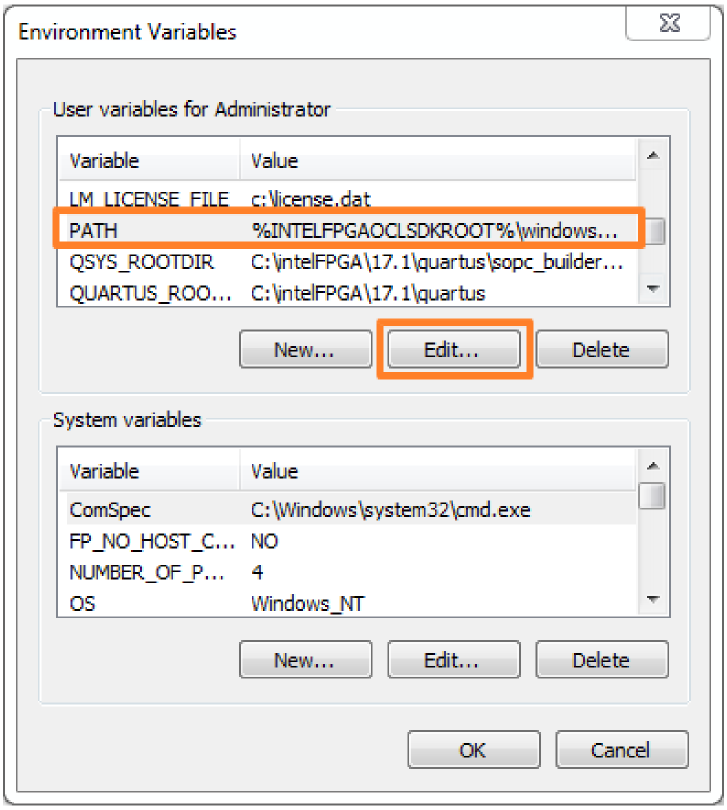
Variable value: 3

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 TSP BSP as shown in **Figure 2-6** and **Figure 2-7**.



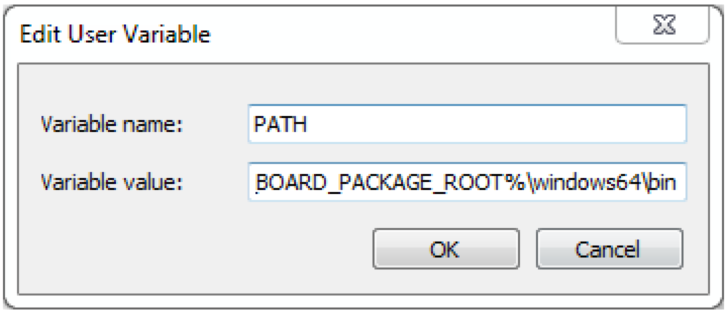
User variables for Administrator

Variable	Value
LM_LICENSE_FILE	c:\license.dat
PATH	%INTELFPGAOCLSDKROOT%\windows...
QSYS_ROOTDIR	C:\intelFPGA\17.1\quartus\sopc_builder...
QUARTUS_ROO...	C:\intelFPGA\17.1\quartus

System variables

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	4
OS	Windows_NT

Figure 2-6 Select "PATH" and click "Edit"



Variable name: PATH

Variable value: BOARD_PACKAGE_ROOT%\windows64\bin

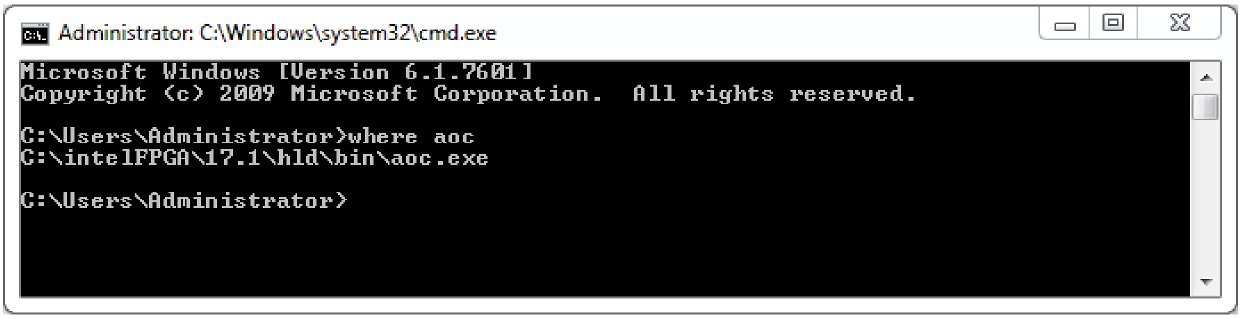
Figure 2-7 Edit PATH environment variable

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.**

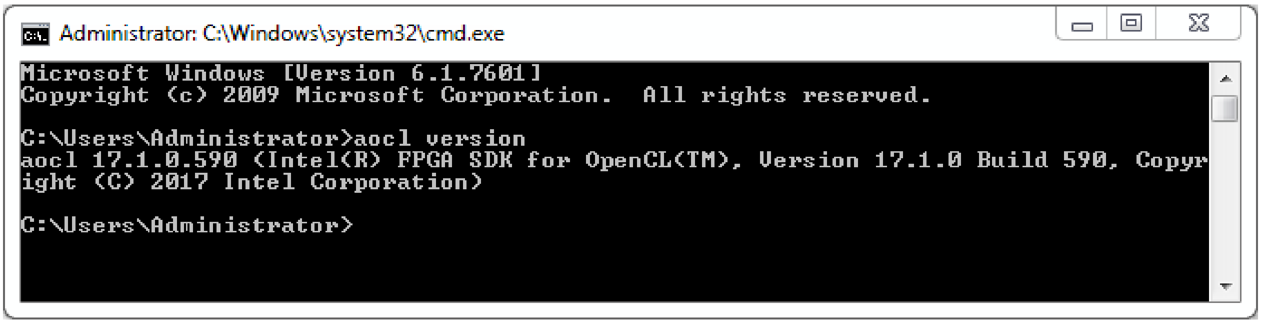


```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

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**.

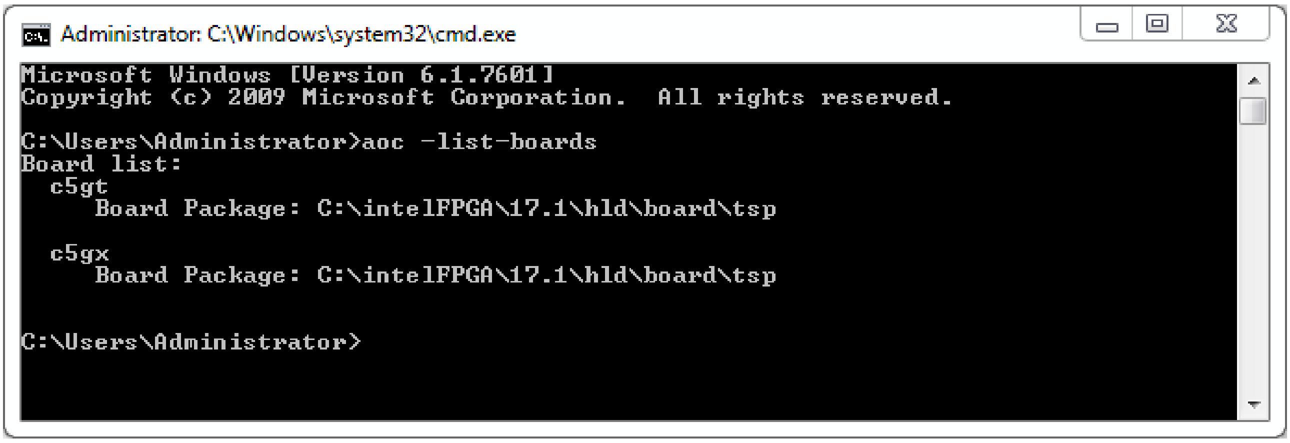


```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Administrator>aocl version
aocl 17.1.0.590 (Intel(R) FPGA SDK for OpenCL(TM), Version 17.1.0 Build 590, Copyr
ight (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 "**tsp**" is listed in **Board list** as shown in **Figure 2-10**.



```
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\tsp

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


C:\Users\Administrator>
```

Figure 2-10 ‘tsp’ 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 TSP, please follow the below procedures to set up TSP board on your PC as shown in Figure 2-11.

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

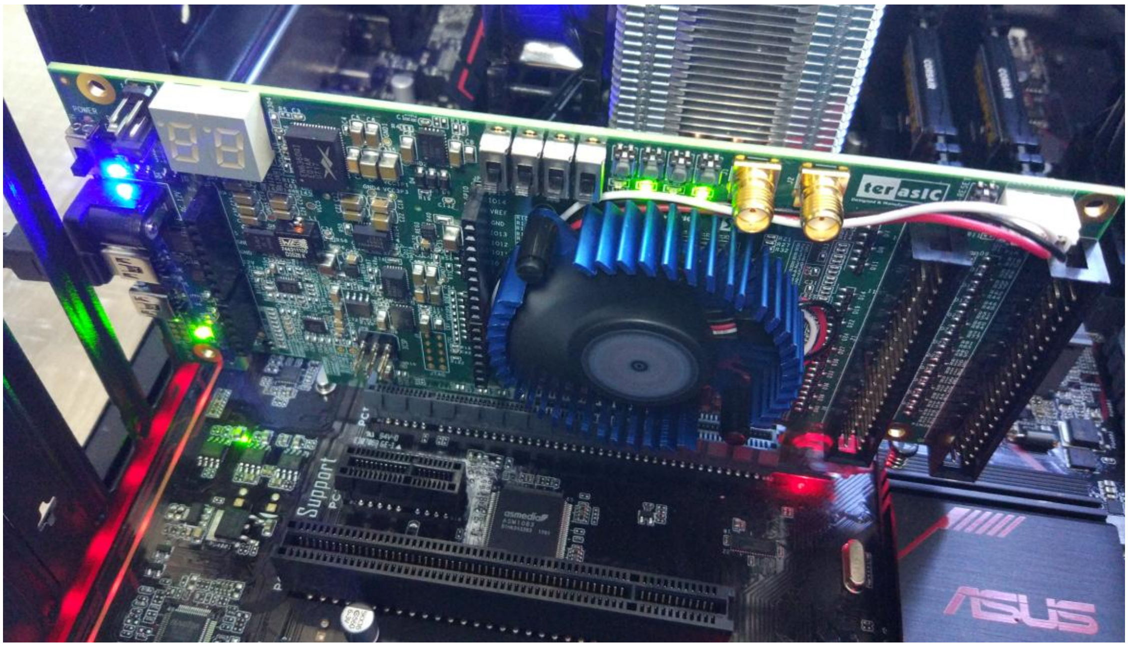


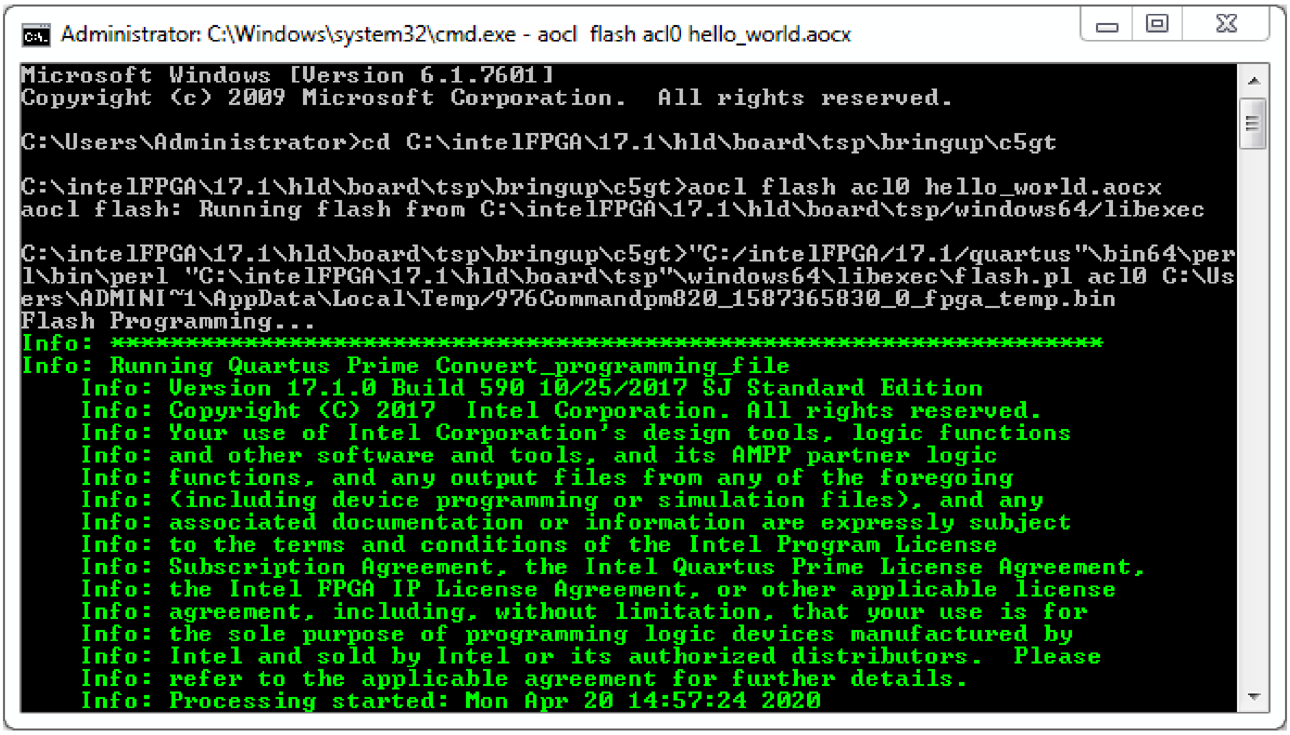
Figure 2-11 Setup TSP board on PC

■ 'aocl flash' program the image into FLASH

The **flash** utility in the TSP OpenCL BSP 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\tsp\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 TSP. It will take about 5 minutes for flash programming as shown in Figure 2-12.


Administrator: C:\Windows\system32\cmd.exe - aocl flash acl0 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\tsp\bringup\c5gt

C:\intelFPGA\17.1\hld\board\tsp\bringup\c5gt>aocl flash acl0 hello_world.aocx
aocl flash: Running flash from C:\intelFPGA\17.1\hld\board\tsp/windows64/libexec

C:\intelFPGA\17.1\hld\board\tsp\bringup\c5gt>"C:/intelFPGA/17.1/quartus"\bin64\per
l\bin\perl "C:\intelFPGA\17.1\hld\board\tsp"\windows64\libexec\flash.pl acl0 C:\Us
ers\ADMINI~1\AppData\Local\Temp/976Commandpm820_1587365830_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: Mon Apr 20 14:57:24 2020
```

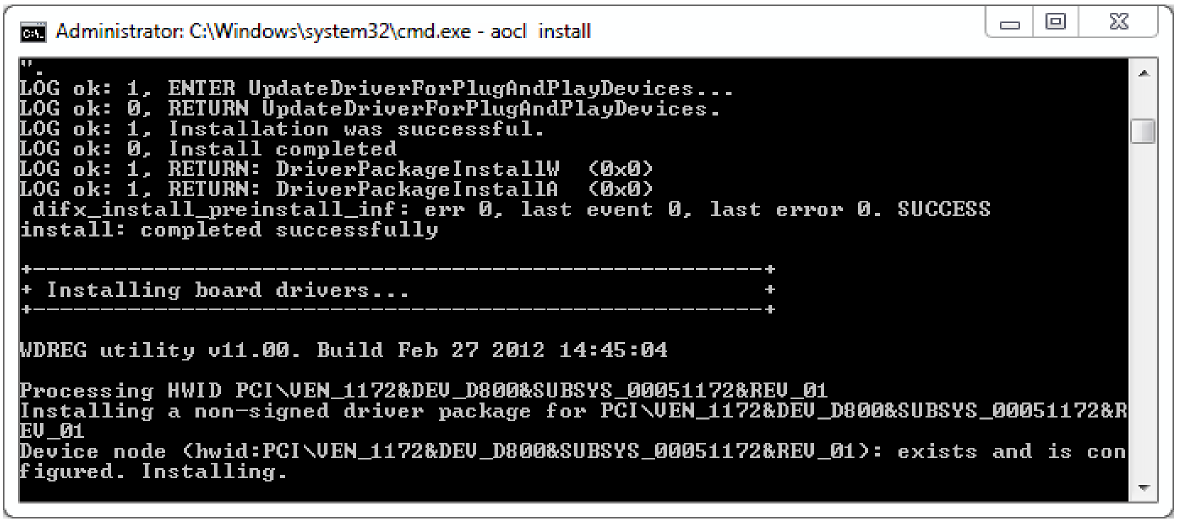
Figure 2-12 aocl flash acl0 hello_world.aocx"

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

■ Driver Installation

The **install** utility in the TSP OpenCL BSP 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.

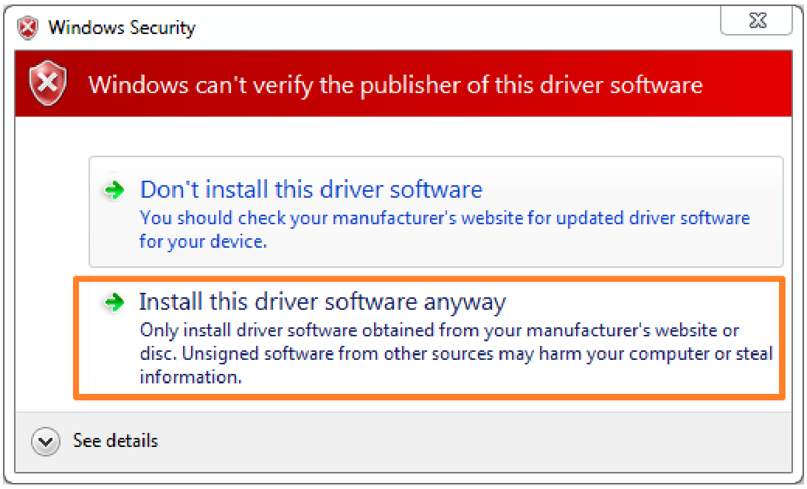
In **Command Prompt** window, type "**aocl install**" to install the driver as shown in Figure 2-13. Note that users need to have administrator privileges to install the driver.



```
".
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&R
EV_01
Device node (hwid:PCI\VEN_1172&DEV_D800&SUBSYS_00051172&REV_01): exists and is con
figured. 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**.



Windows Security

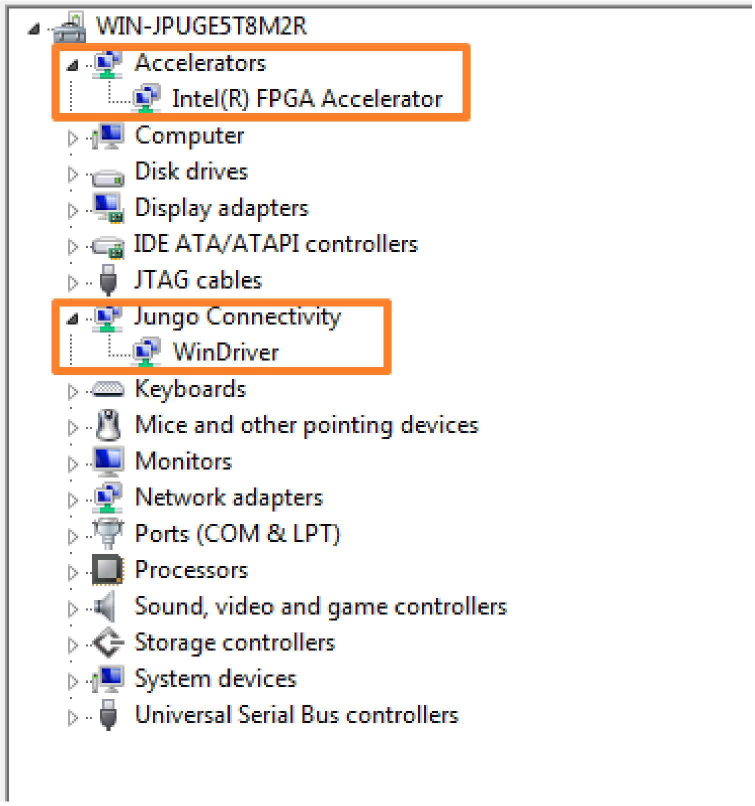
Windows can't verify the publisher of this driver software

Don't install this driver software
You should check your manufacturer's website for updated driver software for your device.

Install this driver software anyway
Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information.

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**.



- WIN-JPUGE5T8M2R
  - Accelerators
    - Intel(R) FPGA Accelerator
  - Computer
  - Disk drives
  - Display adapters
  - IDE ATA/ATAPI controllers
  - JTAG cables
  - Jungo Connectivity
    - WinDriver
  - Keyboards
  - Mice and other pointing devices
  - Monitors
  - Network adapters
  - Ports (COM & LPT)
  - Processors
  - Sound, video and game controllers
  - Storage controllers
  - System devices
  - Universal Serial Bus controllers

Figure 2-15 driver installation success

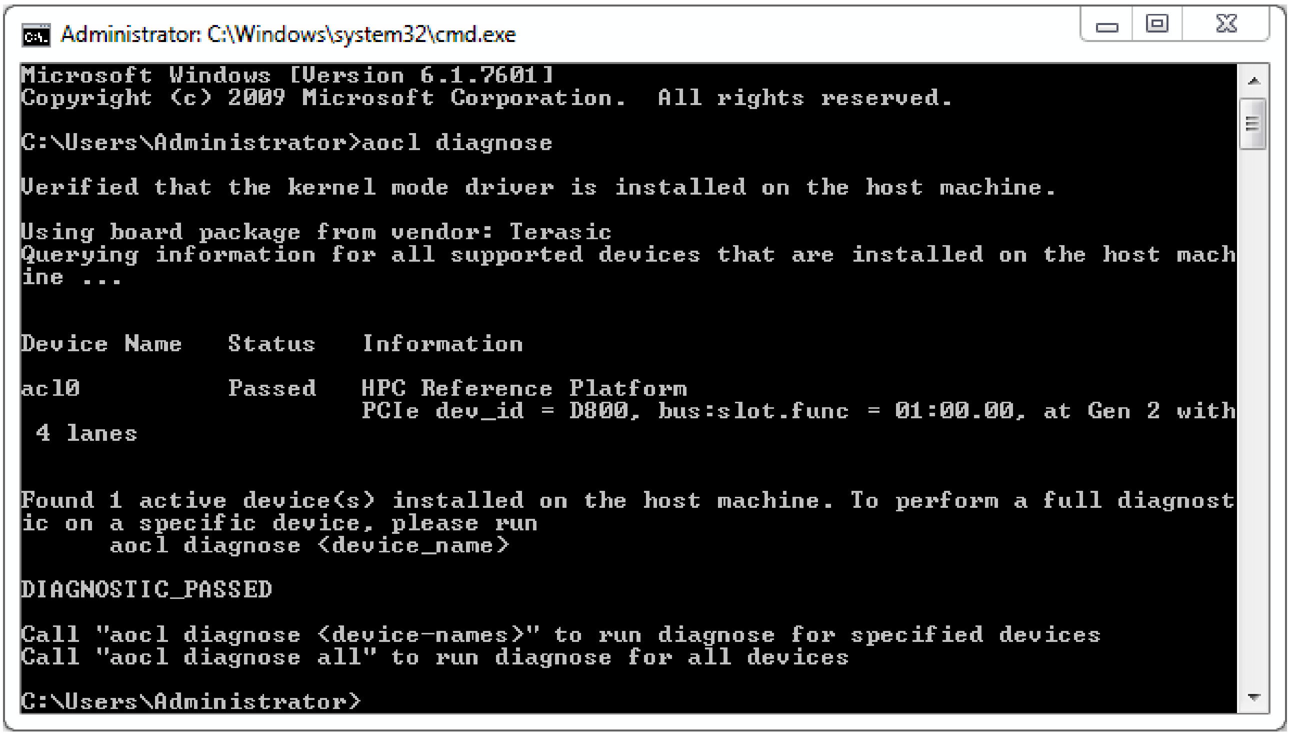
For driver installation on **Windows10** x64 OS, please refer to the **Appendix** of Chapter 4.

2.5 OpenCL Runtime Verify

■ Test ‘aocl diagnose’ Command

The **diagnose** utility in the TSP 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.



```
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 mach
ine ...


Device Name   Status   Information

acl0          Passed   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 diagnost
ic 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>
```

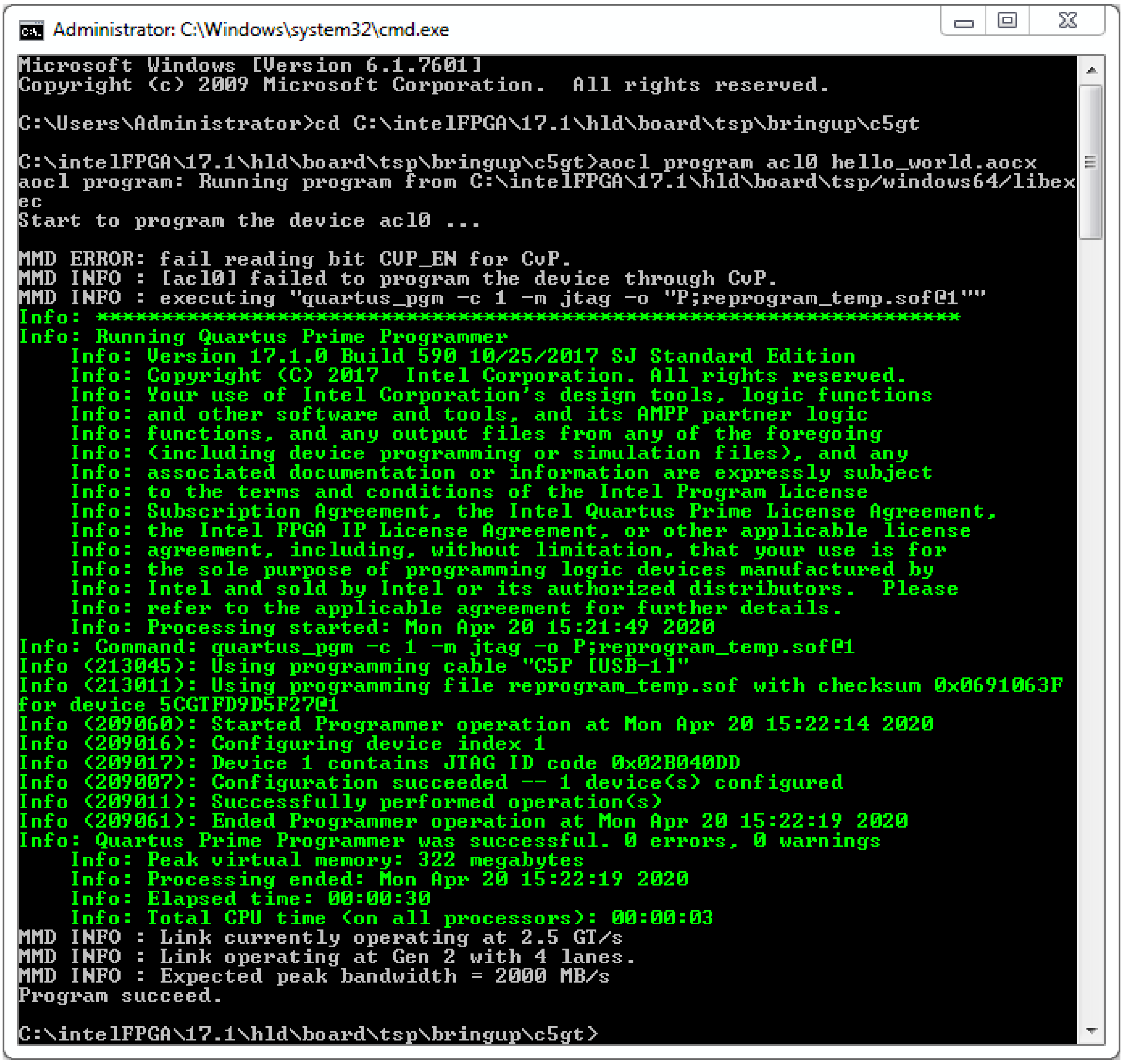
Figure 2-16 “aocl diagnose” messages

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 TSP OpenCL BSP 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\tsp\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.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

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

C:\intelFPGA\17.1\hld\board\tsp\bringup\c5gt>aocl program acl0 hello_world.aocx
aocl program: Running program from C:\intelFPGA\17.1\hld\board\tsp/windows64/libex
ec
Start to program the device acl0 ...

MMD ERROR: fail reading bit CVP_EN for CvP.
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: Mon Apr 20 15:21:49 2020
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 5CGTFD9D5F27@1
Info (209060): Started Programmer operation at Mon Apr 20 15:22:14 2020
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 Mon Apr 20 15:22:19 2020
Info: Quartus Prime Programmer was successful. 0 errors, 0 warnings
    Info: Peak virtual memory: 322 megabytes
    Info: Processing ended: Mon Apr 20 15:22:19 2020
    Info: Elapsed time: 00:00:30
    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\tsp\bringup\c5gt>
```

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

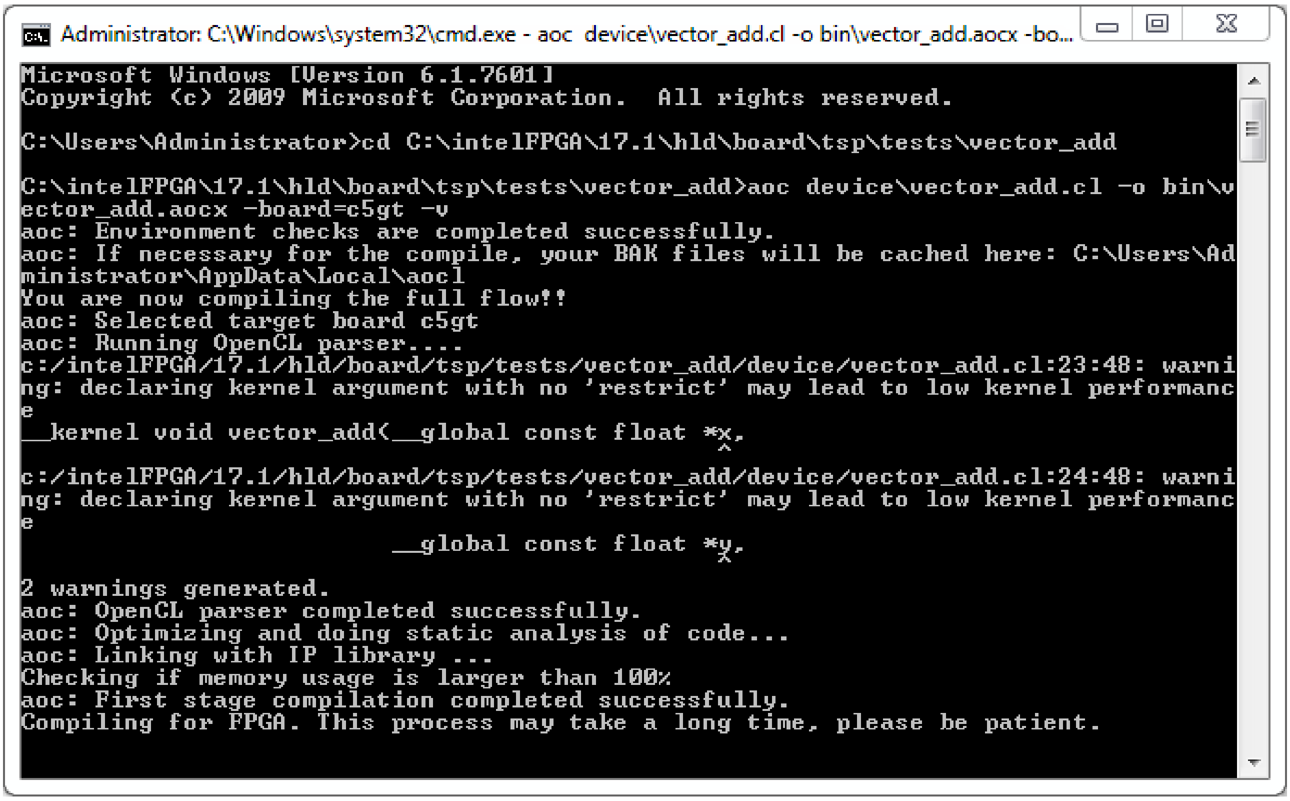
2.6 Compile and Test OpenCL Project

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 TSP.

■ 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\tsp\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

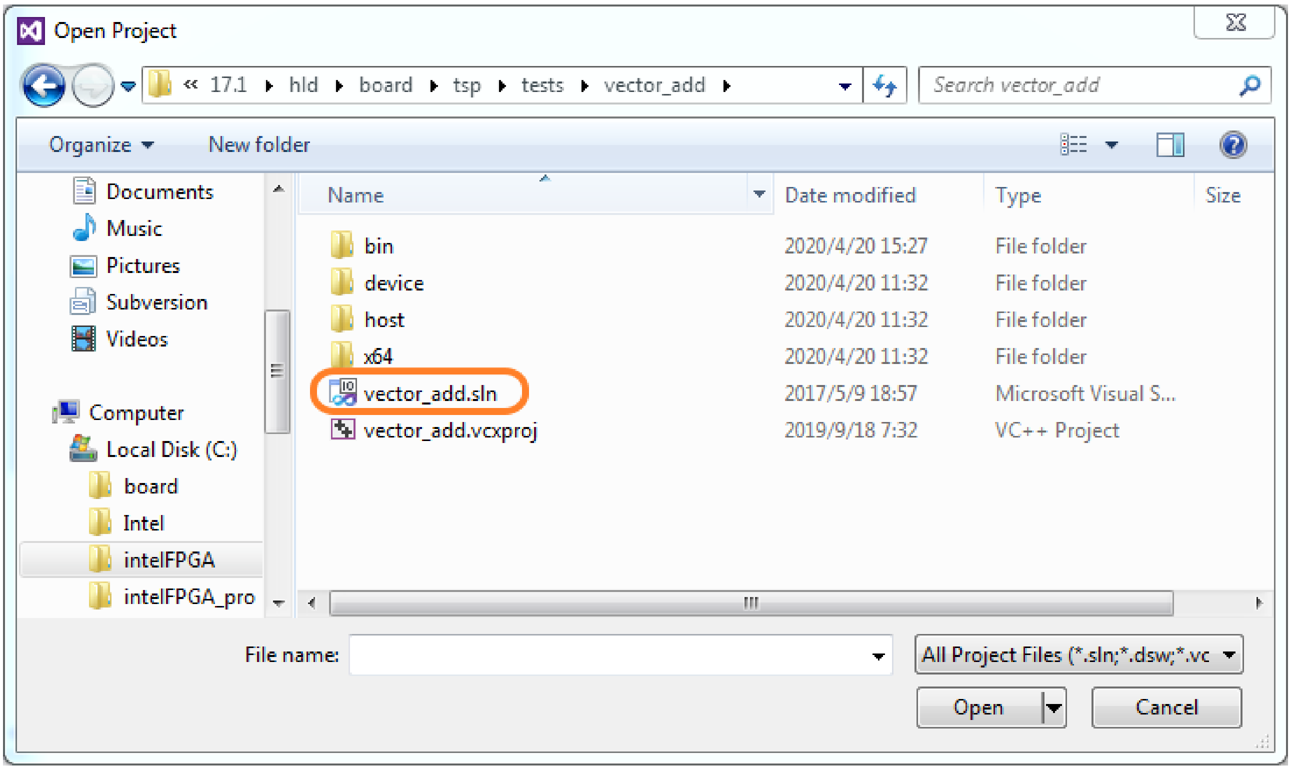

Administrator: C:\Windows\system32\cmd.exe - aoc device\vector_add.cl -o bin\vector_add.aocx -bo...

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

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

C:\intelFPGA\17.1\hld\board\tsp\tests\vector_add>aoc device\vector_add.cl -o bin\v
ector_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\Ad
ministrator\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/tsp/tests/vector_add/device/vector_add.cl:23:48: warni
ng: declaring kernel argument with no 'restrict' may lead to low kernel performanc
e
__kernel void vector_add(__global const float *x,
                                               ^
c:/intelFPGA/17.1/hld/board/tsp/tests/vector_add/device/vector_add.cl:24:48: warni
ng: declaring kernel argument with no 'restrict' may lead to low kernel performanc
e
                         __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\tsp\tests\vector_add”, and select “vector_add.sln” as shown **Figure 2-19**.

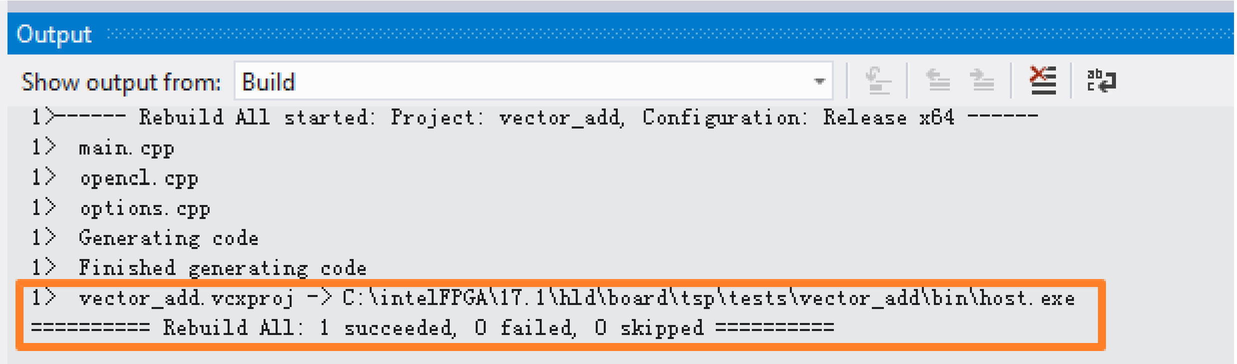


Name	Date modified	Type	Size
bin	2020/4/20 15:27	File folder	
device	2020/4/20 11:32	File folder	
host	2020/4/20 11:32	File folder	
x64	2020/4/20 11:32	File folder	
vector_add.sln	2017/5/9 18:57	Microsoft Visual S...	
vector_add.vcxproj	2019/9/18 7:32	VC++ Project	

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\tsp\tests\vector_add\bin\host.exe”



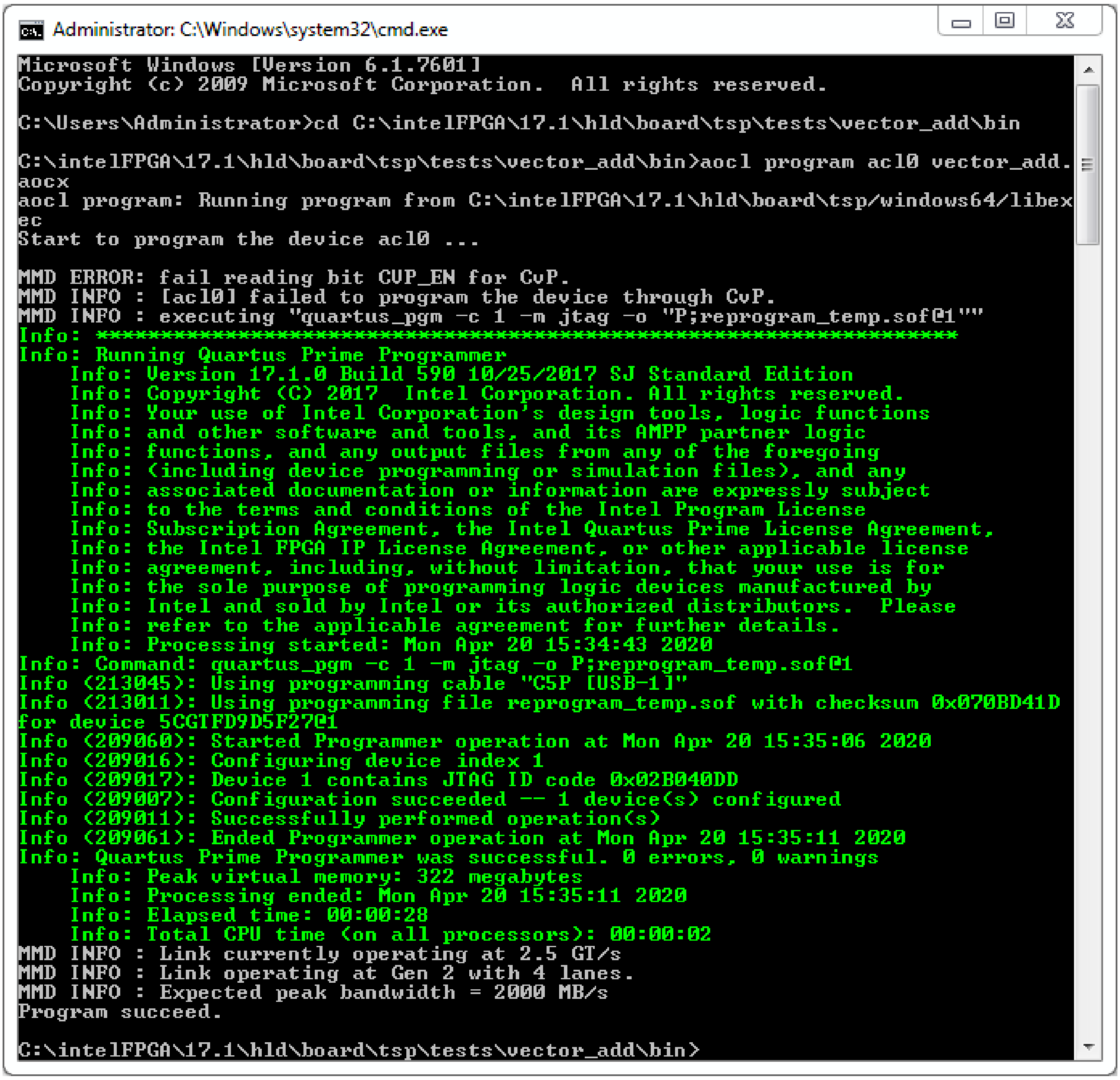
```
1>------ Rebuild All started: Project: vector_add, Configuration: Release x64 ------
1>  main.cpp
1>  opencl.cpp
1>  options.cpp
1>  Generating code
1>  Finished generating code
1>  vector_add.vcxproj -> C:\intelFPGA\17.1\hld\board\tsp\tests\vector_add\bin\host.exe
========== Rebuild All: 1 succeeded, 0 failed, 0 skipped ==========
```

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\tsp\tests\vector_add\bin” to go to **vector_add\bin** project folder,

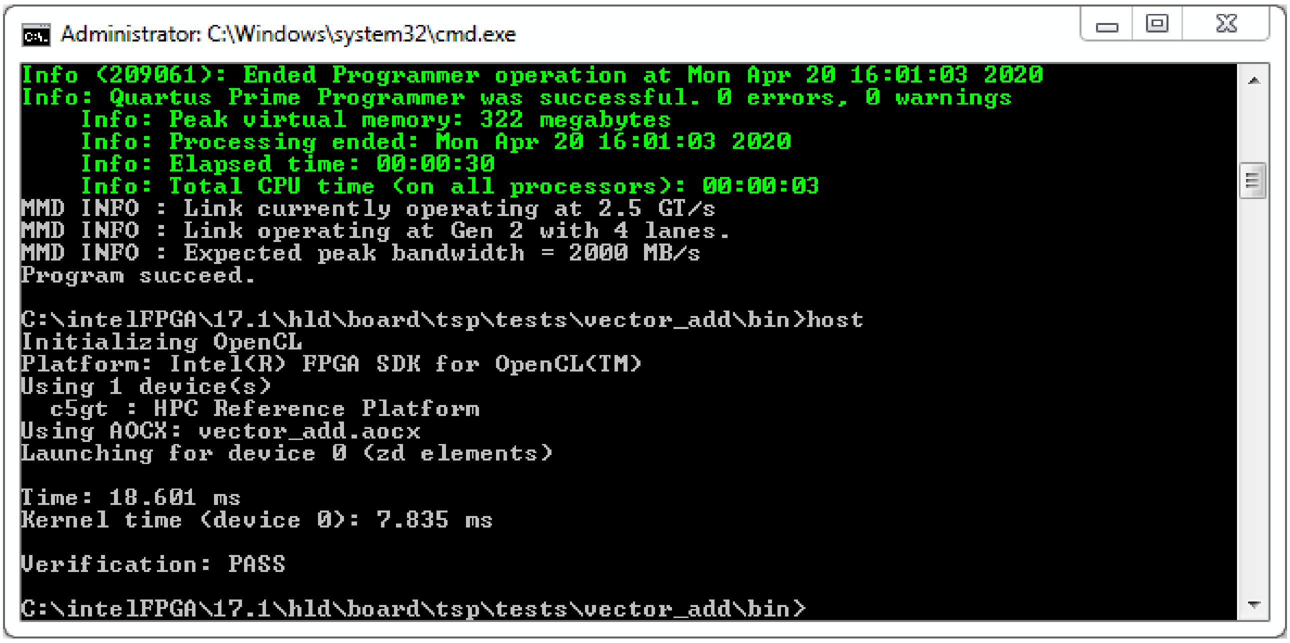
And type “**aocl program acl0 vector_add.aocx**” to program the bitstream into FPGA board as show in **Figure 2-21.**



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

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

C:\intelFPGA\17.1\hld\board\tsp\tests\vector_add\bin>aocl program acl0 vector_add.
aocx
aocl program: Running program from C:\intelFPGA\17.1\hld\board\tsp/windows64/libex
ec
Start to program the device acl0 ...

MMD ERROR: fail reading bit CVP_EN for CvP.
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: Mon Apr 20 15:34:43 2020
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 Mon Apr 20 15:35:06 2020
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 Mon Apr 20 15:35:11 2020
Info: Quartus Prime Programmer was successful. 0 errors, 0 warnings
    Info: Peak virtual memory: 322 megabytes
    Info: Processing ended: Mon Apr 20 15:35:11 2020
    Info: Elapsed time: 00:00:28
    Info: Total CPU time (on all processors): 00:00:02
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\tsp\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.



```
Info (209061): Ended Programmer operation at Mon Apr 20 16:01:03 2020
Info: Quartus Prime Programmer was successful. 0 errors, 0 warnings
    Info: Peak virtual memory: 322 megabytes
    Info: Processing ended: Mon Apr 20 16:01:03 2020
    Info: Elapsed time: 00:00:30
    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\tsp\tests\vector_add\bin>host
Initializing OpenCL
Platform: Intel(R) FPGA SDK for OpenCL(TM)
Using 1 device(s)
  c5gt : HPC Reference Platform
Using AOCX: vector_add.aocx
Launching for device 0 (zd elements)

Time: 18.601 ms
Kernel time (device 0): 7.835 ms

Verification: PASS

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

Figure 2-22 “vector_add” test successfully

Chapter 3

OpenCL for Linux

This chapter describes how to setup TSP 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 TSP. 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.

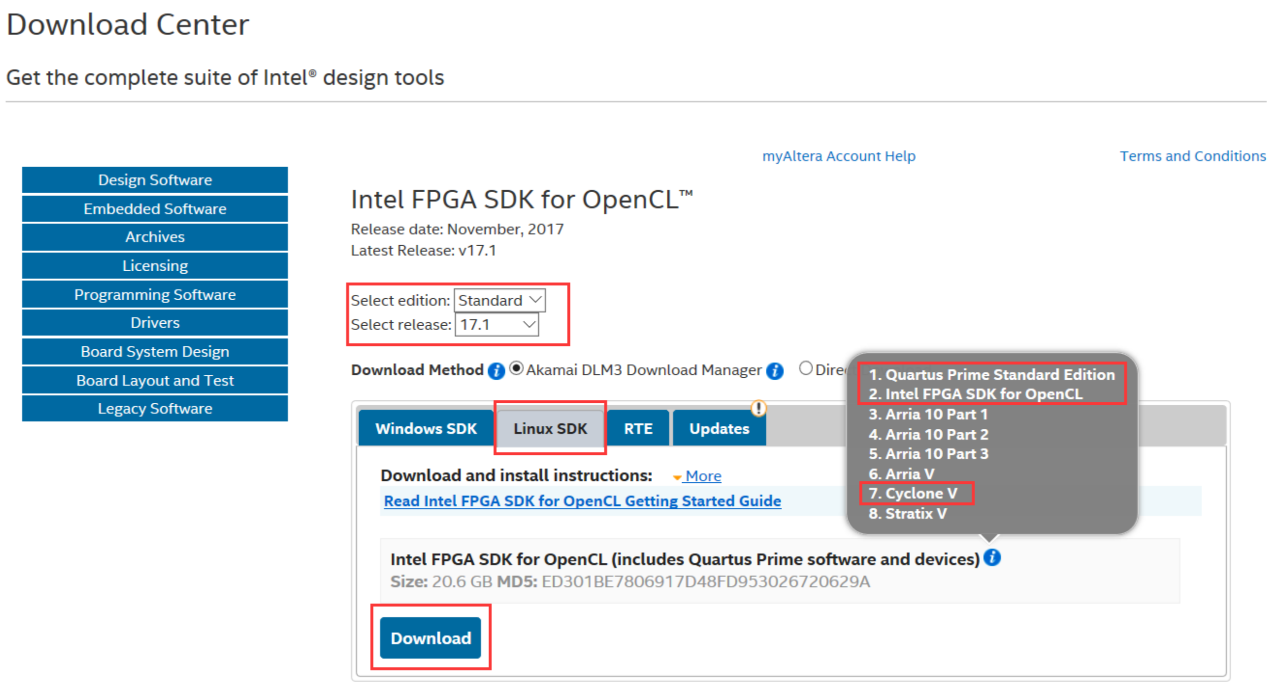
■ Intel Quartus Prime and OpenCL

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.



Download Center

Get the complete suite of Intel® design tools

myAltera Account Help

Terms and Conditions

- 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 ○ Dire

Windows SDK | Linux SDK | RTE | Updates

Download and install instructions: More

Read Intel FPGA SDK for OpenCL Getting Started Guide

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

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

Size: 20.6 GB **MD5:** ED301BE7806917D48FD953026720629A

Download

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 TSP. 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

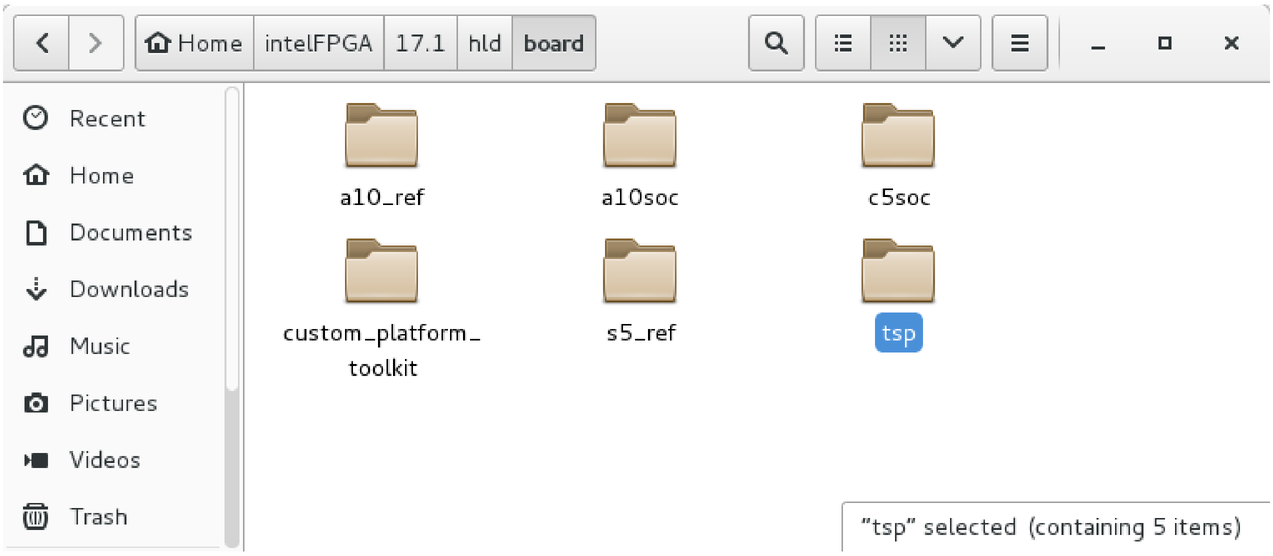
Note: To install the SDK on Linux, you must install it in a directory that you own (a directory which is not a system directory). You must also have sudo or root privileges.

■ TSP OpenCL BSP (Board Support Package)

After Quartus Prime and OpenCL SDK are installed, please download the TSP_OpenCL_BSP_17.1.tar.gz linux BSP for Intel FPGA OpenCL 17.1 from the web:

http://tsp.terasic.com/cd

Then, decompress TSP_OpenCL_BSP_17.1.tar.gz to the "**tsp**" 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**.



"tsp" selected (containing 5 items)

Figure 3-2 TSP OpenCL BSP Content

For more details about TSP 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 TSP 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 TSP 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 TSP correctly, the developers need to create an environment variable for the TSP board **AOCL_BOARD_PACKAGE_ROOT**, and set its value as:

"$INTELFPGAOCLSDKROOT"/board/tsp”

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.

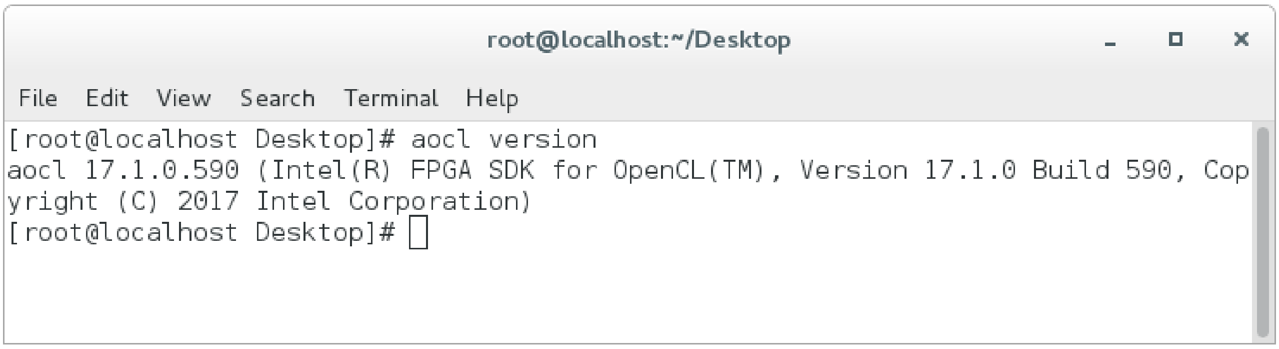
```
export  QUARTUS_ROOTDIR=/root/intelFPGA/17.1/quartus
export  INTELFPGAOCLSDKROOT=/root/  intelFPGA/17.1/hld
export AOCL_BOARD_PACKAGE_ROOT=/root/intelFPGA/17.1/hld/board/tsp
export PATH=$PATH:$INTELFPGAOCLSDKROOT/linux64/bin:$INTELFPGAOCLSDKROOT/bin:$INTELFPGAOCLSDKROOT/host/linux64/bin:$QUARTUS_ROOTDIR/bin
export  LD_LIBRARY_PATH=$AOCL_BOARD_PACKAGE_ROOT/tests/extlibs/lib:$INTELFPGAOCLSDKROOT/host/linux64/lib:$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**.

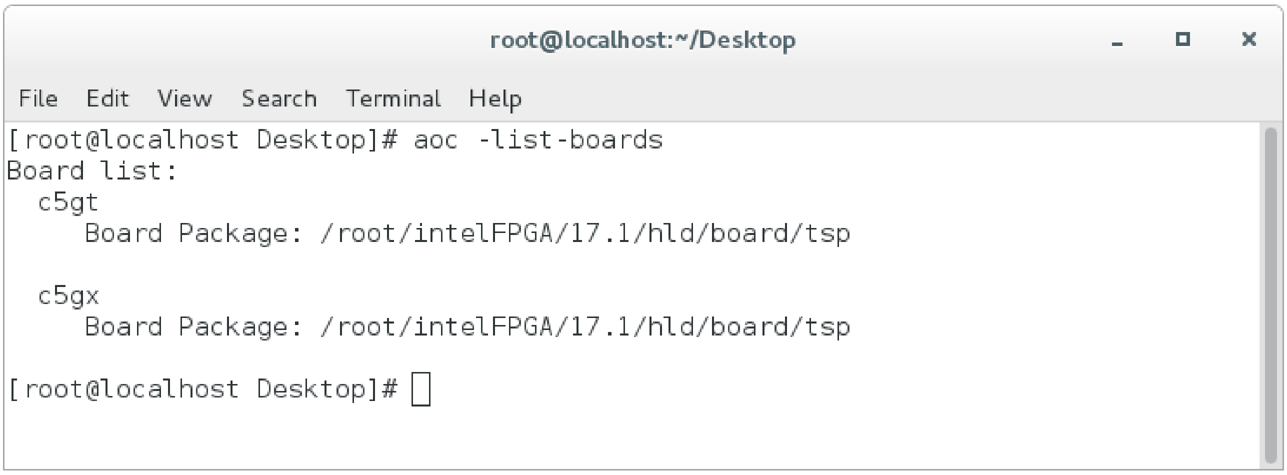


```
[root@localhost Desktop]# aocl version
aocl 17.1.0.590 (Intel(R) FPGA SDK for OpenCL(TM), Version 17.1.0 Build 590, Cop
yright (C) 2017 Intel Corporation)
[root@localhost Desktop]#
```

Figure 3-3 Version of OpenCL SDK

■ Target Board

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



```
[root@localhost Desktop]# aoc -list-boards
Board list:
  c5gt
     Board Package: /root/intelFPGA/17.1/hld/board/tsp

  c5gx
     Board Package: /root/intelFPGA/17.1/hld/board/tsp

[root@localhost Desktop]#
```

Figure 3-4 'tsp' 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 TSP, 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 TSP board into PCI Express x4/x8 or x16 slot.
3. Connect 12V power source to the TSP
4. Connect PC's USB port to TSP UB2 port using an USB cable.

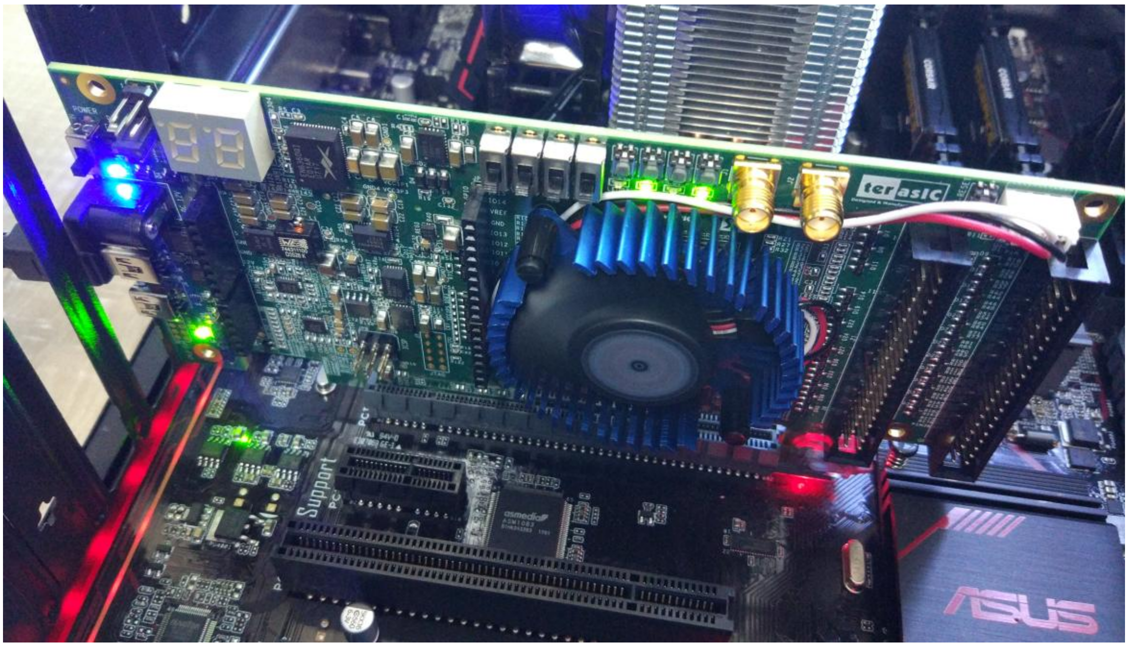
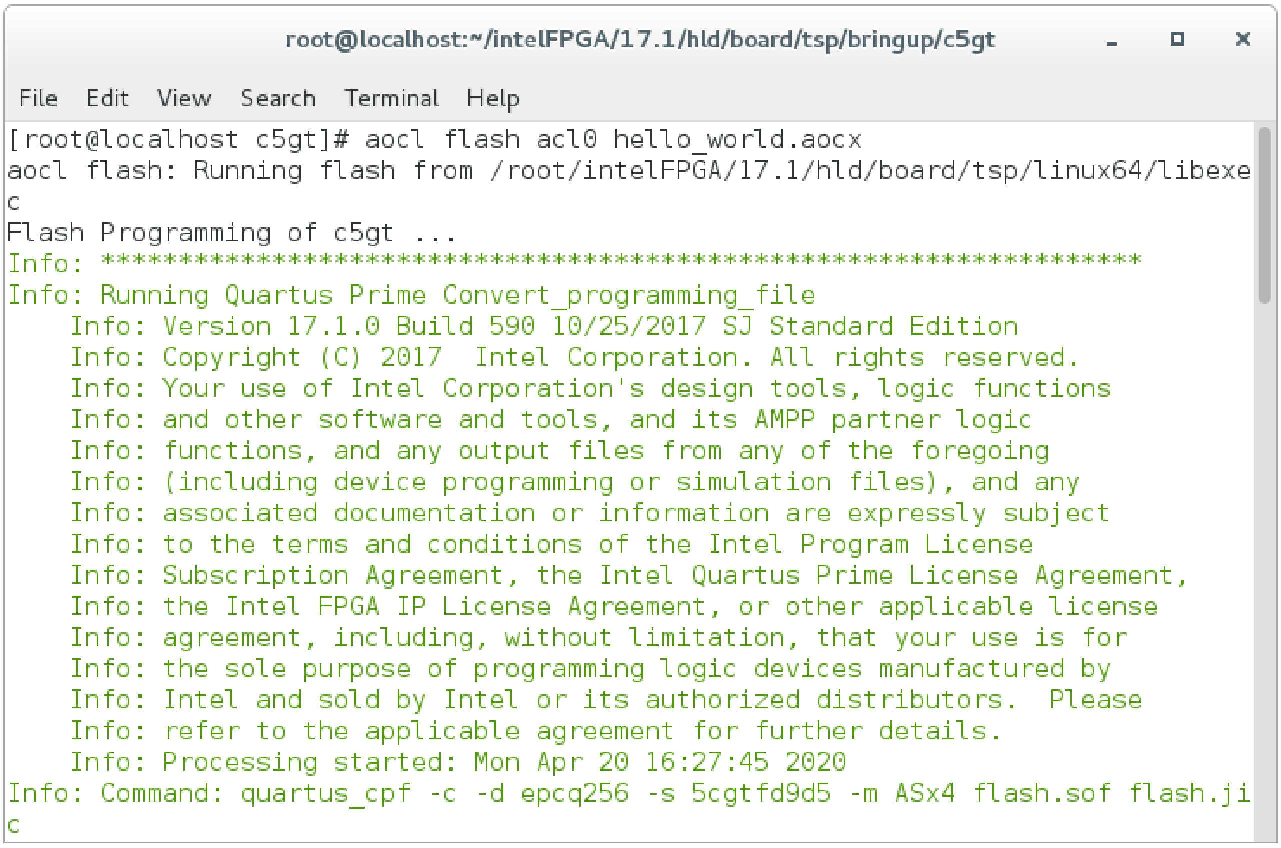


Figure 3-5 Setup TSP board on PC

■ 'aocl flash' program

The **flash** utility in the TSP OpenCL BSP 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/tsp/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 TSP. It will take about 5 minutes for flash programming as shown in **Figure 3-6**.


root@localhost:~/intelFPGA/17.1/hld/board/tsp/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/tsp/linux64/libexe
c
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: Mon Apr 20 16:27:45 2020
Info: Command: quartus_cpf -c -d epcq256 -s 5cgtfd9d5 -m ASx4 flash.sof flash.ji
c
```

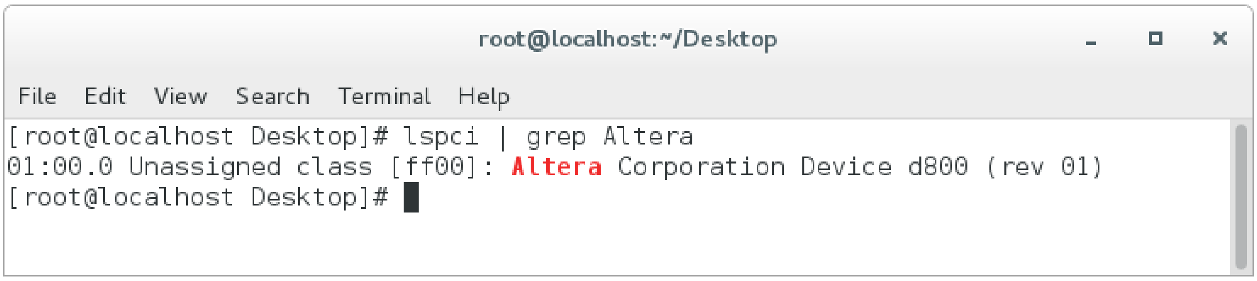
Figure 3-6 'aocl flash acl0 hello_world.aocx"

After flash programming is done successfully, **developers must power off TSP 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.

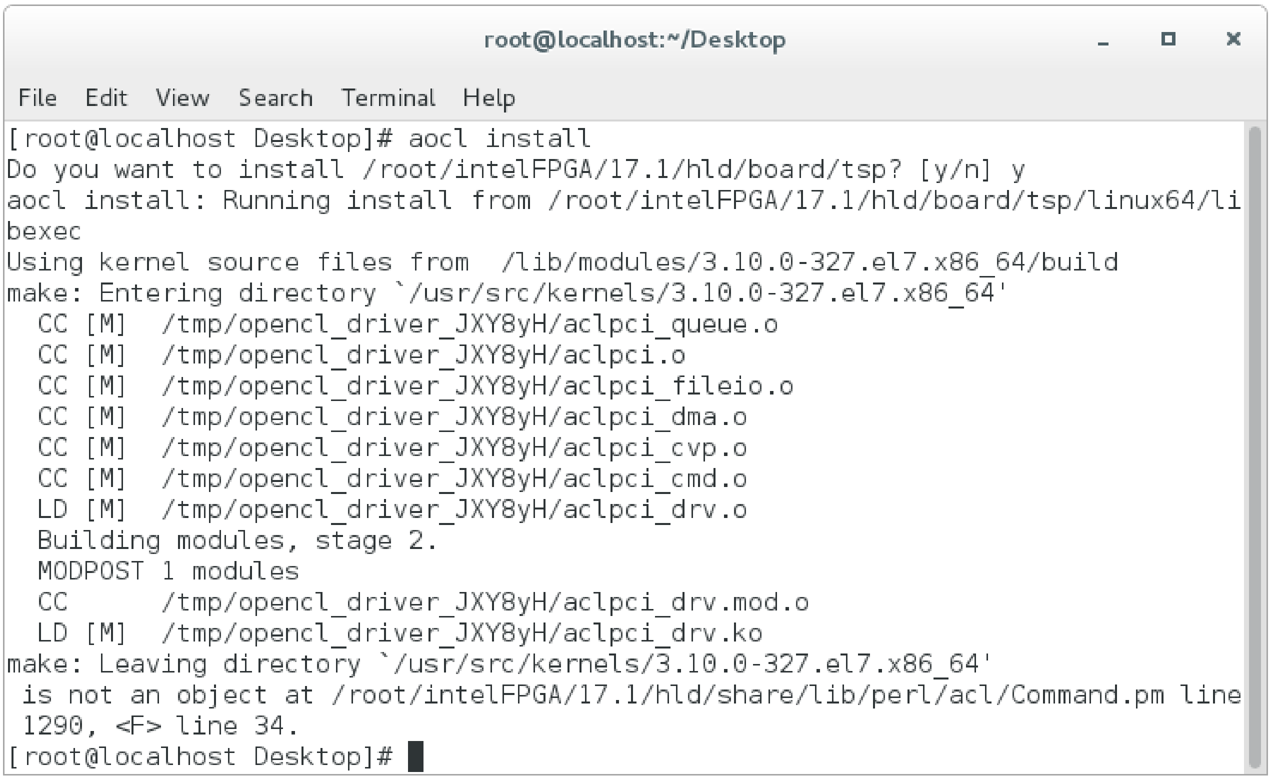
First, in the Linux **terminal**, type '**lspci |grep Altera**' to make sure the system recognizes the PCIe card as shown in Figure 3-7.



```
[root@localhost Desktop]# 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 Desktop]# aocl install
Do you want to install /root/intelFPGA/17.1/hld/board/tsp? [y/n] y
aocl install: Running install from /root/intelFPGA/17.1/hld/board/tsp/linux64/li
bexec
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/opencl_driver_JXY8yH/aclpci_queue.o
  CC [M]  /tmp/opencl_driver_JXY8yH/aclpci.o
  CC [M]  /tmp/opencl_driver_JXY8yH/aclpci_fileio.o
  CC [M]  /tmp/opencl_driver_JXY8yH/aclpci_dma.o
  CC [M]  /tmp/opencl_driver_JXY8yH/aclpci_cvp.o
  CC [M]  /tmp/opencl_driver_JXY8yH/aclpci_cmd.o
  LD [M]  /tmp/opencl_driver_JXY8yH/aclpci_drv.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /tmp/opencl_driver_JXY8yH/aclpci_drv.mod.o
  LD [M]  /tmp/opencl_driver_JXY8yH/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 Desktop]# 
```

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/tsp/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.

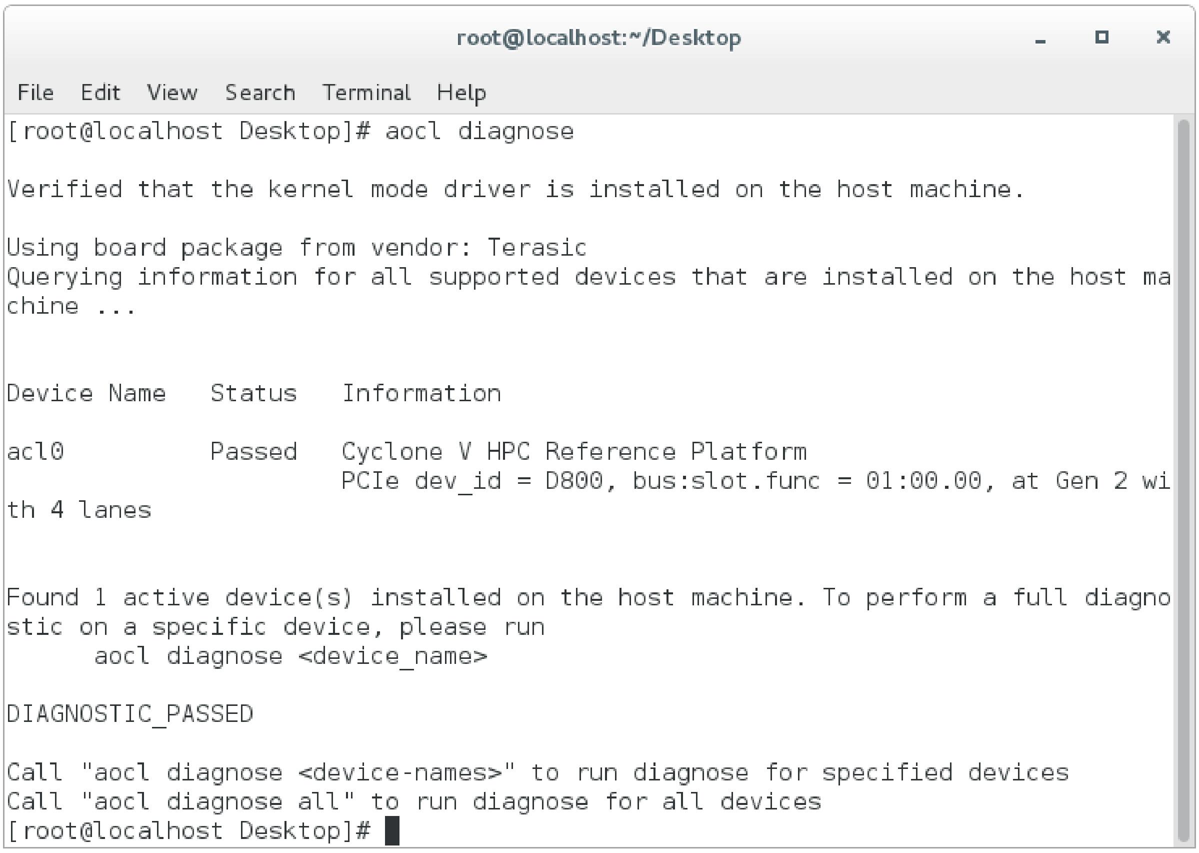
3.5 OpenCL Runtime Verify

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 TSP OpenCL BSP 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 Desktop]# 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 ma
chine ...


Device Name   Status   Information

acl0          Passed   Cyclone V HPC Reference Platform
                       PCIe dev_id = D800, bus:slot.func = 01:00.00, at Gen 2 wi
th 4 lanes


Found 1 active device(s) installed on the host machine. To perform a full diagno
stic 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 Desktop]# █
```

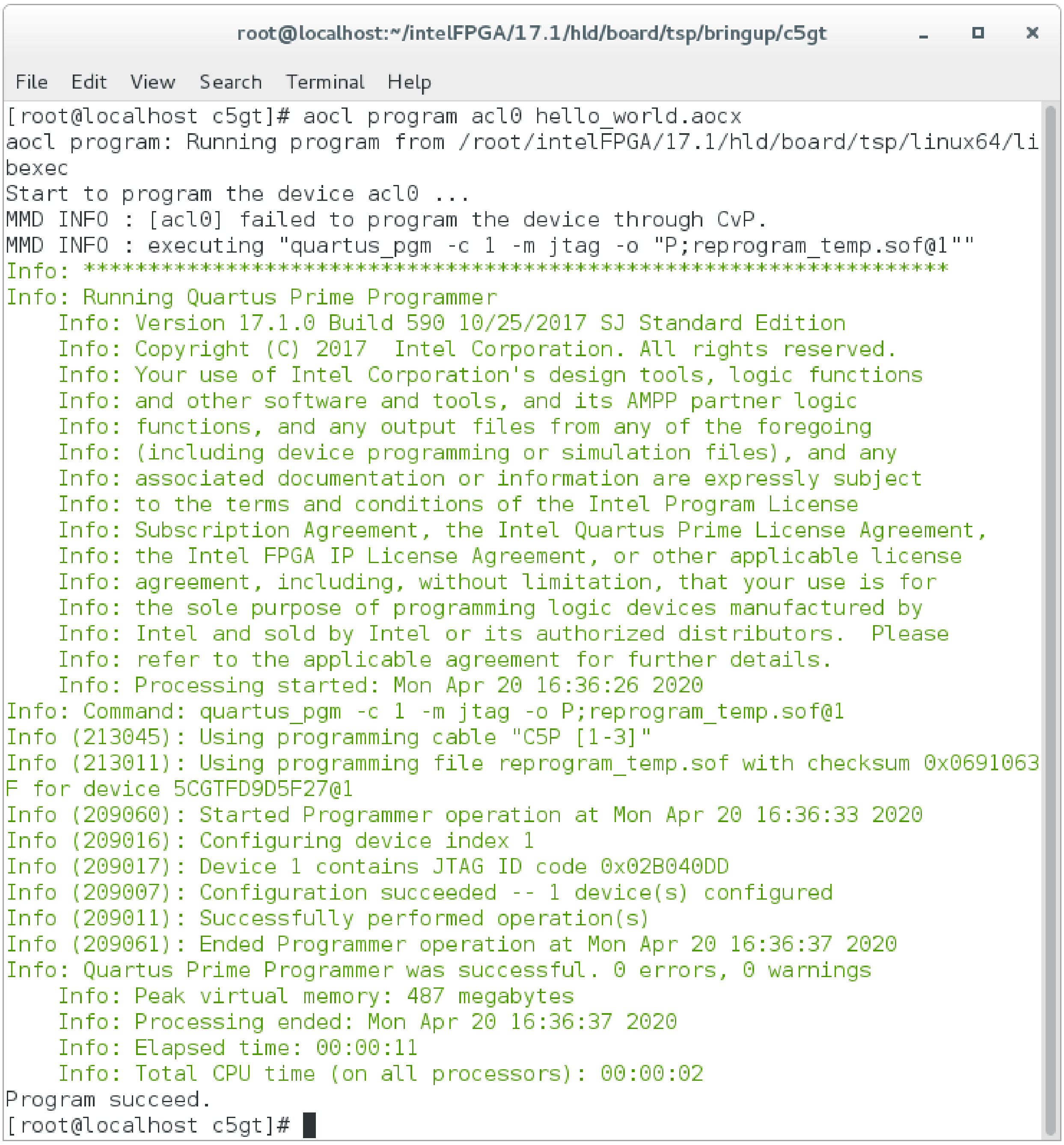
Figure 3-9 "aocl diagnose" messages

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 TSP OpenCL BSP programs the board with the specified .aocx file use the UB2 port.

Check whether the **hello_world** OpenCL image configures the FPGA successfully when TSP is power on. In the Linux terminal, type "*cd /root/intelFPGA/17.1/hld/board/tsp/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 c5gt]# aocl program acl0 hello_world.aocx
aocl program: Running program from /root/intelFPGA/17.1/hld/board/tsp/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: Mon Apr 20 16:36:26 2020
Info: Command: quartus_pgm -c 1 -m jtag -o P;reprogram_temp.sof@1
Info (213045): Using programming cable "C5P [1-3]"
Info (213011): Using programming file reprogram_temp.sof with checksum 0x0691063
F for device 5CGTFD9D5F27@1
Info (209060): Started Programmer operation at Mon Apr 20 16:36:33 2020
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 Mon Apr 20 16:36:37 2020
Info: Quartus Prime Programmer was successful. 0 errors, 0 warnings
    Info: Peak virtual memory: 487 megabytes
    Info: Processing ended: Mon Apr 20 16:36:37 2020
    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

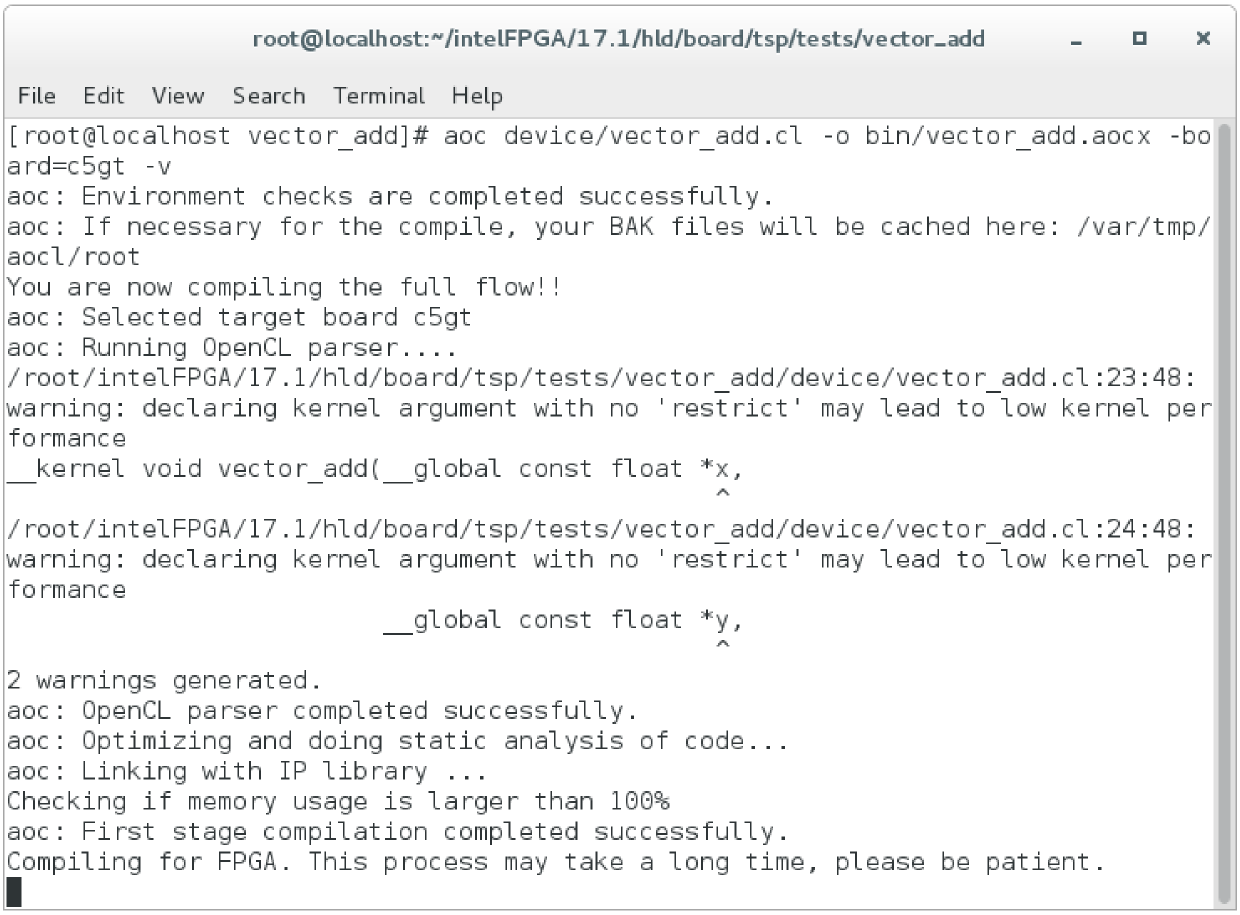
This section will show how to compile and run the OpenCL kernel and OpenCL Host Program for the vector_add example project. Developers can use the same procedures to compile and test other OpenCL examples (included in the kit) for TSP.

■ Compile OpenCL Kernel

In the terminal, type "*cd /root/intelFPGA/17.1/hld/board/tsp/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/tsp/tests/vector_add
File Edit View Search Terminal Help

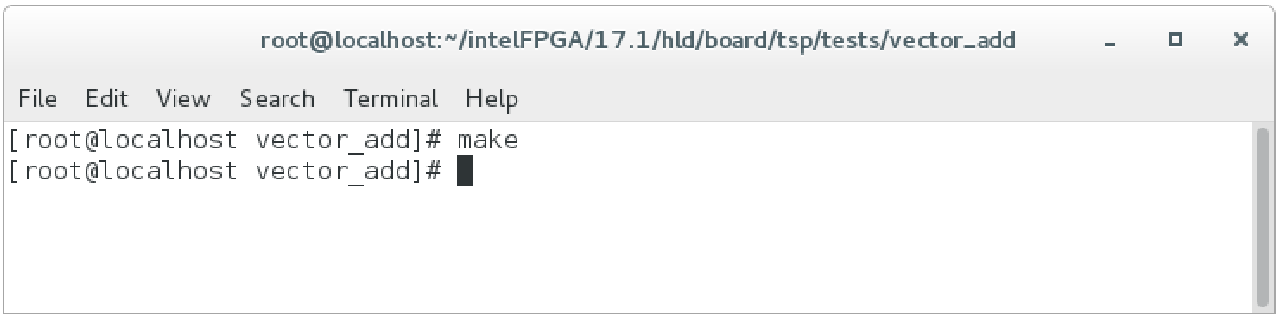
```
[root@localhost vector_add]# aoc device/vector_add.cl -o bin/vector_add.aocx -bo
ard=c5gt -v
aoc: Environment checks are completed successfully.
aoc: If necessary for the compile, your BAK files will be cached here: /var/tmp/
aocl/root
You are now compiling the full flow!!
aoc: Selected target board c5gt
aoc: Running OpenCL parser....
/root/intelFPGA/17.1/hld/board/tsp/tests/vector_add/device/vector_add.cl:23:48:
warning: declaring kernel argument with no 'restrict' may lead to low kernel per
formance
__kernel void vector_add(__global const float *x,
                                               ^

/root/intelFPGA/17.1/hld/board/tsp/tests/vector_add/device/vector_add.cl:24:48:
warning: declaring kernel argument with no 'restrict' may lead to low kernel per
formance
                         __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/tsp/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/tsp/tests/vector_add
File Edit View Search Terminal Help

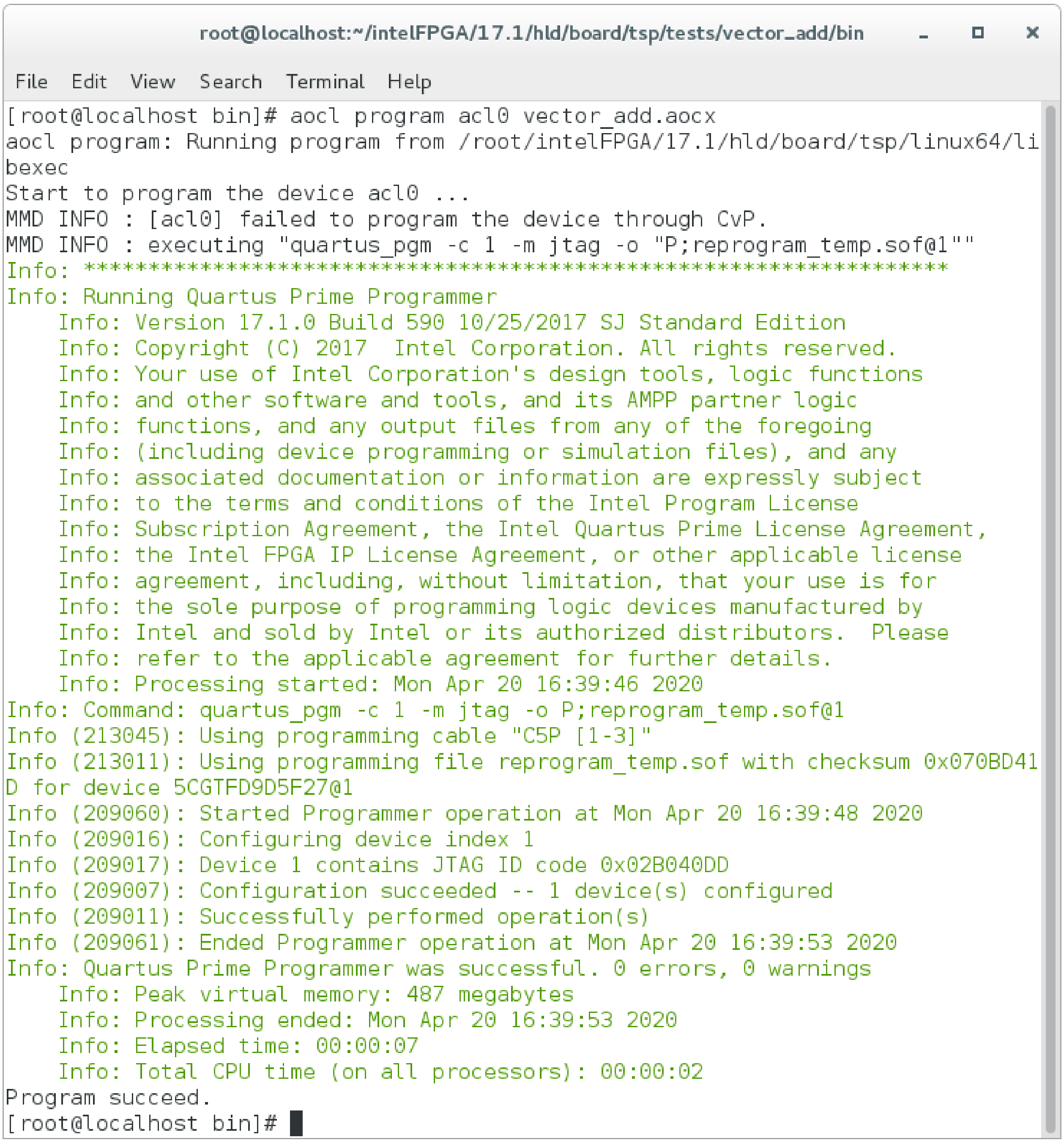
```
[root@localhost vector_add]# make
[root@localhost vector_add]# 
```

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/tsp/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.**

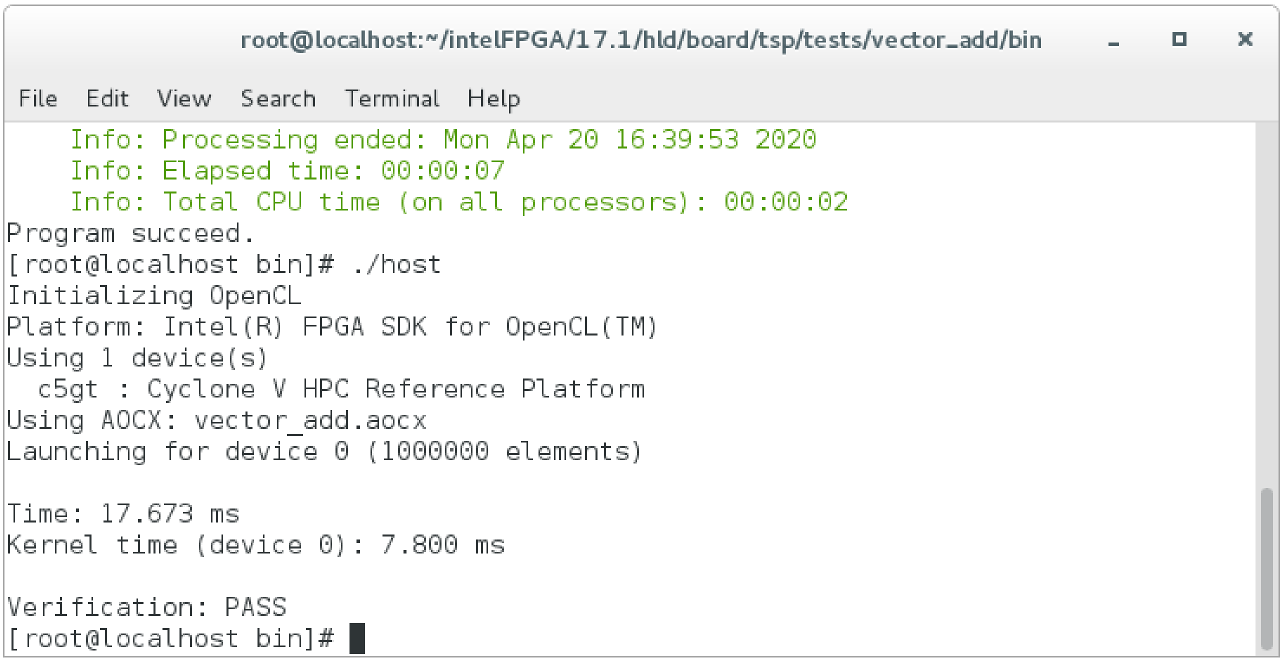


root@localhost:~/intelFPGA/17.1/hld/board/tsp/tests/vector_add/bin
File Edit View Search Terminal Help

```
[root@localhost bin]# aocl program acl0 vector_add.aocx
aocl program: Running program from /root/intelFPGA/17.1/hld/board/tsp/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: Mon Apr 20 16:39:46 2020
Info: Command: quartus_pgm -c 1 -m jtag -o P;reprogram_temp.sof@1
Info (213045): Using programming cable "C5P [1-3]"
Info (213011): Using programming file reprogram_temp.sof with checksum 0x070BD41
D for device 5CGTFD9D5F27@1
Info (209060): Started Programmer operation at Mon Apr 20 16:39:48 2020
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 Mon Apr 20 16:39:53 2020
Info: Quartus Prime Programmer was successful. 0 errors, 0 warnings
    Info: Peak virtual memory: 487 megabytes
    Info: Processing ended: Mon Apr 20 16:39:53 2020
    Info: Elapsed time: 00:00:07
    Info: Total CPU time (on all processors): 00:00:02
Program succeed.
[root@localhost bin]# █
```

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.



root@localhost:~/intelFPGA/17.1/hld/board/tsp/tests/vector_add/bin
File Edit View Search Terminal Help

```
    Info: Processing ended: Mon Apr 20 16:39:53 2020
    Info: Elapsed time: 00:00:07
    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: 17.673 ms
Kernel time (device 0): 7.800 ms

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

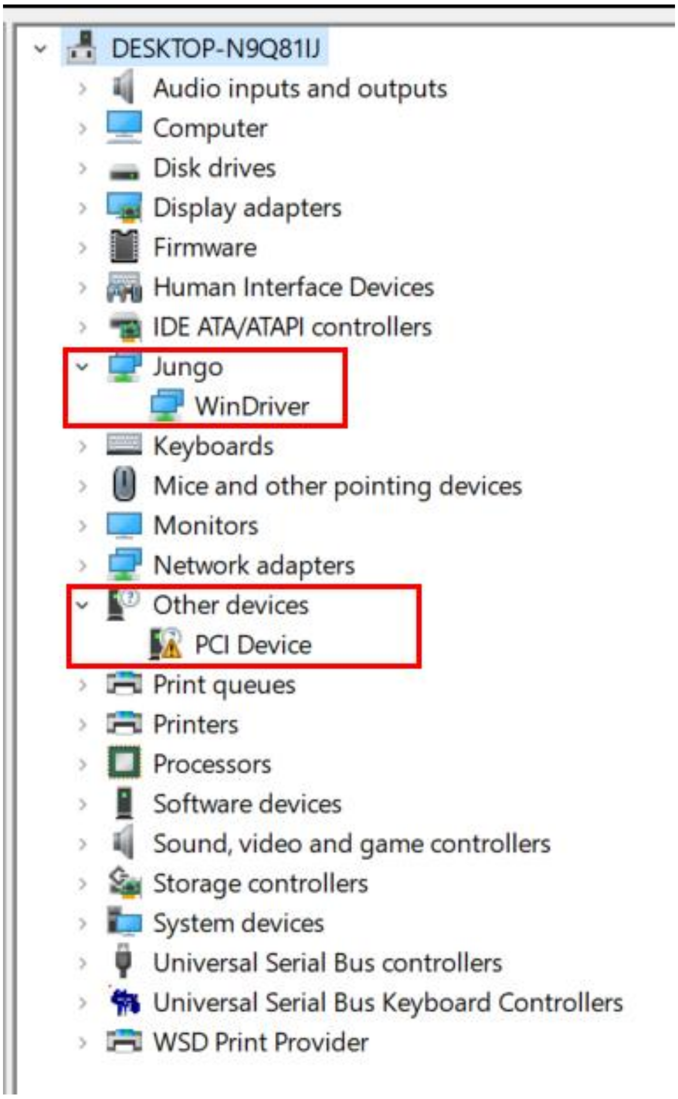
Figure 3-14 Successful Message for "vector_add" test

Chapter 4

Appendix

TSP 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.

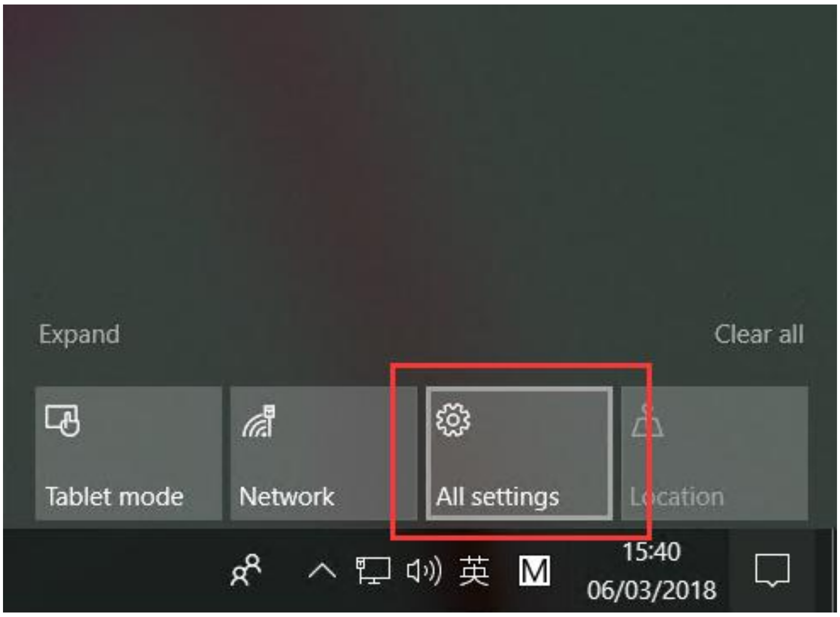


- DESKTOP-N9Q81IJ
  - Audio inputs and outputs
  - Computer
  - Disk drives
  - Display adapters
  - Firmware
  - Human Interface Devices
  - IDE ATA/ATAPI controllers
  - Jungo
    - WinDriver
  - Keyboards
  - Mice and other pointing devices
  - Monitors
  - Network adapters
  - Other devices
    - PCI Device
  - Print queues
  - Printers
  - Processors
  - Software devices
  - Sound, video and game controllers
  - Storage controllers
  - System devices
  - Universal Serial Bus controllers
  - Universal Serial Bus Keyboard Controllers
  - WSD Print Provider

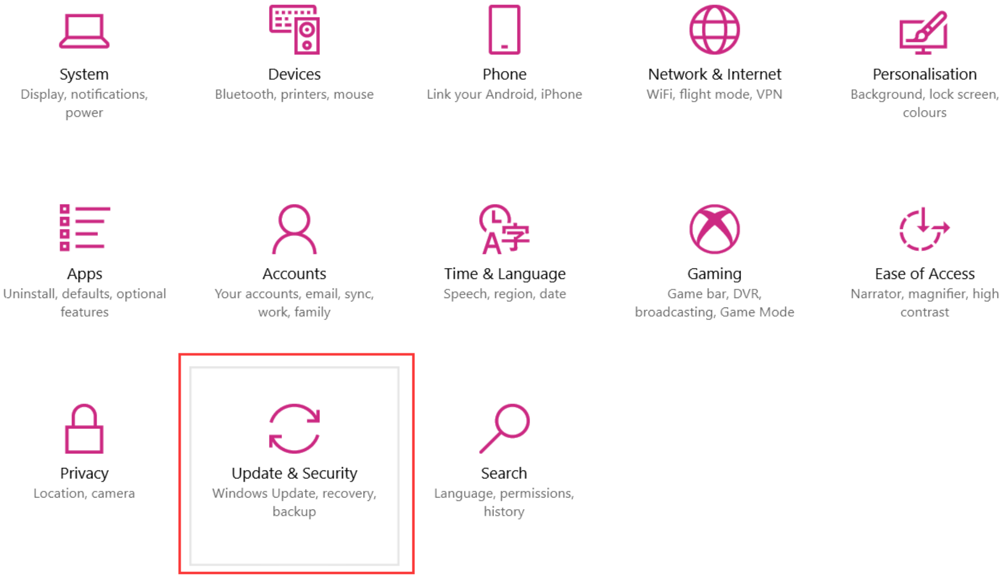
As a solution, it needs to disable the driver signature, then manually install the PCIe driver. The steps are as following:

A. Disable the driver signature in the Win10 system

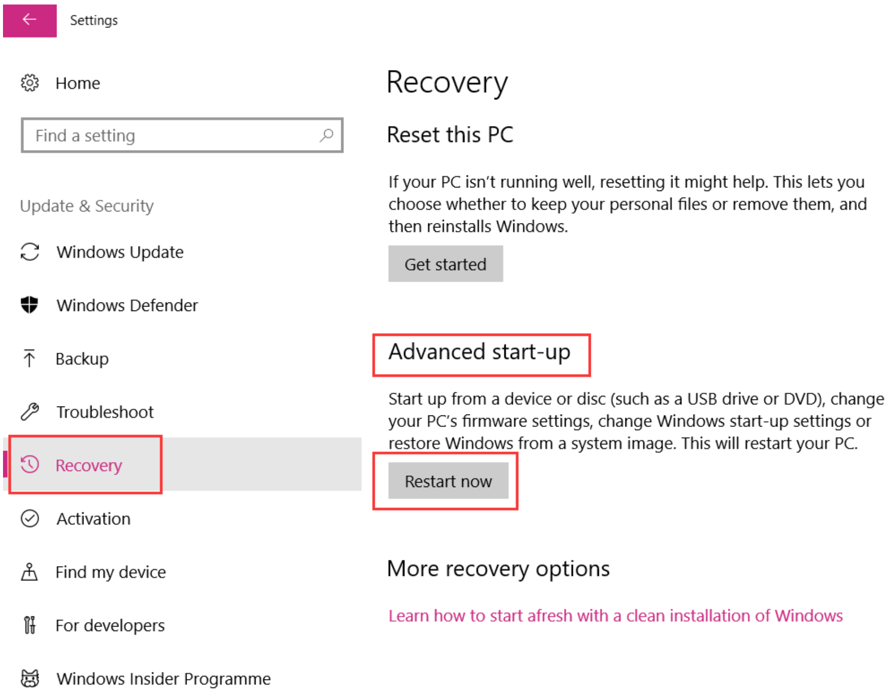
1.Click **Home**， enter “**All settings**”.



2. Access “**Update & Security**”.


System
Display, notifications, power
Devices
Bluetooth, printers, mouse
Phone
Link your Android, iPhone
Network & Internet
WiFi, flight mode, VPN
Personalisation
Background, lock screen, colours
Apps
Uninstall, defaults, optional features
Accounts
Your accounts, email, sync, work, family
Time & Language
Speech, region, date
Gaming
Game bar, DVR, broadcasting, Game Mode
Ease of Access
Narrator, magnifier, high contrast
Privacy
Location, camera
Update & Security
Windows Update, recovery, backup
Search
Language, permissions, history

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



Settings

Home

Find a setting

Update & Security

Windows Update

Windows Defender

Backup

Troubleshoot

Recovery

Activation

Find my device

For developers

Windows Insider Programme

Recovery

Reset this PC

If your PC isn't running well, resetting it might help. This lets you choose whether to keep your personal files or remove them, and then reinstalls Windows.

Get started

Advanced start-up

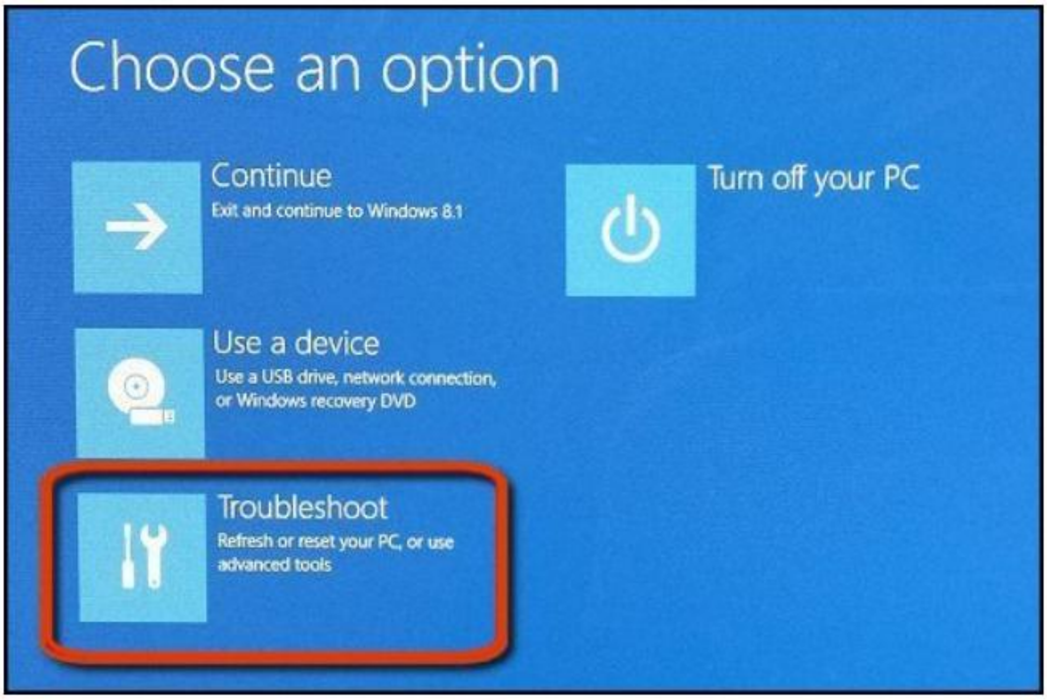
Start up from a device or disc (such as a USB drive or DVD), change your PC's firmware settings, change Windows start-up settings or restore Windows from a system image. This will restart your PC.

Restart now

More recovery options

Learn how to start afresh with a clean installation of Windows

4.After restarting, choose "**Troubleshoot**".



Choose an option

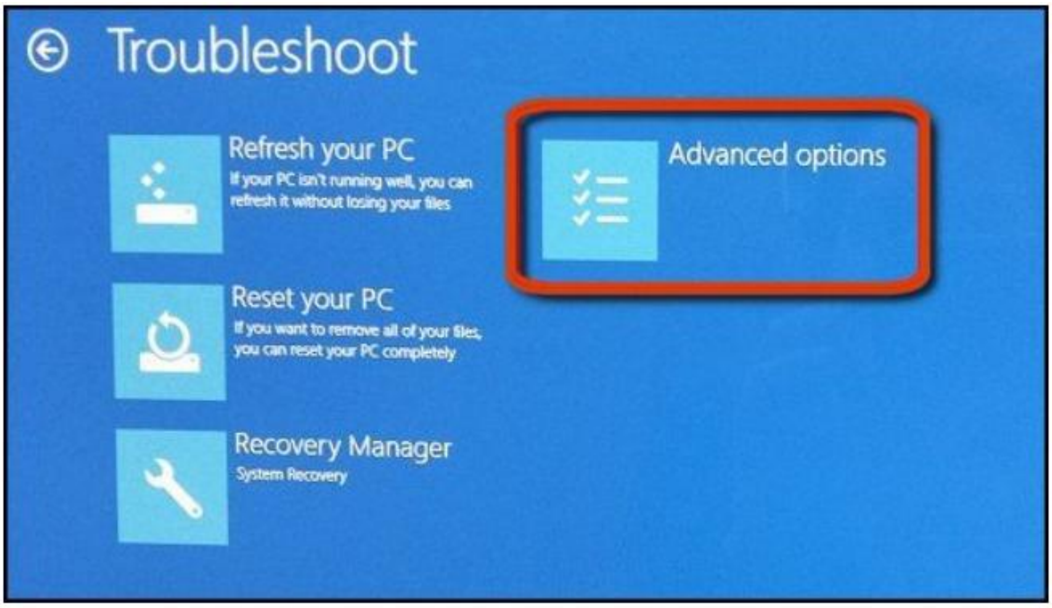
Continue
Exit and continue to Windows 8.1

Turn off your PC

Use a device
Use a USB drive, network connection, or Windows recovery DVD

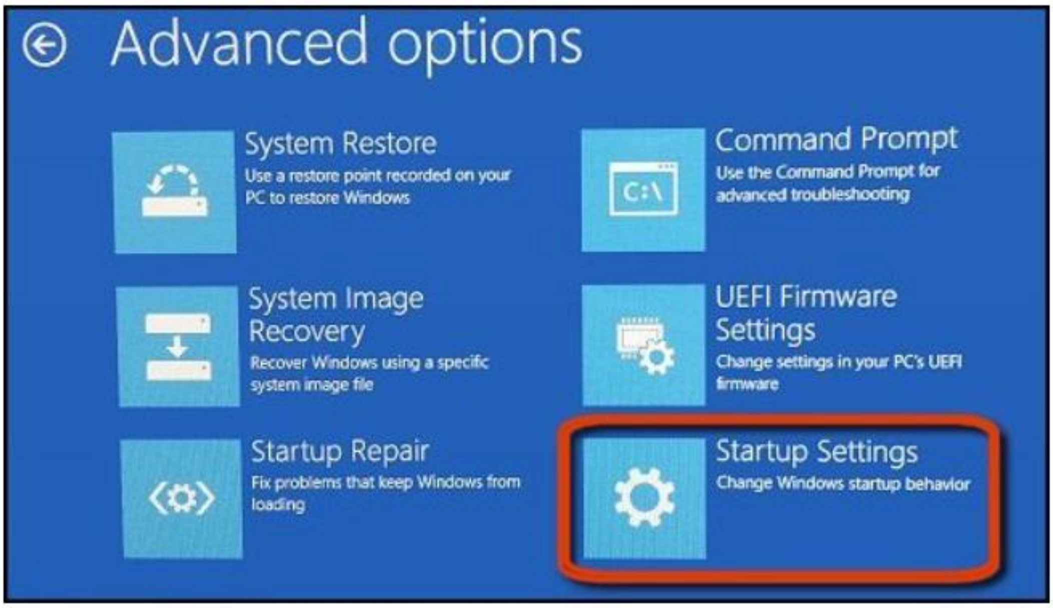
Troubleshoot
Refresh or reset your PC, or use advanced tools

5.Choose "**Advanced options**".



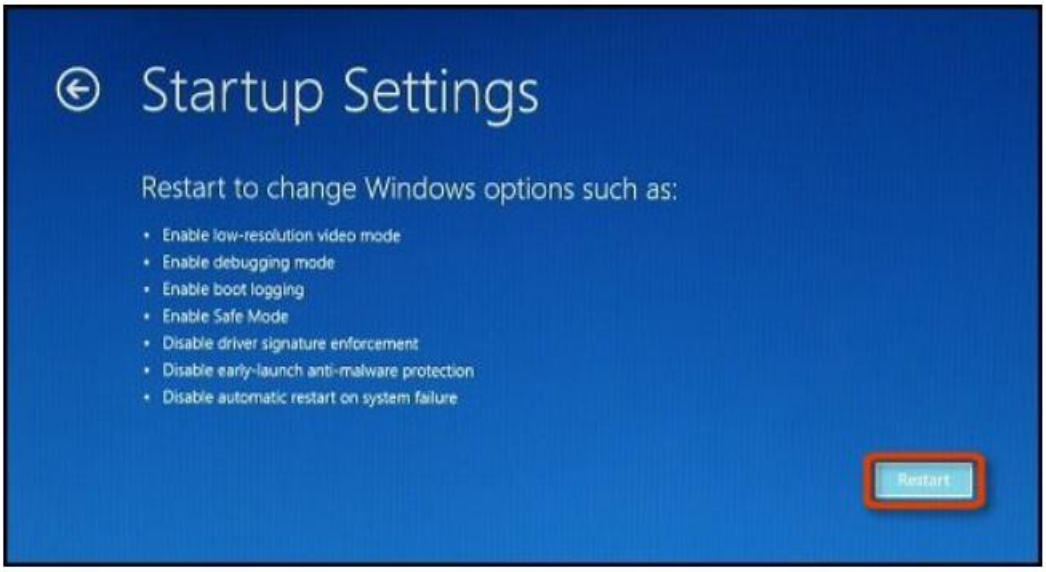
Troubleshoot
Refresh your PC
If your PC isn't running well, you can refresh it without losing your files
Reset your PC
If you want to remove all of your files, you can reset your PC completely
Recovery Manager
System Recovery
Advanced options

6.Choose “**Start-up Settings**”.



Advanced options
System Restore
Use a restore point recorded on your PC to restore Windows
Command Prompt
Use the Command Prompt for advanced troubleshooting
System Image Recovery
Recover Windows using a specific system image file
UEFI Firmware Settings
Change settings in your PC's UEFI firmware
Startup Repair
Fix problems that keep Windows from loading
Startup Settings
Change Windows startup behavior

7.Click “**Restart**”.

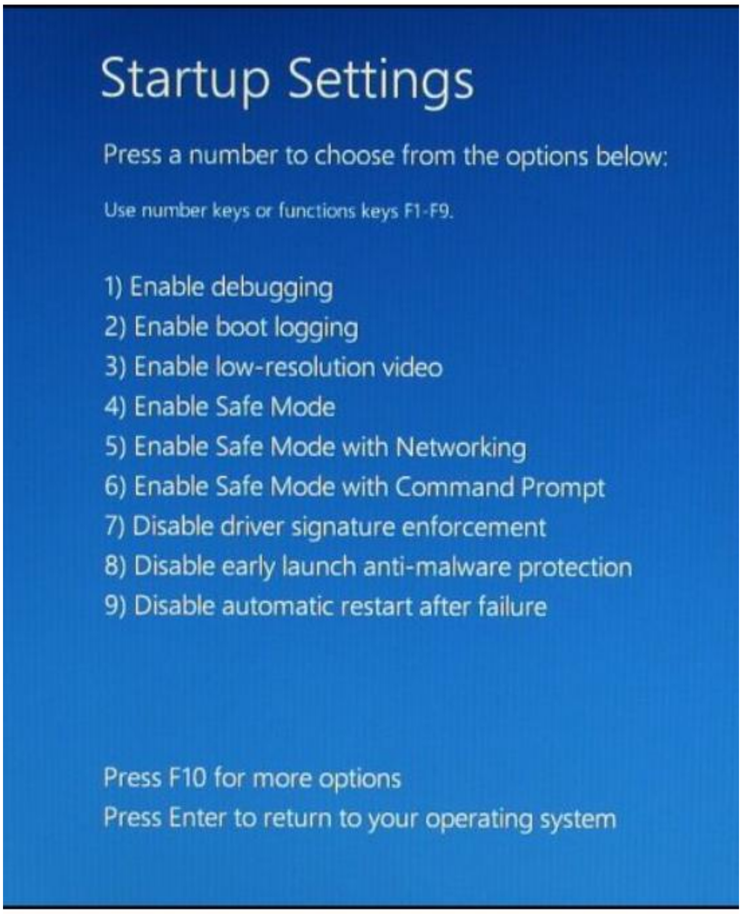


Startup Settings

Restart to change Windows options such as:

- Enable low-resolution video mode
- Enable debugging mode
- Enable boot logging
- Enable Safe Mode
- Disable driver signature enforcement
- Disable early-launch anti-malware protection
- Disable automatic restart on system failure

8.Enter “**F7**” to disable driver signature.



Startup Settings

Press a number to choose from the options below:

Use number keys or functions keys F1-F9.

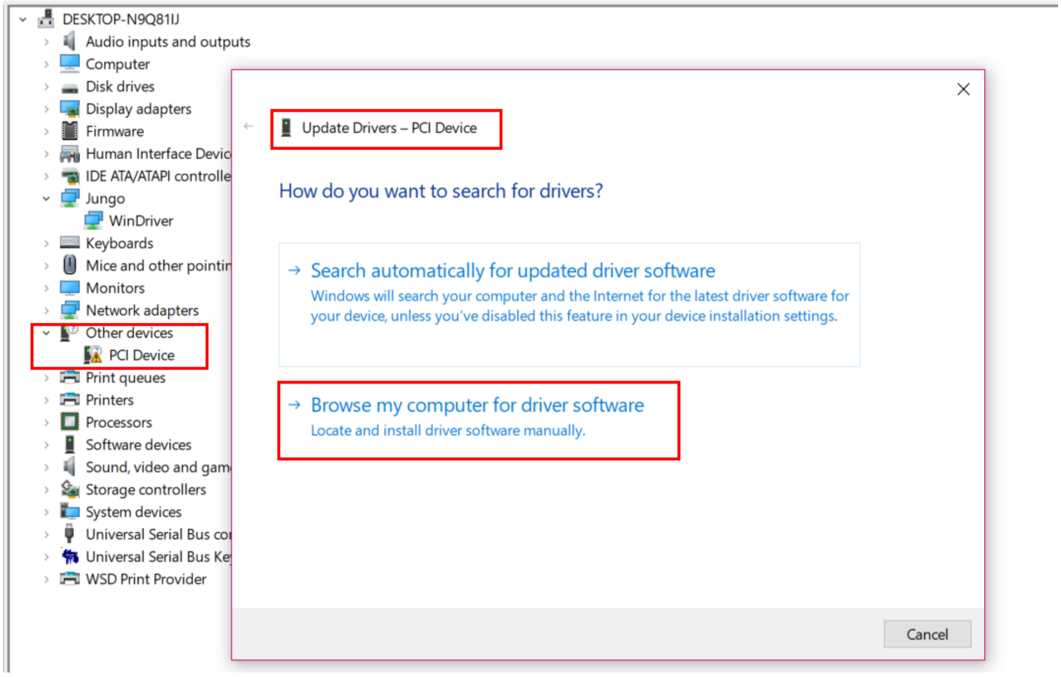
1) Enable debugging
2) Enable boot logging
3) Enable low-resolution video
4) Enable Safe Mode
5) Enable Safe Mode with Networking
6) Enable Safe Mode with Command Prompt
7) Disable driver signature enforcement
8) Disable early launch anti-malware protection
9) Disable automatic restart after failure

Press F10 for more options
Press Enter to return to your operating system

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



Update Drivers – PCI Device

How do you want to search for drivers?

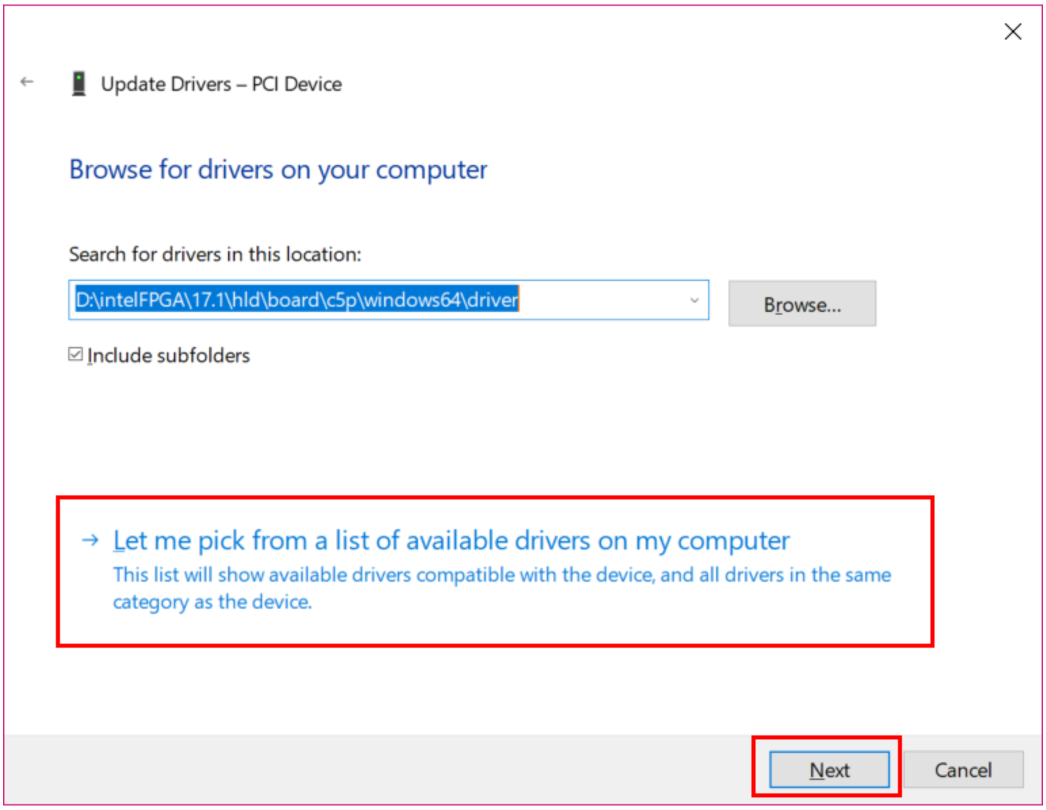
→ Search automatically for updated driver software

Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.

→ Browse my computer for driver software

Locate and install driver software manually.

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



Update Drivers – PCI Device

Browse for drivers on your computer

Search for drivers in this location:

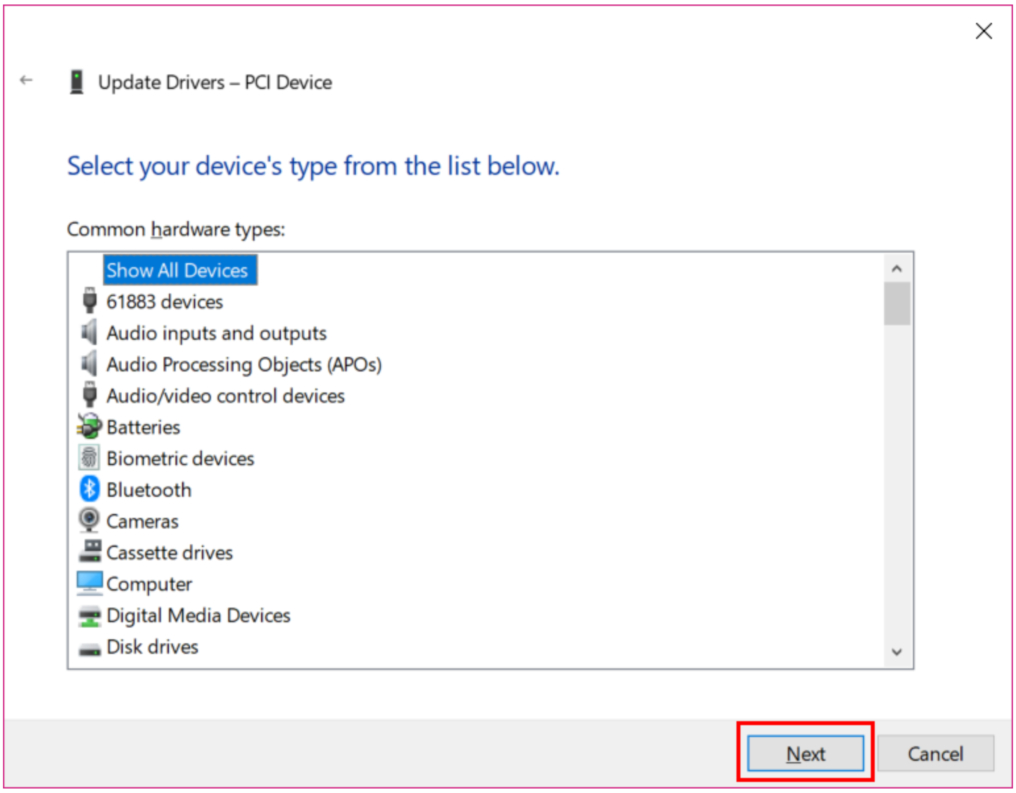
D:\intelFPGA\17.1\hld\board\c5p\windows64\driver

☑ Include subfolders

→ Let me pick from a list of available drivers on my computer

This list will show available drivers compatible with the device, and all drivers in the same category as the device.

3.Continue choose “**Next**”.



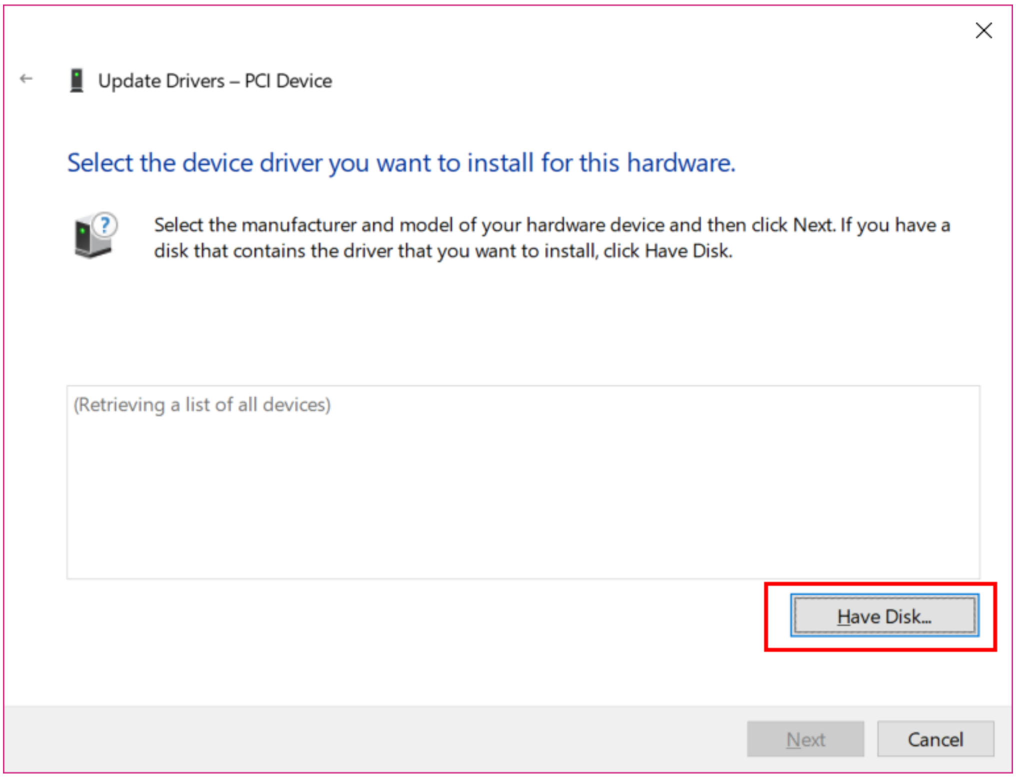
Update Drivers – PCI Device

Select your device's type from the list below.

Common hardware types:

- Show All Devices
- 61883 devices
- Audio inputs and outputs
- Audio Processing Objects (APOs)
- Audio/video control devices
- Batteries
- Biometric devices
- Bluetooth
- Cameras
- Cassette drives
- Computer
- Digital Media Devices
- Disk drives

4.Choose “**Have Disk ..**.”

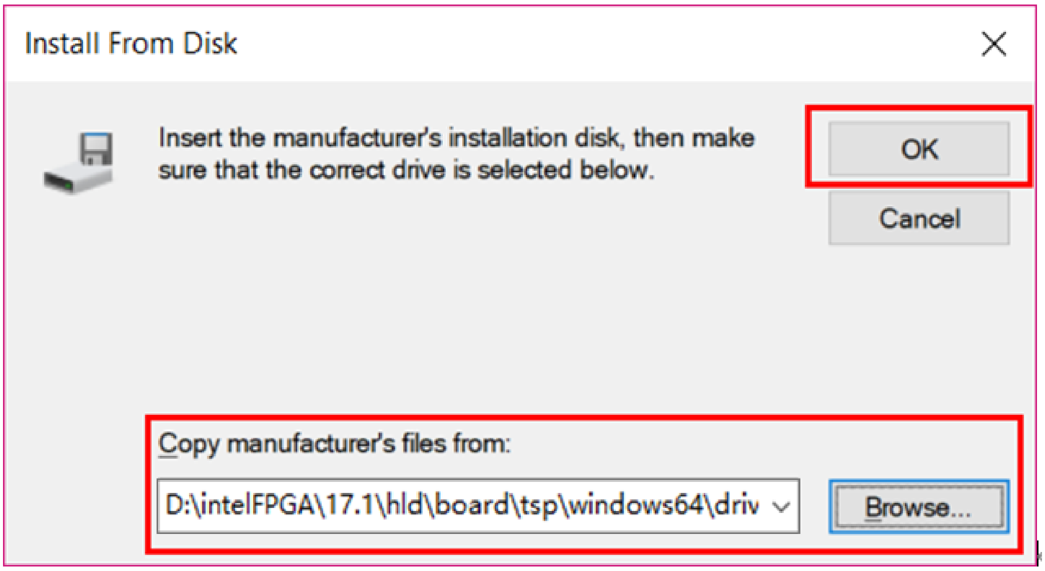


Update Drivers – PCI Device

Select the device driver you want to install for this hardware.

Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver that you want to install, click Have Disk.

(Retrieving a list of all devices)



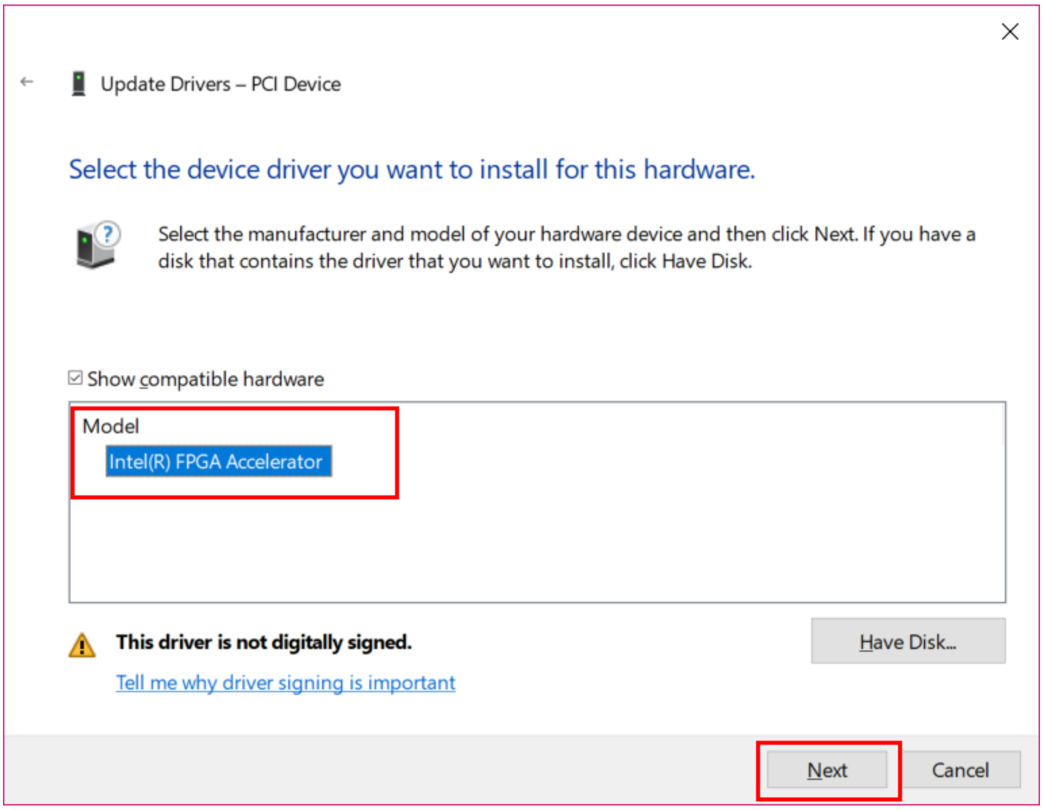
Install From Disk

Insert the manufacturer's installation disk, then make sure that the correct drive is selected below.

Copy manufacturer's files from:

D:\intelFPGA\17.1\hld\board\tsp\windows64\driv

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



Update Drivers – PCI Device

Select the device driver you want to install for this hardware.

Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver that you want to install, click Have Disk.

☑ Show compatible hardware

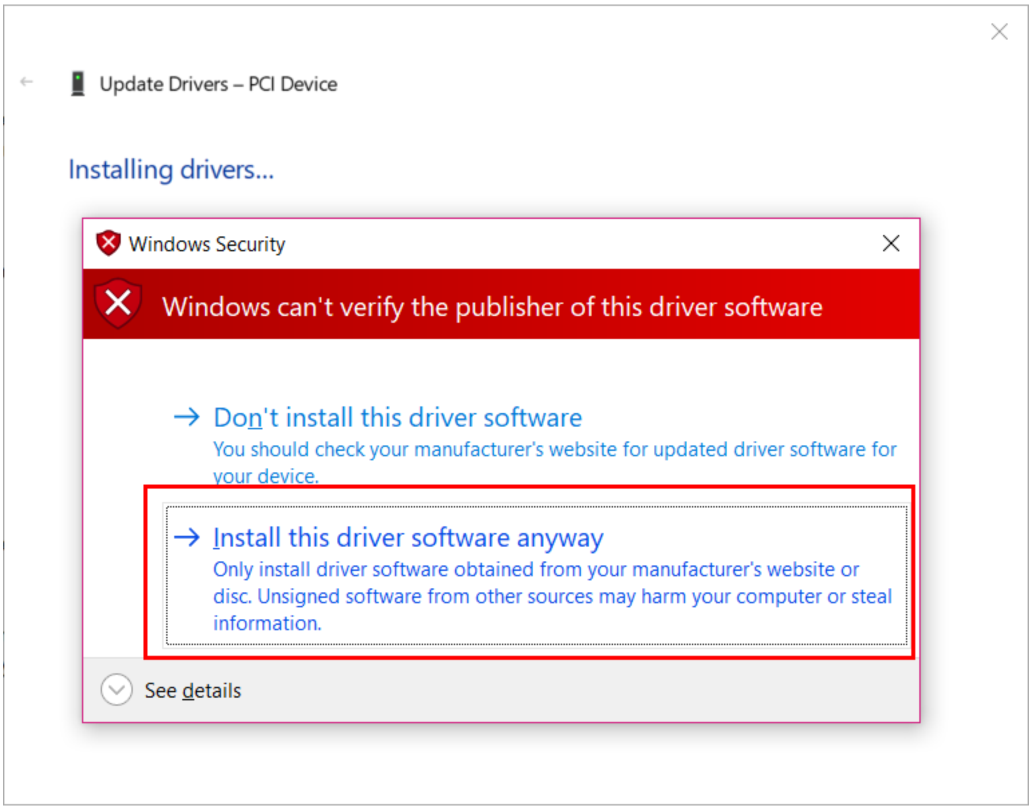
Model

Intel(R) FPGA Accelerator

This driver is not digitally signed.

Tell me why driver signing is important

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


Update Drivers – PCI Device

Installing drivers...

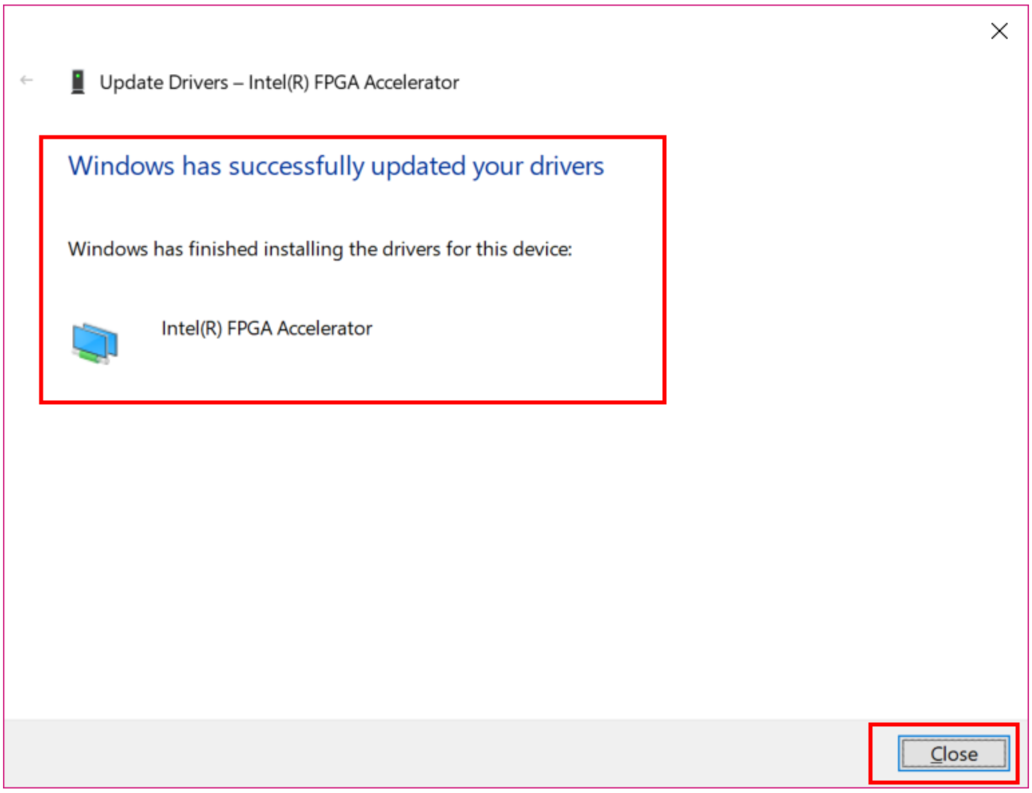
Windows Security

Windows can't verify the publisher of this driver software

→ Don't install this driver software
You should check your manufacturer's website for updated driver software for your device.

→ Install this driver software anyway
Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information.

8.Installation complete.

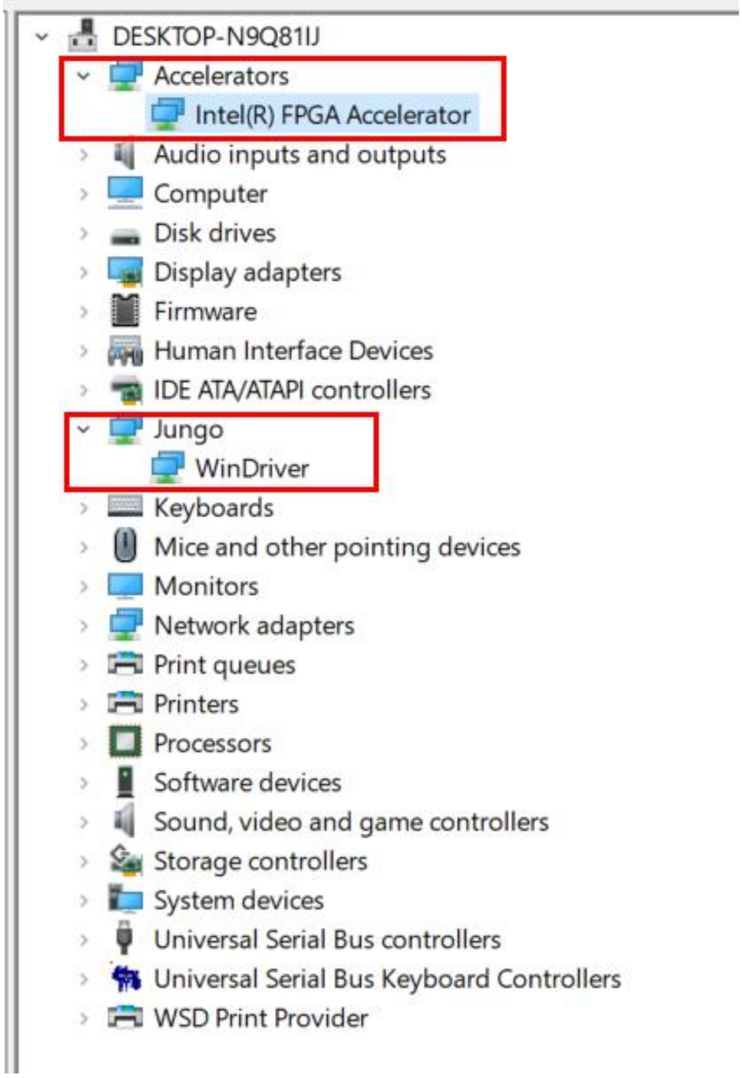

Update Drivers – Intel(R) FPGA Accelerator

Windows has successfully updated your drivers

Windows has finished installing the drivers for this device:

Intel(R) FPGA Accelerator

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



DESKTOP-N9Q81IJ

- Accelerators
  - Intel(R) FPGA Accelerator
- Audio inputs and outputs
- Computer
- Disk drives
- Display adapters
- Firmware
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Jungo
  - WinDriver
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Print queues
- Printers
- Processors
- Software devices
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers
- Universal Serial Bus Keyboard Controllers
- WSD Print Provider

Revision History

Version	*Change Log*
V1.0	**Initial Version**

Copyright Statement