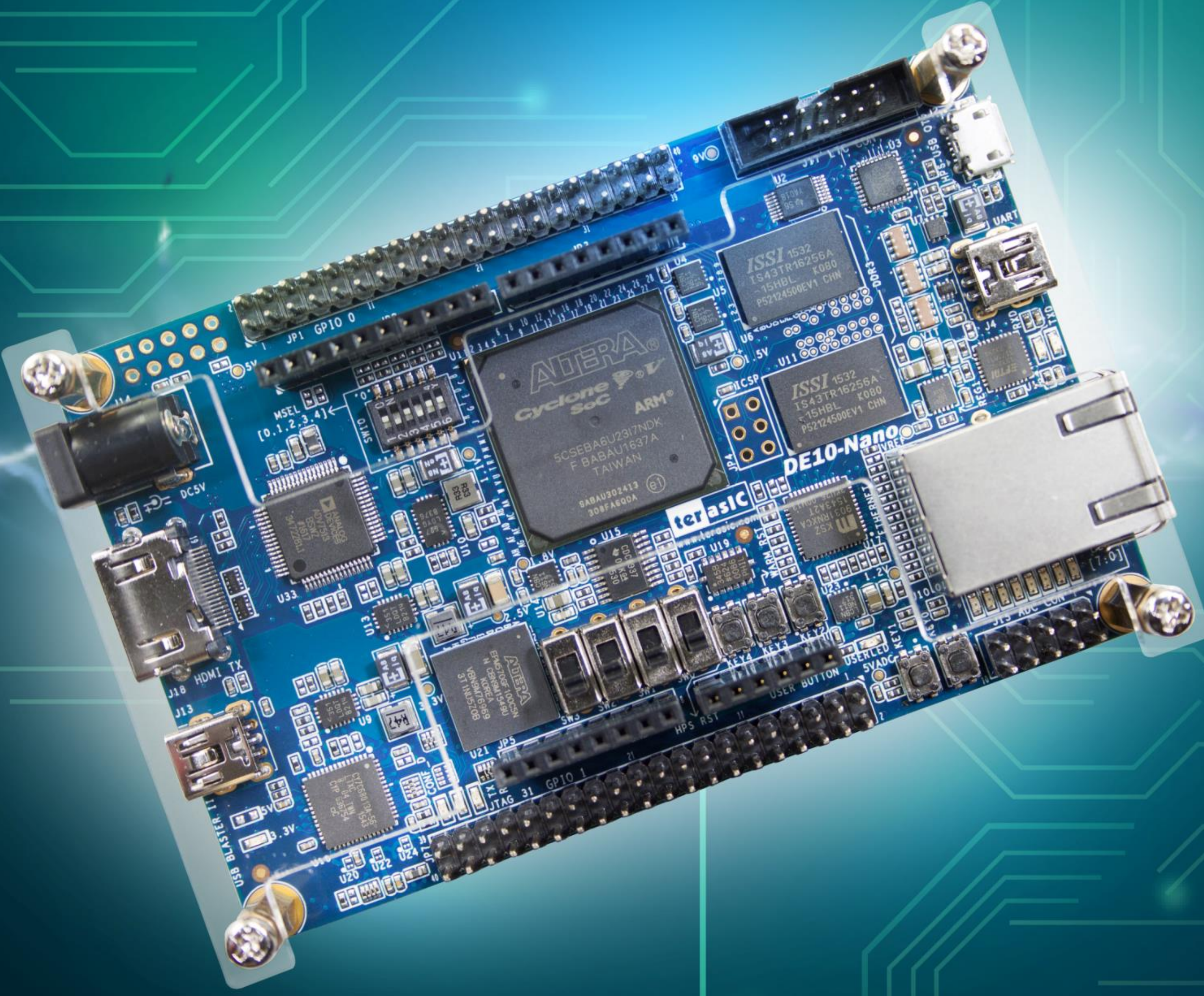
DE10-Nano

OpenCL



www.terasic.com

CONTENTS

Chapter 1 *Introduction*

DE10-Nano is a robust hardware design platform built with Intel System-on-Chip (SoC) FPGA. It is designed for Intel University Program. This document gives introduction on how to setup OpenCL development environment, compile, and execute example projects for DE10-Nano. Users can refer to Intel SDK for OpenCL Programming Guide for more details about OpenCL coding instruction.

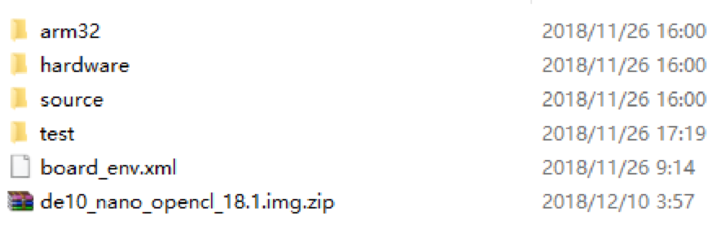
http://www.altera.com/literature/hb/opencl-sdk/aocl_programming_guide.pdf

1.1 DE10-Nano OpenCL BSP

The DE10-Nano OpenCL Board Support Package (BSP) contains required resources for users to develop OpenCL project based on DE10-Nano Board. The BSP is available from the website:

http://de10-nano.terasic.com/cd

For Windows Host, please download DE10-Nano_OpenCL_BSP_18.1.zip. For Linux, please download the DE10-Nano_OpenCL_BSP_18.1.tar.gz. The CentOS 7.2 Linux distribution is recommended for the OpenCL application. These two compressed files are different in the compression type only and their contents are the same. **Figure 1-1** shows the contents of OpenCL BSP for DE10-Nano.



```
arm32                          2018/11/26 16:00
hardware                       2018/11/26 16:00
source                         2018/11/26 16:00
test                           2018/11/26 17:19
board_env.xml                  2018/11/26 9:14
de10_nano_opencl_18.1.img.zip  2018/12/10 3:57
```

Figure 1-1 Contents of OpenCL BSP for DE10-Nano

1.2 System Requirements

The following items are required to setup OpenCL for DE10-Nano board:

- Terasic DE10-Nano board
- microSD card with at least 4GB capacity
- microSD card reader
- USB cable (type A to mini-B)
- Ethernet cable or USB-Storage
- Host PC with
  - USB host port
  - 32GB memory is recommended
  - 64-bit Windows 10 or Linux
  - Win32 Disk Imager
  - PuTTY or Minicom(Linux) utility
  - Intel Quartus Prime Standard Edition 18.1.0.625 installed with valid license
  - Intel FPGA SDK for OpenCL Prime Edition 18.1.0.625 installed without license
  - Intel SoC EDS 18.1.0.625 installed

1.3 OpenCL Architecture

An OpenCL project consists of OpenCL Kernel and Host Program, as shown in **Figure 1-2**. The Kernel is realized on the FPGA part of SoC FPGA. The Host Program is not the ARM part of the SoC FPGA. It is cross-compiled by Intel SoC EDS installed on Windows or Linux. The Kernel is developed in Quartus and OpenCL SDK is installed on Windows or Linux.

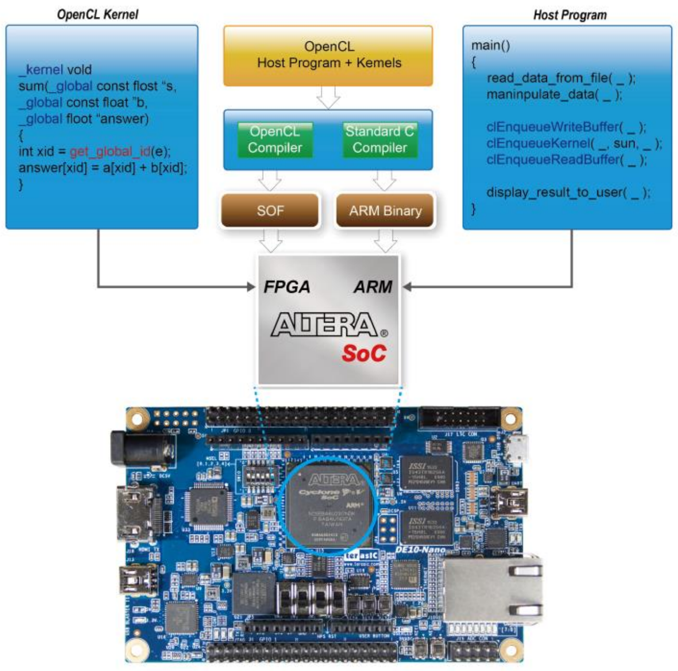

OpenCL Kernel
_kernel void
sum(_global const flost *s,
_global const float *b,
_global floot *answer)
{
int xid = get_global_id(e);
answer[xid] = a[xid] + b[xid];
}
OpenCL
Host Program + Kernels
OpenCL Compiler
Standard C Compiler
SOF
ARM Binary
Host Program
main()
{
read_data_from_file(_);
maninpulate_data(_);
clEnqueueWriteBuffer(_);
clEnqueueKernel(_, sun, _);
clEnqueueReadBuffer(_);
display_result_to_user(_);
}
FPGA
ARM
ALTERA
SoC

Figure 1-2 Intel SoC FPGA OpenCL architecture

1.4 Execute OpenCL Demo on DE10-Nano

This section describes how to execute OpenCL demo on DE10-Nano from the Linux image file included in DE10-Nano Board Support Package (BSP). Windows or Linux Host is required to setup the demo.

■ Using Windows Host PC

The following software should be installed on the Windows host PC to complete the setup.

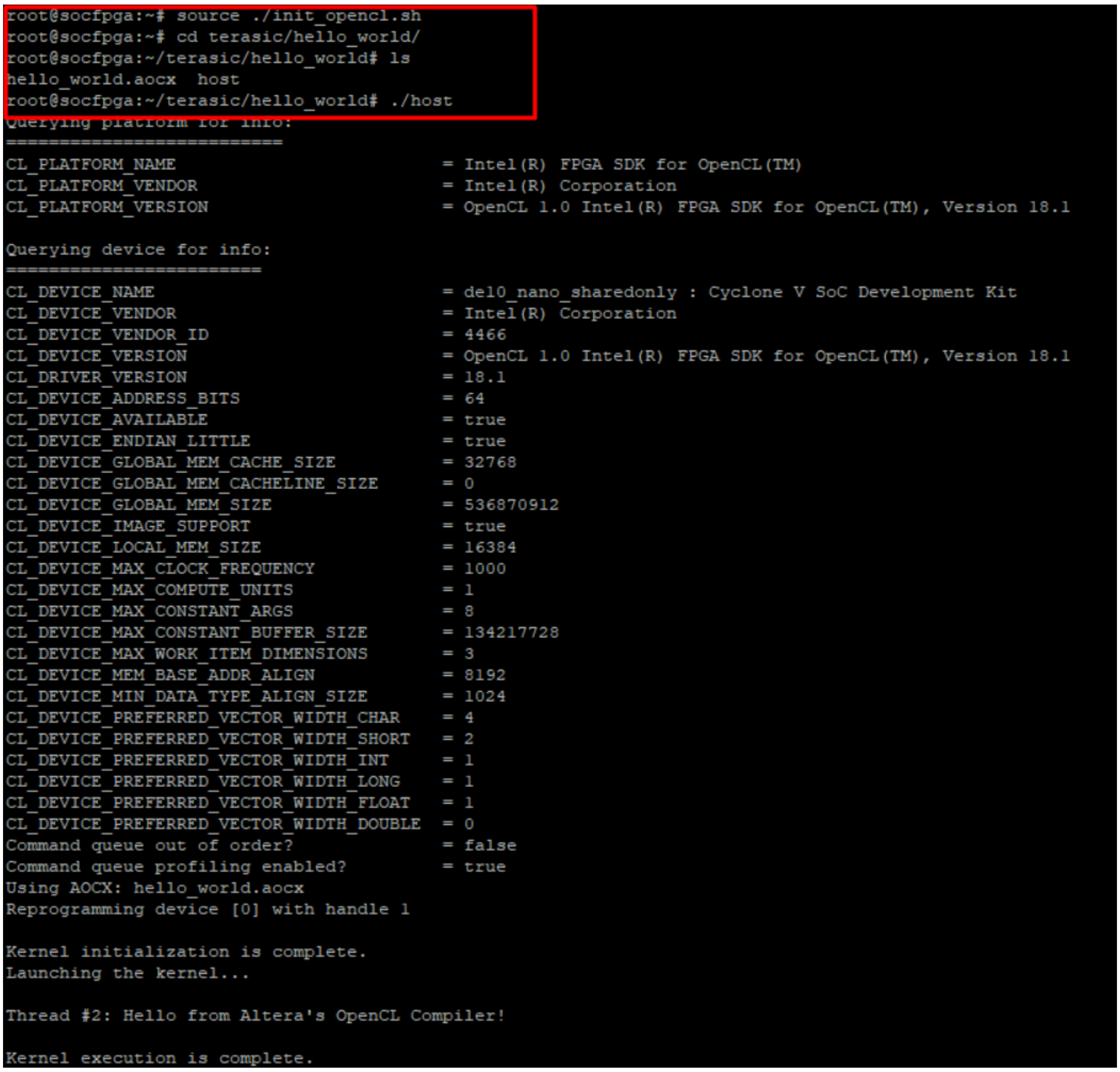
- Disk Imager - available from http://sourceforge.net/projects/win32diskimager
- PuTTY- available from https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe

The procedures to execute the **hello_world** and **vector_add** demos are:

1. Download DE10-Nano BSP from http://de10-nano.terasic.com/cd and extract the Linux image file **de10_nano_opencl_18.1.img** from **de10_nano_opencl_18.1.img.zip**
2. Write the Linux image file **de10_nano_opencl_18.1.img** into a microSD card with Disk Imager Utility.
3. Insert the microSD card into the microSD card socket(J11).
4. Make sure the DIP switch (SW10) **MSEL[4:0] = 01010**.
5. Connect your host PC to the UART-to-USB port (J4) on DE10-Nano via an USB cable.

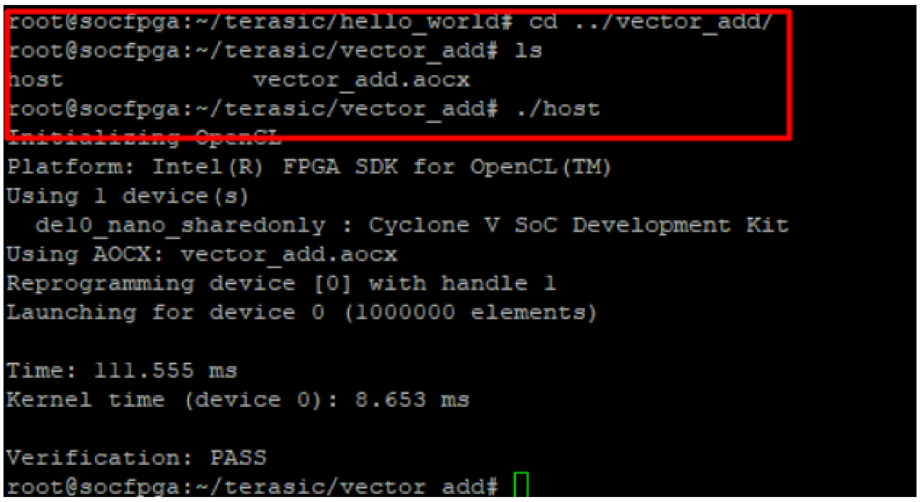
Users need to install the UART-to-USB device driver as described in the DE10-Nano Getting Started Guide.

6. Launch PuTTY utility on your host PC. Make sure the baud rate is set to 115200.
7. Power on DE10-Nano to boot Linux. Login Linux with username '**root**' (password is not required).
8. Type "source ./init_opencl.sh" to load the OpenCL Linux kernel driver and setup environment variable for OpenCL Run-Time library, which is already installed on the microSD card.
9. Launch hello_world demo:
   - Type "cd terasic/hello_world/" to go to the hello_world folder.
   - Type "ls" to see the files in this folder.
   - Type "./host" to launch the hello_world host application, as shown in **Figure 1-3**.
10. Launch vector_add demo:
   - Type "cd ../vector_add/" to go to the vector_add folder.
   - Type "ls" to see the files in this folder.
   - Type "./host" to launch the vector_add host application, as shown in **Figure 1-4** .



```
root@socfpga:~# source ./init_opencl.sh
root@socfpga:~# cd terasic/hello_world/
root@socfpga:~/terasic/hello_world# ls
hello_world.aocx  host
root@socfpga:~/terasic/hello_world# ./host
Querying platform for info:
==========================
CL_PLATFORM_NAME                         = Intel(R) FPGA SDK for OpenCL(TM)
CL_PLATFORM_VENDOR                       = Intel(R) Corporation
CL_PLATFORM_VERSION                      = OpenCL 1.0 Intel(R) FPGA SDK for OpenCL(TM), Version 18.1

Querying device for info:
========================
CL_DEVICE_NAME                           = de10_nano_sharedonly : Cyclone V SoC Development Kit
CL_DEVICE_VENDOR                         = Intel(R) Corporation
CL_DEVICE_VENDOR_ID                      = 4466
CL_DEVICE_VERSION                        = OpenCL 1.0 Intel(R) FPGA SDK for OpenCL(TM), Version 18.1
CL_DRIVER_VERSION                        = 18.1
CL_DEVICE_ADDRESS_BITS                   = 64
CL_DEVICE_AVAILABLE                      = true
CL_DEVICE_ENDIAN_LITTLE                  = true
CL_DEVICE_GLOBAL_MEM_CACHE_SIZE          = 32768
CL_DEVICE_GLOBAL_MEM_CACHELINE_SIZE      = 0
CL_DEVICE_GLOBAL_MEM_SIZE                = 536870912
CL_DEVICE_IMAGE_SUPPORT                  = true
CL_DEVICE_LOCAL_MEM_SIZE                 = 16384
CL_DEVICE_MAX_CLOCK_FREQUENCY            = 1000
CL_DEVICE_MAX_COMPUTE_UNITS              = 1
CL_DEVICE_MAX_CONSTANT_ARGS              = 8
CL_DEVICE_MAX_CONSTANT_BUFFER_SIZE       = 134217728
CL_DEVICE_MAX_WORK_ITEM_DIMENSIONS       = 3
CL_DEVICE_MEM_BASE_ADDR_ALIGN            = 8192
CL_DEVICE_MIN_DATA_TYPE_ALIGN_SIZE       = 1024
CL_DEVICE_PREFERRED_VECTOR_WIDTH_CHAR    = 4
CL_DEVICE_PREFERRED_VECTOR_WIDTH_SHORT   = 2
CL_DEVICE_PREFERRED_VECTOR_WIDTH_INT     = 1
CL_DEVICE_PREFERRED_VECTOR_WIDTH_LONG    = 1
CL_DEVICE_PREFERRED_VECTOR_WIDTH_FLOAT   = 1
CL_DEVICE_PREFERRED_VECTOR_WIDTH_DOUBLE  = 0
Command queue out of order?              = false
Command queue profiling enabled?         = true
Using AOCX: hello_world.aocx
Reprogramming device [0] with handle 1

Kernel initialization is complete.
Launching the kernel...

Thread #2: Hello from Altera's OpenCL Compiler!

Kernel execution is complete.
```

Figure 1-3 Hello World Demo



```
root@socfpga:~/terasic/hello_world# cd ../vector_add/
root@socfpga:~/terasic/vector_add# ls
host            vector_add.aocx
root@socfpga:~/terasic/vector_add# ./host
Initializing OpenCL
Platform: Intel(R) FPGA SDK for OpenCL(TM)
Using 1 device(s)
  de10_nano_sharedonly : Cyclone V SoC Development Kit
Using AOCX: vector_add.aocx
Reprogramming device [0] with handle 1
Launching for device 0 (1000000 elements)

Time: 111.555 ms
Kernel time (device 0): 8.653 ms

Verification: PASS
root@socfpga:~/terasic/vector_add#
```

Figure 1-4 Vector Add Demo

■ Using Linux Host PC with Root Privilege

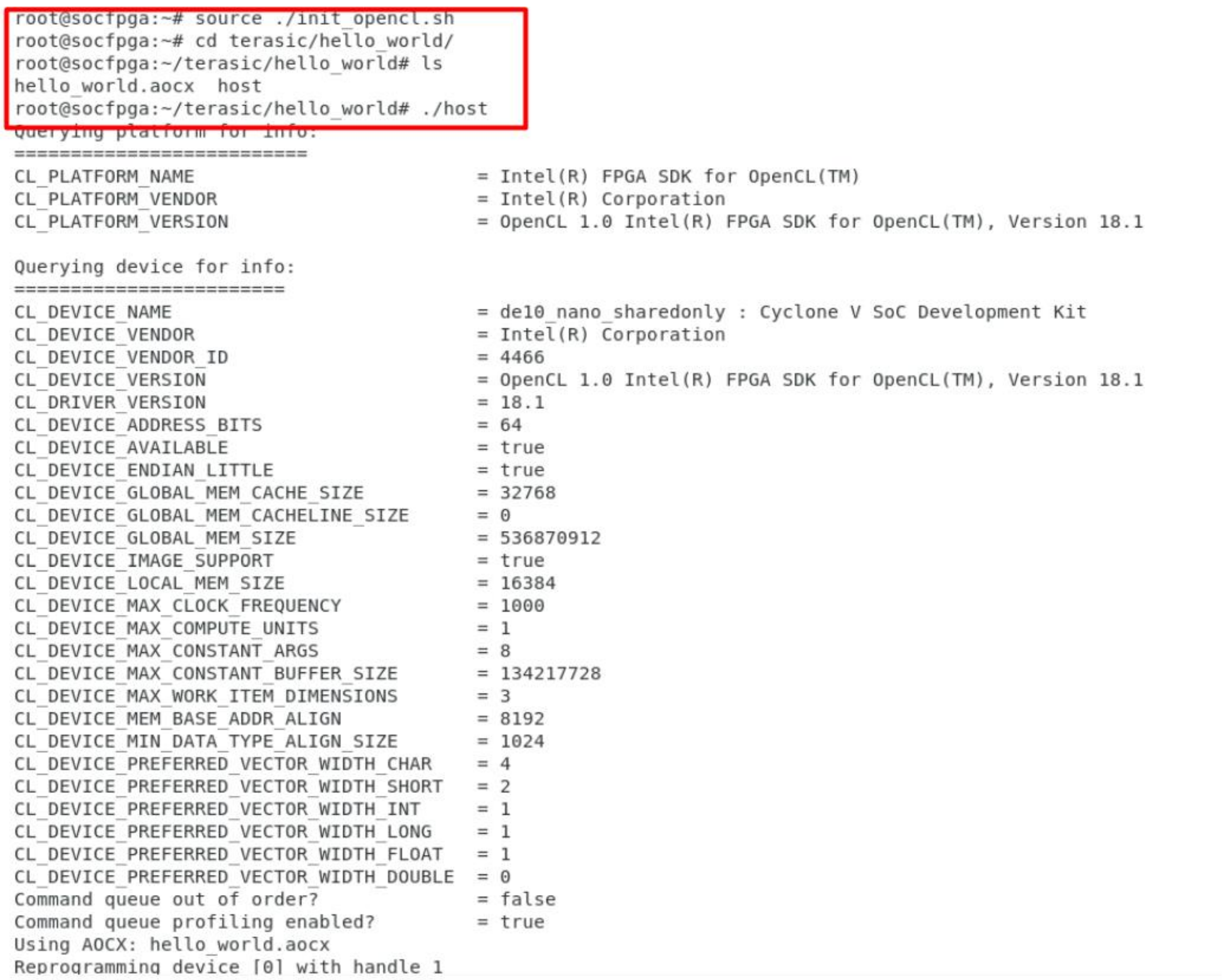
The following software should be installed on the Linux host PC to complete the setup.

NOTE: Minicom – a terminal which can be installed via command “yum install minicom” or “apt-get install minicom”

1. Download DE10-Nano BSP from http://de10-nano.terasic.com/cd and extract the linux image file **de10_nano_opencl_18.1.img** from **de10_nano_opencl_18.1.img.zip.**
2. Write the Linux image file **de10_nano_opencl_18.1.img** into the microSD card with Disk Imager.
   - Insert the microSD card into a card reader and connect it to the host PC. If the microSD card already contains an image, existing partitions will be mounted automatically. Please unmounts all the partitions.
   - Type dmesg|tail command to check which device name is assigned to the microSD card. It’s likely to be /dev/sdb (change /dev/sdb to the device name found in the previous step).
   - Run “sudo dd if=de10_Nano_opencl.img of=/dev/sdb bs=1M”
   - Run “sync”
3. Insert the microSD card into the microSD card slot (J11) of DE10-Nano.
4. The DIP switch (SW10) on DE10-Nano for MSEL[4:0] must be set to 01010.
5. Connect the host PC to the UART-to-USB port (J4) on DE10-Nano with an USB cable. Users should install the UART-to-USB device Linux driver as described in the FTDI driver download web page http://www.ftdichip.com/Drivers/VCP.html.
6. Launch Minicom utility (“minicom -s” for the configuration when it’s launched the first time)

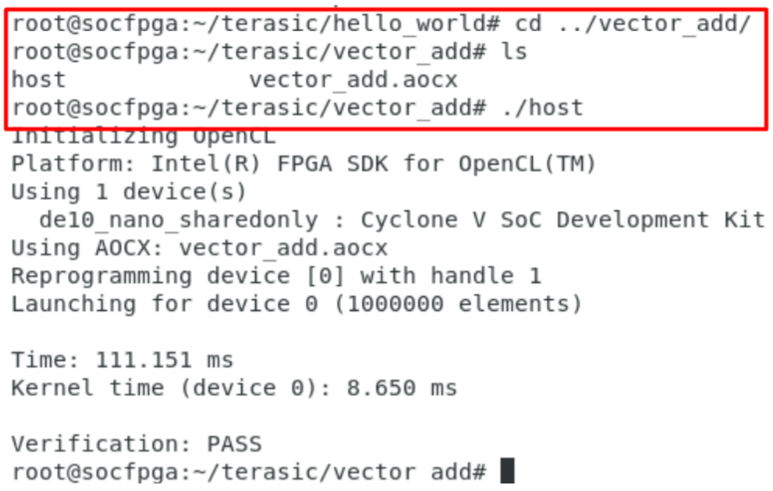
on the host PC. The Serial Device should be set to /dev/ttyUSB0, Hardware Flow Control set as NO. Modify the Modem and dialing: Set Init string, Reset string, Hang-up string to blank. And baud rate should be set to 115200. Shutdown the hardware flow control.

7. Power on DE10-Nano to boot Linux and log in as root. There's no password required.
8. Type "source ./init_opencl.sh" to load the OpenCL Linux kernel driver and setup environment variables for OpenCL Run-Time Environment that is already installed on the microSD card.
9. Launch hello_world demo:
   - Type "cd terasic/hello_world/" to change the current directory to the hello_world folder.
   - Type "ls" to see the files in this folder.
   - Type "./host" to launch the hello_world host application, as shown in **Figure 1-5**.
10. Launch vector_add demo:
    - Type "cd ../vector_add/" to change the directory to the vector_add folder.
    - Type "ls" to see the files in this folder.
    - Type "./host" to launch the vector_add host application, as shown in **Figure 1-6**.



```
root@socfpga:~# source ./init_opencl.sh
root@socfpga:~# cd terasic/hello_world/
root@socfpga:~/terasic/hello_world# ls
hello_world.aocx  host
root@socfpga:~/terasic/hello_world# ./host
Querying platform for info:
==========================
CL_PLATFORM_NAME                         = Intel(R) FPGA SDK for OpenCL(TM)
CL_PLATFORM_VENDOR                       = Intel(R) Corporation
CL_PLATFORM_VERSION                      = OpenCL 1.0 Intel(R) FPGA SDK for OpenCL(TM), Version 18.1

Querying device for info:
========================
CL_DEVICE_NAME                           = de10_nano_sharedonly : Cyclone V SoC Development Kit
CL_DEVICE_VENDOR                         = Intel(R) Corporation
CL_DEVICE_VENDOR_ID                      = 4466
CL_DEVICE_VERSION                        = OpenCL 1.0 Intel(R) FPGA SDK for OpenCL(TM), Version 18.1
CL_DRIVER_VERSION                        = 18.1
CL_DEVICE_ADDRESS_BITS                   = 64
CL_DEVICE_AVAILABLE                      = true
CL_DEVICE_ENDIAN_LITTLE                  = true
CL_DEVICE_GLOBAL_MEM_CACHE_SIZE          = 32768
CL_DEVICE_GLOBAL_MEM_CACHELINE_SIZE      = 0
CL_DEVICE_GLOBAL_MEM_SIZE                = 536870912
CL_DEVICE_IMAGE_SUPPORT                  = true
CL_DEVICE_LOCAL_MEM_SIZE                 = 16384
CL_DEVICE_MAX_CLOCK_FREQUENCY            = 1000
CL_DEVICE_MAX_COMPUTE_UNITS              = 1
CL_DEVICE_MAX_CONSTANT_ARGS              = 8
CL_DEVICE_MAX_CONSTANT_BUFFER_SIZE       = 134217728
CL_DEVICE_MAX_WORK_ITEM_DIMENSIONS       = 3
CL_DEVICE_MEM_BASE_ADDR_ALIGN            = 8192
CL_DEVICE_MIN_DATA_TYPE_ALIGN_SIZE       = 1024
CL_DEVICE_PREFERRED_VECTOR_WIDTH_CHAR    = 4
CL_DEVICE_PREFERRED_VECTOR_WIDTH_SHORT   = 2
CL_DEVICE_PREFERRED_VECTOR_WIDTH_INT     = 1
CL_DEVICE_PREFERRED_VECTOR_WIDTH_LONG    = 1
CL_DEVICE_PREFERRED_VECTOR_WIDTH_FLOAT   = 1
CL_DEVICE_PREFERRED_VECTOR_WIDTH_DOUBLE  = 0
Command queue out of order?              = false
Command queue profiling enabled?         = true
Using AOCX: hello_world.aocx
Reprogramming device [0] with handle 1
```

Figure 1-5 Hello World Demo



```
root@socfpga:~/terasic/hello_world# cd ../vector_add/
root@socfpga:~/terasic/vector_add# ls
host               vector_add.aocx
root@socfpga:~/terasic/vector_add# ./host
Initializing OpenCL
Platform: Intel(R) FPGA SDK for OpenCL(TM)
Using 1 device(s)
  de10_nano_sharedonly : Cyclone V SoC Development Kit
Using AOCX: vector_add.aocx
Reprogramming device [0] with handle 1
Launching for device 0 (1000000 elements)

Time: 111.151 ms
Kernel time (device 0): 8.650 ms

Verification: PASS
root@socfpga:~/terasic/vector_add# 
```

Figure 1-6 Vector Add Demo

Chapter 2

OpenCL on Windows

This chapter describes how to setup DE10-Nano OpenCL development environment in Windows 64-bit and then build and execute OpenCL project on DE10-Nano. For more details about getting started with Intel OpenCL for Cyclone V SoC, please refer to:

http://www.altera.com/literature/hb/opencl-sdk/aocl_c5soc_getting_started.pdf

2.1 Software Installation

This section describes how to install the software required for developing OpenCL project on DE10-Nano.

■ Install Intel Quartus Prime and OpenCL SDK

Intel Quartus Prime and OpenCL SDK are available from the website:

http:/fpgasoftware.intel.com/18.1/?edition=standard&platform=windows&download_manager=dlm3

For Quartus Prime installation, please make sure the Cyclone V device package is selected.

■ Install Intel SoC EDS

Intel SoC EDS tool is required to cross-compile the Host Program for ARM processor. The software is available from the website:

http://fpgasoftware.intel.com/soceds/18.1/?edition=standard&platform=windows&download_manager=dlm3

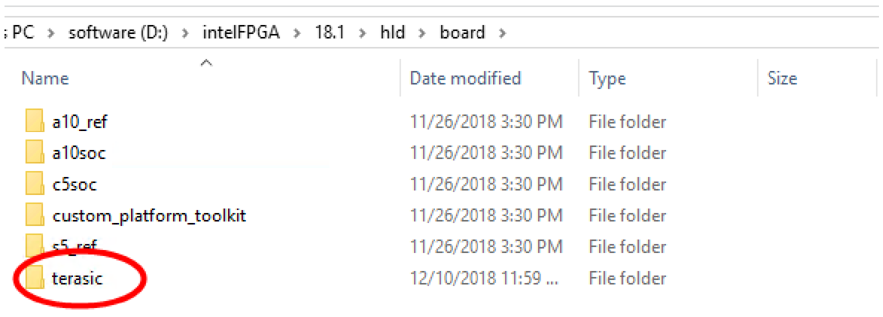
Please make sure the DS-5 is installed during the installation of SoC EDS.

■ Install DE10-Nano OpenCL Board Support Package (BSP)

After Quartus Prime and OpenCL SDK are installed, please create a new folder "terasic" under the folder "D:\intelFPGA\18.1\hld\board", assuming Intel Quartus Prime is installed under the folder "D:\intelFPGA\18.1". Please download the DE10-Nano BSP file **DE10-Nano_OpenCL_BSP_18.1.zip** from

http://de10-nano.terasic.com/cd

Please uncompress the zip file and copy the "de10_nano" folder to the "terasic" folder created previously, as shown in **Figure 2-1**.


PC › software (D:) › intelFPGA › 18.1 › hld › board ›

Name	Date modified	Type	Size
a10_ref	11/26/2018 3:30 PM	File folder	
a10soc	11/26/2018 3:30 PM	File folder	
c5soc	11/26/2018 3:30 PM	File folder	
custom_platform_toolkit	11/26/2018 3:30 PM	File folder	
s5_ref	11/26/2018 3:30 PM	File folder	
terasic	12/10/2018 11:59 ...	File folder	

Figure 2-1 Location of "D:\intelFPGA\18.1\hld\board\terasic" folder

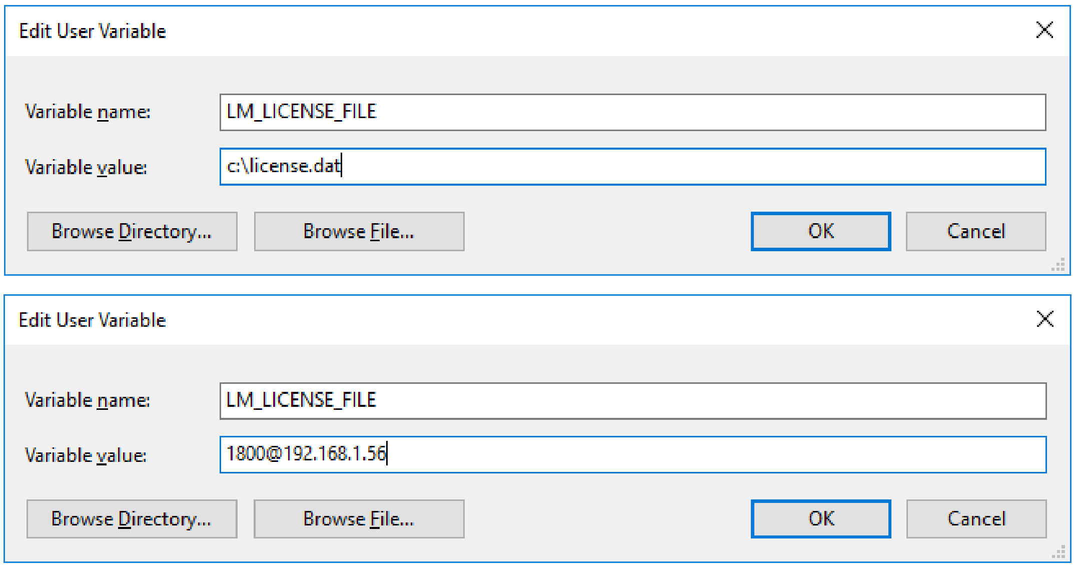
2.2 License Installation

An Quartus license is required to compile OpenCL project. Users can purchase the license from Intel. A file named "license.dat" will be given upon purchasing Quartus license. For license installation, please place the file "license.dat" in the local disk drive "c:\" and create a Windows environment variable **LM_LICENSE_FILE** with value "c:\license.dat". The value of this environment variable needs to match the actual location of "license.dat" file.

The procedures below show how to create the **LM_LICENSE_FILE** environment variable in Windows 10:

1. Right click on **Start Menu** and select **File Explorer**.
2. Right click on **This PC** and select **Properties**.
3. Select **Advanced system settings.**

4. Select **Environment Variables** from the **Advanced** tab.
5. Select **New**.
6. In the **New User Variable** dialog shown in **Figure 2-2**, type "**LM_LICENSE_FILE**" in the **Variable name** field. Then if you use a license file type "**c:\license.dat**" in the **Variable value** field, else a floating license on a license server type "port number @server IP address".


Edit User Variable

Variable name: LM_LICENSE_FILE

Variable value: c:\license.dat

Edit User Variable

Variable name: LM_LICENSE_FILE

Variable value: 1800@192.168.1.56

Figure 2-2 Setup LM_LICENSE_FILE environment variable

2.3 OpenCL Environment Configuration

Please add the following paths into the User environment variables:

■ OpenCL Configuration

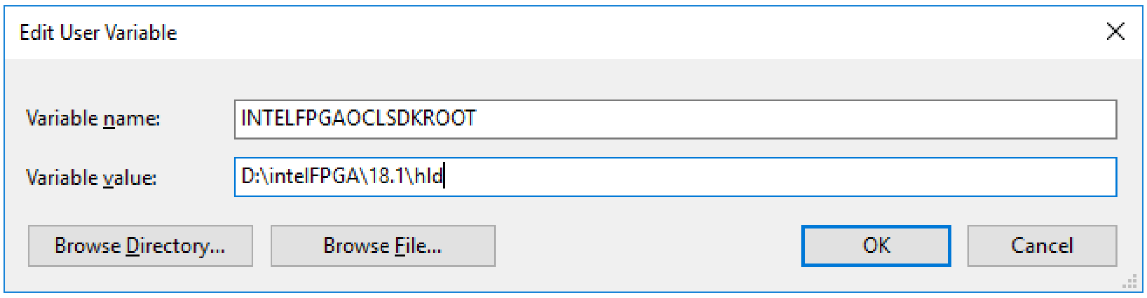
For the operating system to find the OpenCL utilities correctly, users need to create a new environment variable named INTELFPGAOCLSDKROOT and add the following paths into the PATH environment variable:

1. %INTELFPGAOCLSDKROOT%\bin
2. %INTELFPGAOCLSDKROOT%\host\windows64\bin

Here are the procedures about how to do those on Windows 10:

1. Right click on **Start Menu** and select **File Explorer**.

2. Right click on **This PC** and select **Properties**.
3. Select **Advanced system settings.**
4. Select **Environment Variables** from the **Advanced** tab.
5. Select **New** to New a User variable.
6. In the **New User Variable** dialog. Type **INTELFPGAOCLSDKROOT** in the Variable name filed and type the path where your software installed in the Variable value filed, as shown in Figure 2-3 . (**If there is a variable named ALTERAOCLSDKROOT, please delete it**)

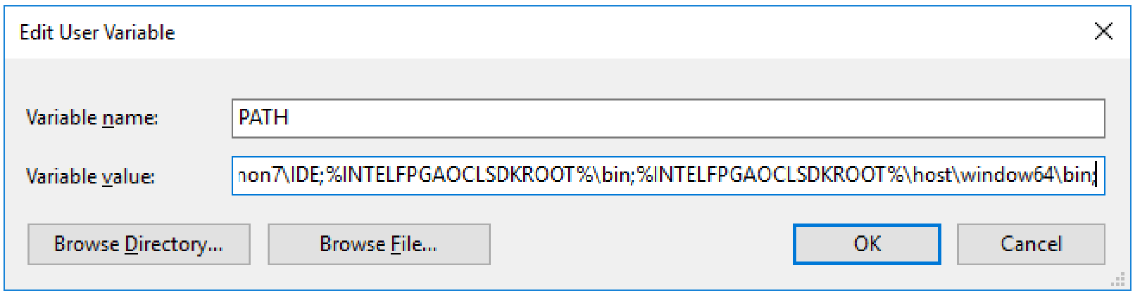

Edit User Variable

Variable name: INTELFPGAOCLSDKROOT

Variable value: D:\intelFPGA\18.1\hld

Figure 2-3 Add INTELFPGAOCLSDKROOT Environment Variable

7. Select **PATH** item in the Variable list and click the **Edit** button.
8. In the **Edit User Variable** dialog shown in Figure 2-4, add the following two strings into the **Value** edit box. The strings should be separated by the symbol ";".
   - %INTELFPGAOCLSDKROOT%\bin
   - %INTELFPGAOCLSDKROOT%\host\windows64\bin


Edit User Variable

Variable name: PATH

Variable value: ıon7\IDE;%INTELFPGAOCLSDKROOT%\bin;%INTELFPGAOCLSDKROOT%\host\window64\bin;

Figure 2-4 Modify PATH Environment Variable

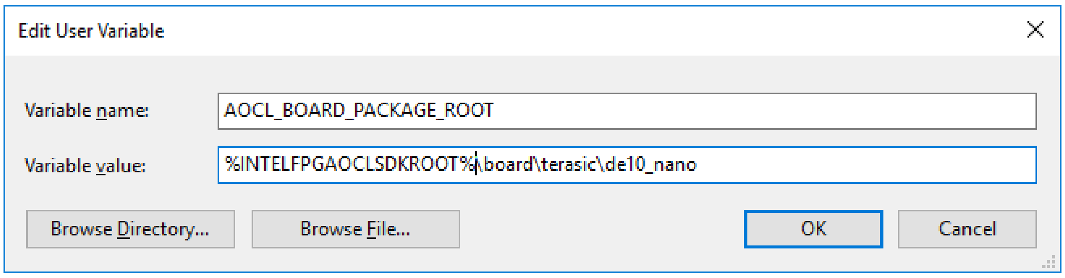
■ DE10-Nano BSP Configuration

For Intel OpenCL SDK to find the kit location of DE10-Nano correctly, Users need to create an environment variable **AOCL_BOARD_PACKAGE_ROOT** and set its value as:

“%INTELFPGAOCLSDKROOT%\board\terasic\de10_nano”

The procedures to create the required **AOCL_BOARD_PACKAGE_ROOT** environment variable in Windows 10 are:

1. Right click on **Start Menu** and select **File Explorer**.
2. Right click on **This PC** and select **Properties**.
3. Select **Advanced system settings**.
4. Select **Environment Variables** from the **Advanced** tab.
5. Select **New**.
6. In the **New User Variable** dialog shown in Figure 2-5, type “**AOCL_BOARD_PACKAGE_ROOT**” in the **Variable name** field and type “**%INTELFPGAOCLSDKROOT%\board\terasic\de10_nano**” in the **Variable value** field.


Edit User Variable

Variable name: AOCL_BOARD_PACKAGE_ROOT

Variable value: %INTELFPGAOCLSDKROOT%\board\terasic\de10_nano

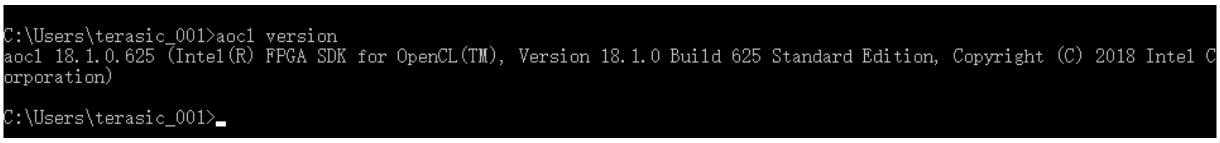
Figure 2-5 Setup AOCL_BOARD_PACKAGE_ROOT environment variable

2.4 OpenCL Environment Verification

This section shows how to confirm the OpenCL environment is setup correctly. Please open **Command Prompt** window by clicking Windows **Start** button and click **All Programs**. Click **Accessories** and then click **Command Prompt**.

■ Verify Utility

Type “aocl version” command in the command prompt window and see if the version displayed matches the number shown in Figure 2-6. If the ‘aocl’ command cannot be found, please check if the “**%INTELFPGAOCLSDKROOT**%\bin” path is added to the **PATH** environment variable correctly.

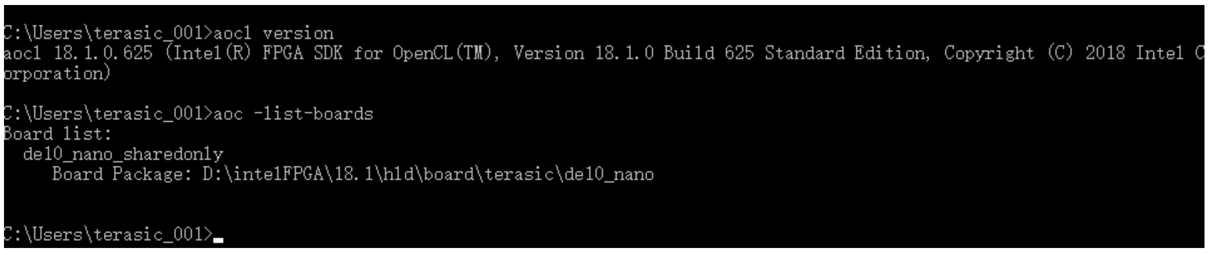


```
C:\Users\terasic_001>aocl version
aocl 18.1.0.625 (Intel(R) FPGA SDK for OpenCL(TM), Version 18.1.0 Build 625 Standard Edition, Copyright (C) 2018 Intel C
orporation)

C:\Users\terasic_001>_
```

Figure 2-6 Aocl version of Intel FPGA SDK for OpenCL in the command prompt

■ Verify Target Board

Type "aoc -list-boards" command in the command prompt window and make sure "de10_nano_sharedonly" is listed in the **Board list**, as shown in Figure 2-7. If "de10_nano _sharedonly" is not listed, please check if the **AOCL_BOARD_PACKAGE_ROOT** environment variable is assigned correctly.

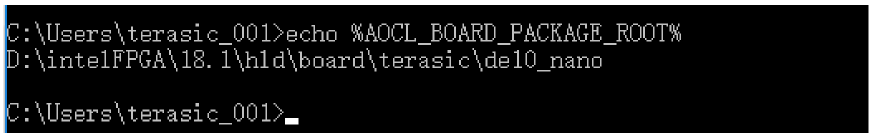


```
C:\Users\terasic_001>aocl version
aocl 18.1.0.625 (Intel(R) FPGA SDK for OpenCL(TM), Version 18.1.0 Build 625 Standard Edition, Copyright (C) 2018 Intel C
orporation)

C:\Users\terasic_001>aoc -list-boards
Board list:
  de10_nano_sharedonly
     Board Package: D:\intelFPGA\18.1\hld\board\terasic\de10_nano

C:\Users\terasic_001>_
```

Figure 2-7 'de10_nano_sharedonly' is listed in the Board list

■ How to Check Environment Variables

The value of environment variables can be retrieved by typing 'echo' in the command prompt window. For example, type "echo %AOCL_BOARD_PACKAGE_ROOT%" can retrieve the value of environment variable **AOCL_BOARD_PACKAGE_ROOT,** as shown in Figure 2-8.



```
C:\Users\terasic_001>echo %AOCL_BOARD_PACKAGE_ROOT%
D:\intelFPGA\18.1\hld\board\terasic\de10_nano

C:\Users\terasic_001>_
```

Figure 2-8 The value of AOCL_BOARD_PACKAGE_ROOT environment variable

2.5 Compile and Execute OpenCL Project

This section shows how to compile and execute OpenCL kernel and OpenCL Host Program provided in the DE10-Nano BSP. Users can follow the same procedures to compile and execute other OpenCL examples for DE10-Nano.

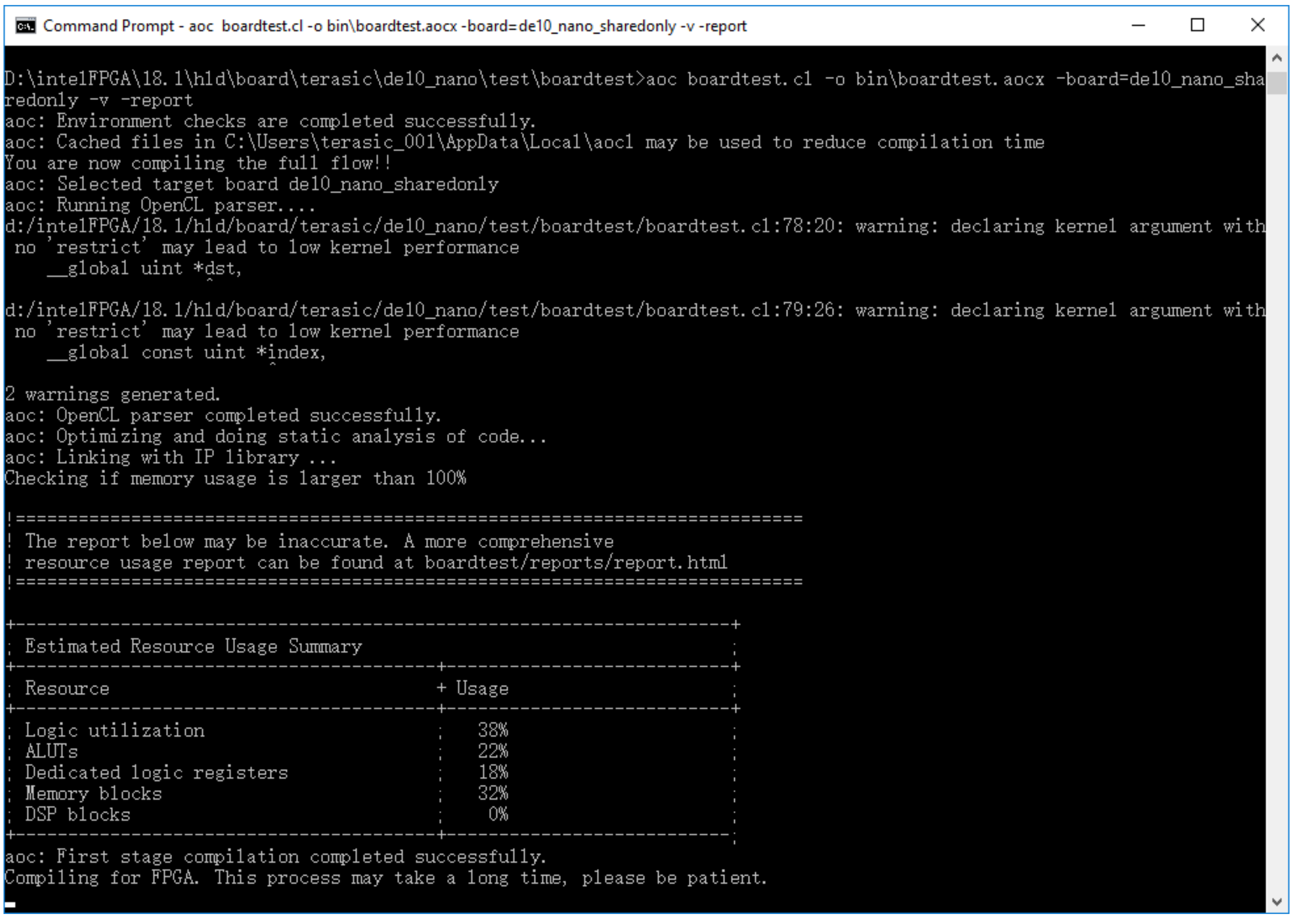
Compile OpenCL Kernel

The utility **aoc** (Intel SDK for OpenCL Kernel Compiler) is used to compile OpenCL kernel. Type "d:&cd D:\intelFPGA\18.1\hld\board\terasic\de10_nano\test\boardtest" in the command prompt window to change the current directory to the folder **boardtest** and then type:

"aoc boardtest.cl -o bin\boardtest.aocx -board=de10_nano_sharedonly -v -report"

to compile the OpenCL kernel. It will take approximately half an hour to complete the compilation. When the compilation process is complete, an OpenCL image file **boardtest.aocx** is generated under the **bin** folder. **Figure 2-9** shows the OpenCL kernel is compiling. For more details about the usage of **aoc**, please refer to the **Intel SDK for OpenCL Programming Guide**:

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


Command Prompt - aoc boardtest.cl -o bin\boardtest.aocx -board=de10_nano_sharedonly -v -report

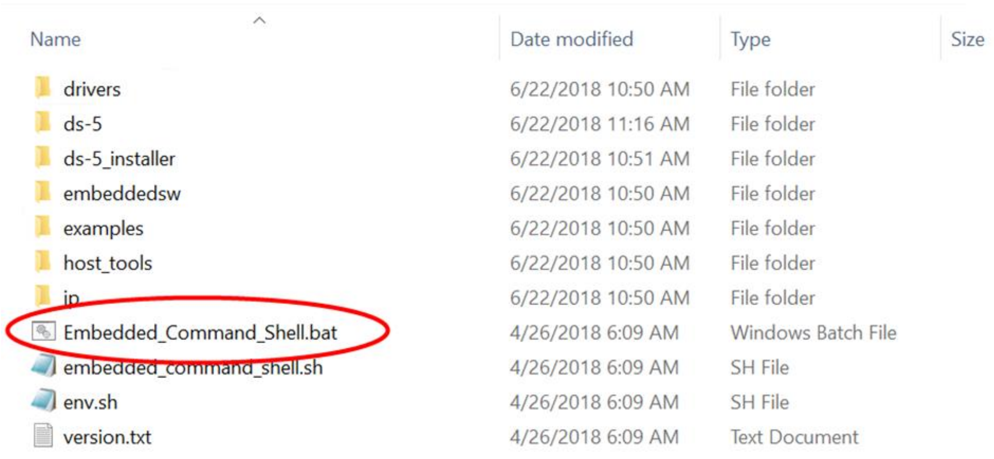
```
D:\intelFPGA\18.1\hld\board\terasic\de10_nano\test\boardtest>aoc boardtest.cl -o bin\boardtest.aocx -board=de10_nano_sha
redonly -v -report
aoc: Environment checks are completed successfully.
aoc: Cached files in C:\Users\terasic_001\AppData\Local\aocl may be used to reduce compilation time
You are now compiling the full flow!!
aoc: Selected target board de10_nano_sharedonly
aoc: Running OpenCL parser....
d:/intelFPGA/18.1/hld/board/terasic/de10_nano/test/boardtest/boardtest.cl:78:20: warning: declaring kernel argument with
 no 'restrict' may lead to low kernel performance
    __global uint *dst,
                   ^

d:/intelFPGA/18.1/hld/board/terasic/de10_nano/test/boardtest/boardtest.cl:79:26: warning: declaring kernel argument with
 no 'restrict' may lead to low kernel performance
    __global const uint *index,
                         ^

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%

!===========================================================================
! The report below may be inaccurate. A more comprehensive
! resource usage report can be found at boardtest/reports/report.html
!===========================================================================

+--------------------------------------------------------------------+
; Estimated Resource Usage Summary                                   ;
+----------------------------------------+---------------------------+
; Resource                               + Usage                     ;
+----------------------------------------+---------------------------+
; Logic utilization                      ;   38%                     ;
; ALUTs                                  ;   22%                     ;
; Dedicated logic registers              ;   18%                     ;
; Memory blocks                          ;   32%                     ;
; DSP blocks                             ;    0%                     ;
+----------------------------------------+---------------------------;
aoc: First stage compilation completed successfully.
Compiling for FPGA. This process may take a long time, please be patient.
_
```

Figure 2-9 Screenshot of OpenCL kernel is compiling

Compile Host Program

The Host Program is compiled in Intel SoC EDS. Please launch embedded command shell by executing the "Embedded_Command_Shell.bat", as shown in **Figure 2-10**, under the folder

“D:\intelFPGA\18.1\embedded”, assuming Intel SoC EDS is installed under the directory “D:\intelFPGA\18.1”



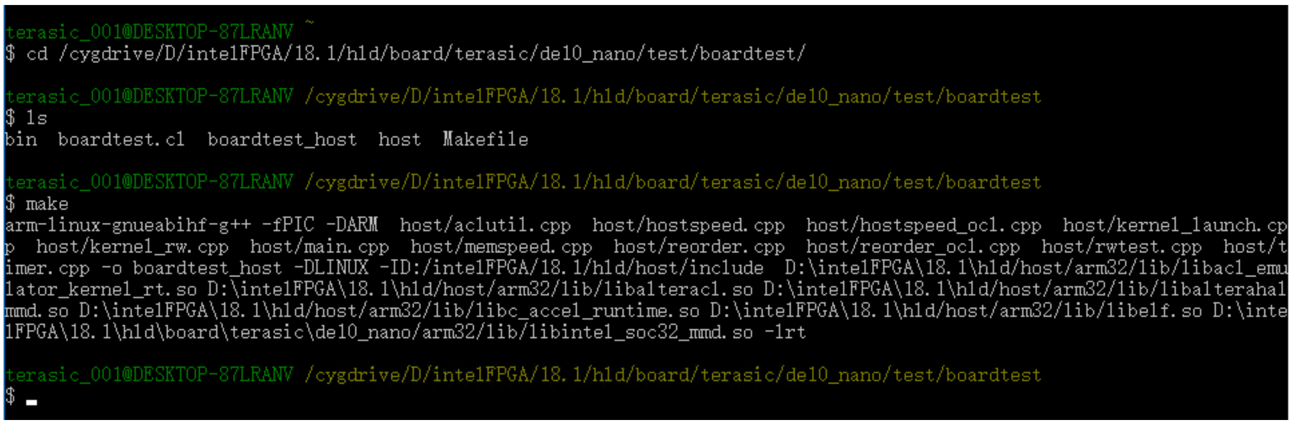
Name	Date modified	Type	Size
drivers	6/22/2018 10:50 AM	File folder	
ds-5	6/22/2018 11:16 AM	File folder	
ds-5_installer	6/22/2018 10:51 AM	File folder	
embeddedsw	6/22/2018 10:50 AM	File folder	
examples	6/22/2018 10:50 AM	File folder	
host_tools	6/22/2018 10:50 AM	File folder	
ip	6/22/2018 10:50 AM	File folder	
Embedded_Command_Shell.bat	4/26/2018 6:09 AM	Windows Batch File	
embedded_command_shell.sh	4/26/2018 6:09 AM	SH File	
env.sh	4/26/2018 6:09 AM	SH File	
version.txt	4/26/2018 6:09 AM	Text Document	

Figure 2-10 Location of Embedded_Command_Shell.bat

Type the following in the command shell:

“cd /cygdrive/D/intelFPGA/18.1/hld/board/terasic/de10_nano/test/boardtest/”

to change the current directory to the folder where the boardtest project is located. Type “make” to build the host project shown in Figure 2-11. If the compilation is successful, a binary file **boardtest_host** will be generated under the boardtest folder.



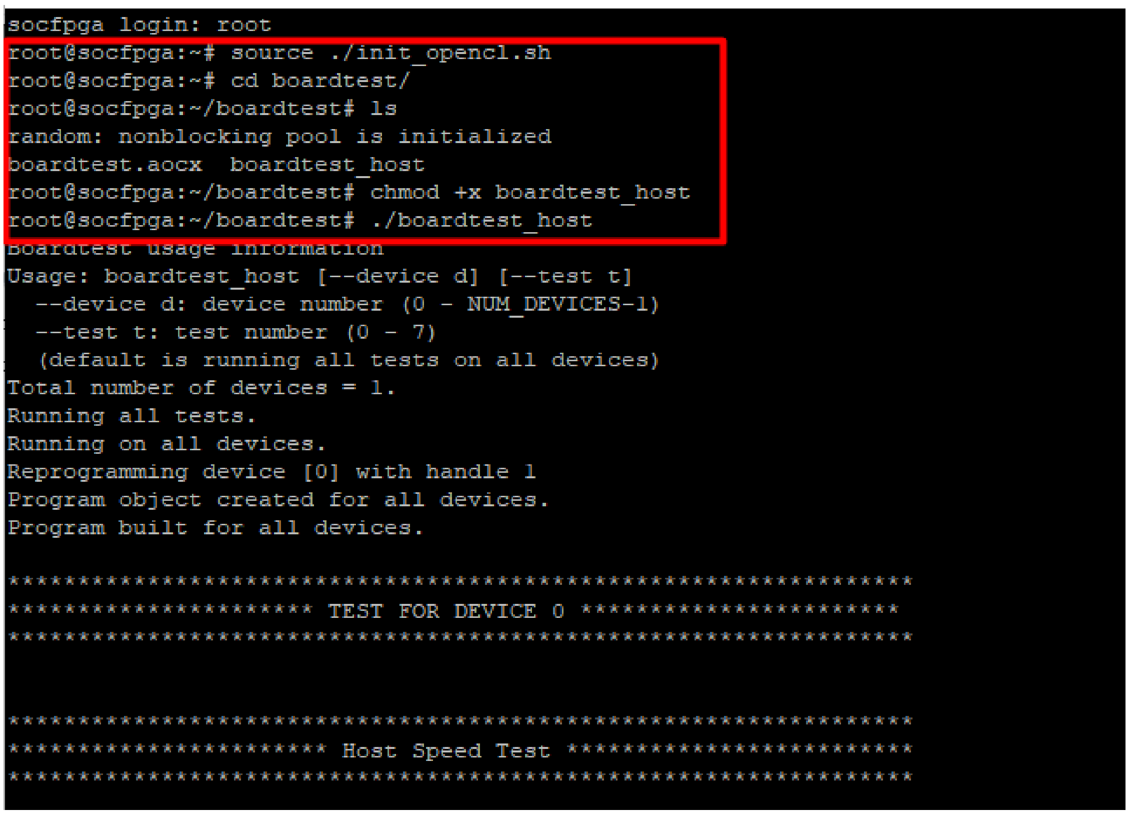
```
terasic_001@DESKTOP-87LRANV ~
$ cd /cygdrive/D/intelFPGA/18.1/hld/board/terasic/de10_nano/test/boardtest/

terasic_001@DESKTOP-87LRANV /cygdrive/D/intelFPGA/18.1/hld/board/terasic/de10_nano/test/boardtest
$ ls
bin  boardtest.cl  boardtest_host  host  Makefile

terasic_001@DESKTOP-87LRANV /cygdrive/D/intelFPGA/18.1/hld/board/terasic/de10_nano/test/boardtest
$ make
arm-linux-gnueabihf-g++ -fPIC -DARM  host/aclutil.cpp  host/hostspeed.cpp  host/hostspeed_ocl.cpp  host/kernel_launch.cp
p  host/kernel_rw.cpp  host/main.cpp  host/memspeed.cpp  host/reorder.cpp  host/reorder_ocl.cpp  host/rwtest.cpp  host/t
imer.cpp -o boardtest_host -DLINUX -ID:/intelFPGA/18.1/hld/host/include  D:\intelFPGA\18.1\hld/host/arm32/lib/libacl_emu
lator_kernel_rt.so D:\intelFPGA\18.1\hld/host/arm32/lib/libalteracl.so D:\intelFPGA\18.1\hld/host/arm32/lib/libalterahal
mmd.so D:\intelFPGA\18.1\hld/host/arm32/lib/libc_accel_runtime.so D:\intelFPGA\18.1\hld/host/arm32/lib/libelf.so D:\inte
lFPGA\18.1\hld\board\terasic\de10_nano/arm32/lib/libintel_soc32_mmd.so -lrt

terasic_001@DESKTOP-87LRANV /cygdrive/D/intelFPGA/18.1/hld/board/terasic/de10_nano/test/boardtest
$ 
```

Figure 2-11 Type ‘make’ to build the boardtest host project

■ Execute BoardTest Project

Please boot DE10-Nano with the Linux image generated from Chapter 1.4 for DE10-Nano OpenCL. Users need to copy both the kernel file **boardtest.aocx** and host file **boardtest_host** generated from the previous section from the host PC to Linux running on DE10-Nano. Users can copy the file by typing Linux “scp” command through Ethernet or USB storage.

After these two files are copied to Linux running on DE10-Nano, please go to the terminal and type “source ./init_opencl.sh” to setup OpneCL environment. Type “chmod +x boardtest_host” to add execution attribute to the host file. Type “./boardtest_host” to launch the host application, as shown in **Figure 2-12.**



```
socfpga login: root
root@socfpga:~# source ./init_opencl.sh
root@socfpga:~# cd boardtest/
root@socfpga:~/boardtest# ls
random: nonblocking pool is initialized
boardtest.aocx  boardtest_host
root@socfpga:~/boardtest# chmod +x boardtest_host
root@socfpga:~/boardtest# ./boardtest_host
Boardtest usage information
Usage: boardtest_host [--device d] [--test t]
  --device d: device number (0 - NUM_DEVICES-1)
  --test t: test number (0 - 7)
  (default is running all tests on all devices)
Total number of devices = 1.
Running all tests.
Running on all devices.
Reprogramming device [0] with handle 1
Program object created for all devices.
Program built for all devices.

*****************************************************************
********************** TEST FOR DEVICE 0 ************************
*****************************************************************


*****************************************************************
*********************** Host Speed Test *************************
*****************************************************************
```

Figure 2-12 BoardTest host is executed successfully

Chapter 3
OpenCL on Linux

This chapter describes how to setup the environment for the development of OpenCL on Linux, build OpenCL project including kernel and host application, followed by execution and verification of OpenCL project.

For more details about OpenCL on Linux, please refer to the Getting Started Guild of Intel OpenCL Cyclone V SoC:

http://www.altera.com/literature/hb/opencl-sdk/aocl_c5soc_getting_started.pdf

3.1 Software Installation

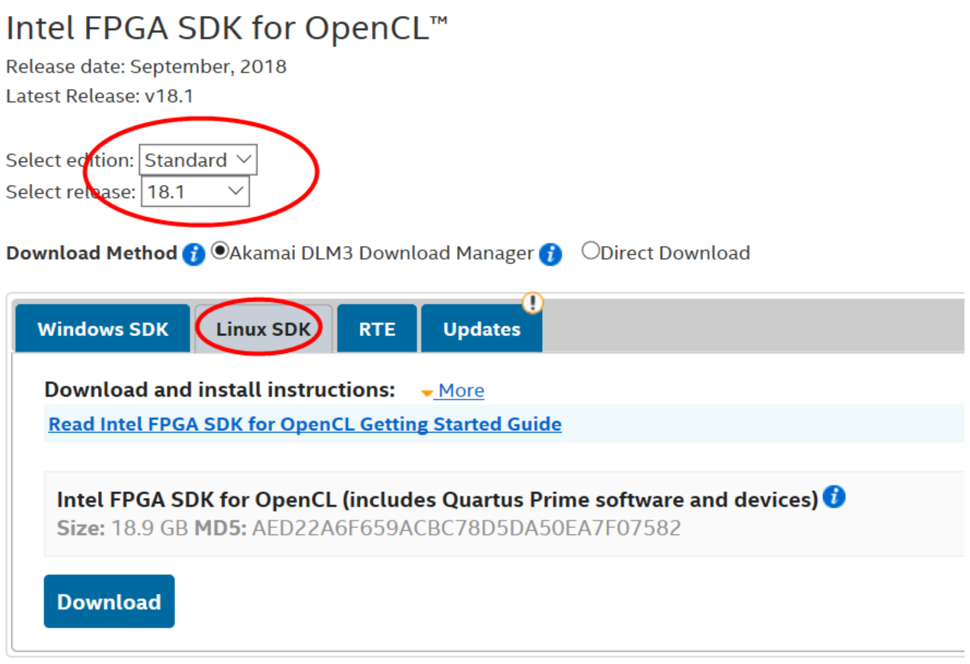
This section describes where to download and how to install the software required for OpenCL on DE10-Nano.

■ Download Intel Quartus Prime and OpenCL SDK

Intel Quartus Prime and OpenCL SDK are available from the website of Intel PSG:

http://fpgasoftware.intel.com/opencl/18.1/?edition=standard&download_manager=dlm3

Follow the link and select Linux operation system with version 18.1, as shown in **Figure 3-1**.


Intel FPGA SDK for OpenCL™
Release date: September, 2018
Latest Release: v18.1

Select edition: Standard
Select release: 18.1

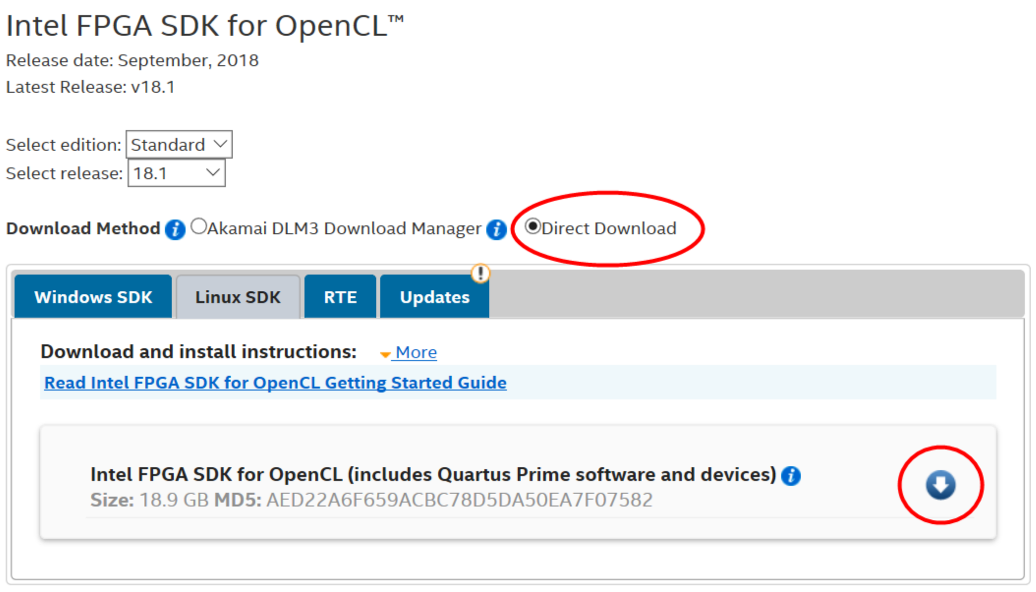
Download Method [✓ Akamai DLM3 Download Manager] Direct Download

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: 18.9 GB MD5: AED22A6F659ACBC78D5DA50EA7F07582

Figure 3-1 Download the Linux version of OpenCL SDK v18.1 from the website of Intel PSG

Choose Direct Download as the download method since the download manager is for Windows only. Click the arrow to download Intel FPGA Design Software, as shown in Figure 3-2., and make sure the Cyclone V device is included. Users can download Intel SDK for OpenCL as standalone installer or RPM package.


Intel FPGA SDK for OpenCL™
Release date: September, 2018
Latest Release: v18.1

Select edition: Standard
Select release: 18.1

Download Method Akamai DLM3 Download Manager [✓ Direct Download]

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: 18.9 GB MD5: AED22A6F659ACBC78D5DA50EA7F07582

Figure 3-2 Download Intel OpenCL SDK Linux version with Cyclone V device support

■ Install Intel SoC EDS

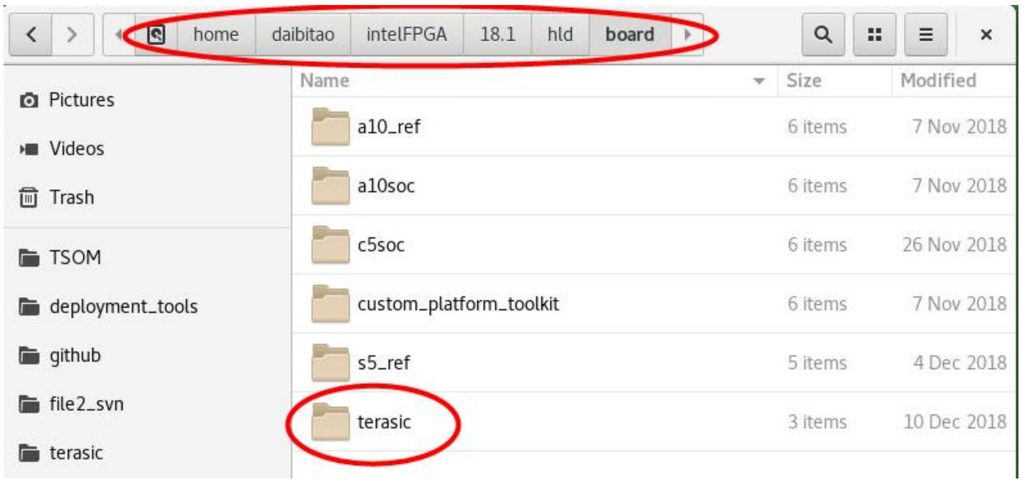
Intel SoC EDS tool is required to cross-compile the host program for ARM processor. The software is available from the website:

http://fpgasoftware.intel.com/soceds/18.1/?edition=standard&platform=linux&download_manager=direct

Please make sure DS-5 is installed during the installation of SoC EDS.

■ Install DE10-Nano OpenCL Board Support Package (BSP)

After Quartus Prime and OpenCL SDK are installed, please create a new folder named as the terasic to the Intel OpenCL SDK folder, my path is "/home/daibitao/intelFPGA/18.1/hld/board/", refer to your own path, and uncompressing the OpenCL BSP to this folder. as shown in **Figure 3-3**.



Name	Size	Modified
a10_ref	6 items	7 Nov 2018
a10soc	6 items	7 Nov 2018
c5soc	6 items	26 Nov 2018
custom_platform_toolkit	6 items	7 Nov 2018
s5_ref	5 items	4 Dec 2018
terasic	3 items	10 Dec 2018

Figure 3-3 Copy the terasic folder to the intelFPGA/18.1/hld/board folder

3.2 OpenCL License Installation

A license for Quartus is required to compile OpenCL project. Users can purchase the license from Intel. After users have obtained a license file named "license.dat", it needs to be placed in the local disk such as "/home/daibitao/intelFPGA/18.1/hld/license.dat". Users also need to create an environment variable **LM_LICENSE_FILE** and set its value as "/home/daibitao/intelFPGA/18.1/hld/license.dat", which corresponds to the actual location of the license file. If you use a floating license on a license server, please set its value as "port number@server IP address”

The next section will describe how to setup the license environment.

3.3 Configuration of Environment Variables

If users install the Intel FPGA development software and OpenCL SDK on a system that does not contain any .cshrc or bash resource file (.bashrc) in the directory, the INTELFPGAOCLSDKROOT and PATH environment variables must be set manually. Users also need to create an environment variable **AOCL_BOARD_PACKAGE_ROOT** for Intel OpenCL SDK to find the kit location of DE10-Nano correctly. The value of this environment variable needs to be set as:

“$INTELFPGAOCLSDKROOT/board/terasic/de10_nano”

Alternatively, users can edit the “/etc/profile”, and append the environment variables to it. It can be done by typing “*gedit /etc/profile”* command in the terminal to open the **profile** file with the **gedit** editor tool and append the following settings to the **profile** file, you can change ROOT to your own path. After the edit is complete, save the file and type “*source /etc/profile”* command in the terminal to apply the settings.

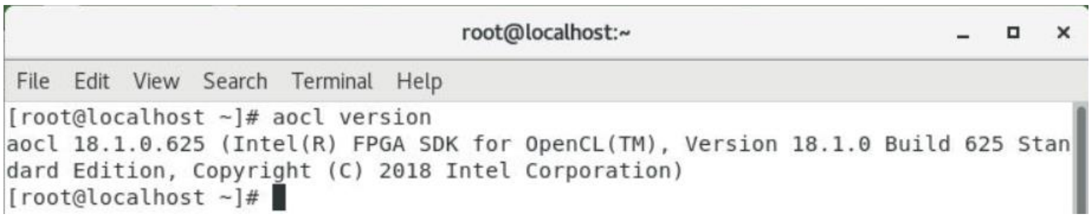
```
export ROOT=/home/daibitao
export QUARTUS_ROOTDIR=$ROOT/intelFPGA/18.1/quartus
export INTELFPGAOCLSDKROOT=$ROOT/intelFPGA/18.1/hld
export                PATH=$PATH:$QUARTUS_ROOTDIR/bin:$ROOT/intelFPGA/18.1/embedded/ds-5/bin:
$ROOT/intelFPGA/18.1/embedded/ds-5/sw/gcc/bin:$INTELFPGAOCLSDKROOT/bin:$INTELFPGAOCLSDKRO
OT/arm32/bin
export LD_LIBRARY_PATH=$INTELFPGAOCLSDKROOT/arm32/lib
export AOCL_BOARD_PACKAGE_ROOT=$INTELFPGAOCLSDKROOT/board/terasic/de10_nano
export QUARTUS_64BIT=1
export LM_LICENSE_FILE=$ROOT/intelFPGA/18.1/hld/license.dat
```

3.4 Verification of OpenCL Environment

This section shows how to make sure the OpenCL environment is setup correctly. Please open the terminal window in Linux.

■ Verify Utility

Type "aocl version" command in the terminal and make sure the aocl version reported matches the information shown in Figure 3-4. If the 'aocl' command cannot be found, please check if the "$INTELFPGAOCLSDKROOT/bin" path is added to the **PATH** environment variable correctly.

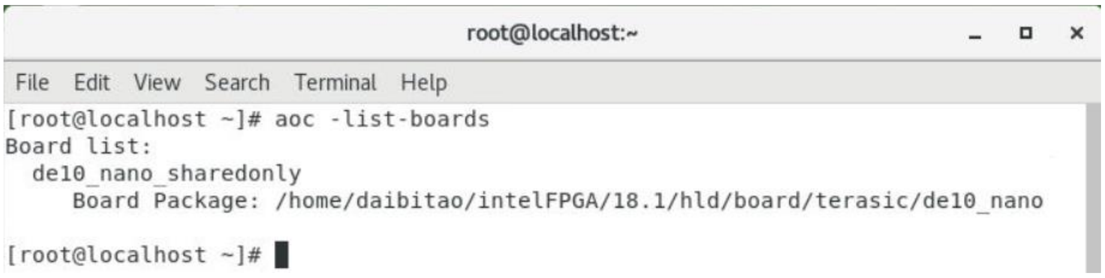


```
[root@localhost ~]# aocl version
aocl 18.1.0.625 (Intel(R) FPGA SDK for OpenCL(TM), Version 18.1.0 Build 625 Stan
dard Edition, Copyright (C) 2018 Intel Corporation)
[root@localhost ~]# █
```

Figure 3-4 The information about aocl version

■ Verify Target Board

Type "aoc -list-boards" command in the terminal and make sure "de10_nano_sharedonly" is displayed in the Board list, as shown in Figure 3-5. If "de10_nano_sharedonly" is not listed, please check if the **AOCL_BOARD_PACKAGE_ROOT** environment variable is assigned correctly.

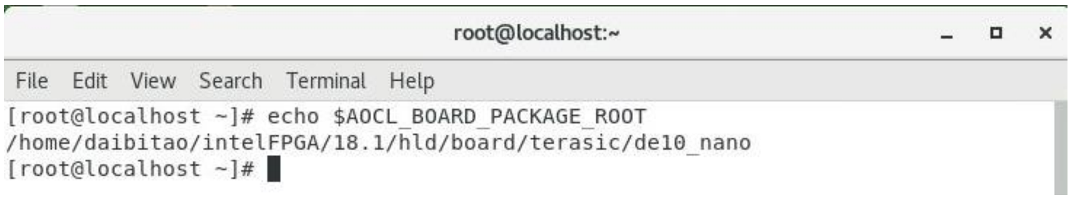


```
[root@localhost ~]# aoc -list-boards
Board list:
  de10_nano_sharedonly
     Board Package: /home/daibitao/intelFPGA/18.1/hld/board/terasic/de10_nano

[root@localhost ~]# █
```

Figure 3-5 'de10_nano_sharedonly' is shown in the Board list

■ Check Environment Variables

The values of environment variables can be retrieved by typing 'echo' command in the terminal. For example, type "echo $AOCL_BOARD_PACKAGE_ROOT" can dump the value of environment variable **AOCL_BOARD_PACKAGE_ROOT**, as shown in Figure 3-6.



```
[root@localhost ~]# echo $AOCL_BOARD_PACKAGE_ROOT
/home/daibitao/intelFPGA/18.1/hld/board/terasic/de10_nano
[root@localhost ~]# █
```

Figure 3-6 Check the environment variable AOCL_BOARD_PACKAGE_ROOT

3.5 Build and Execute OpenCL Project

This section shows how to compile and test OpenCL example project and OpenCL host program provided in DE10-Nano BSP. Users can follow the same procedures to compile and test other OpenCL projects for DE10-Nano.

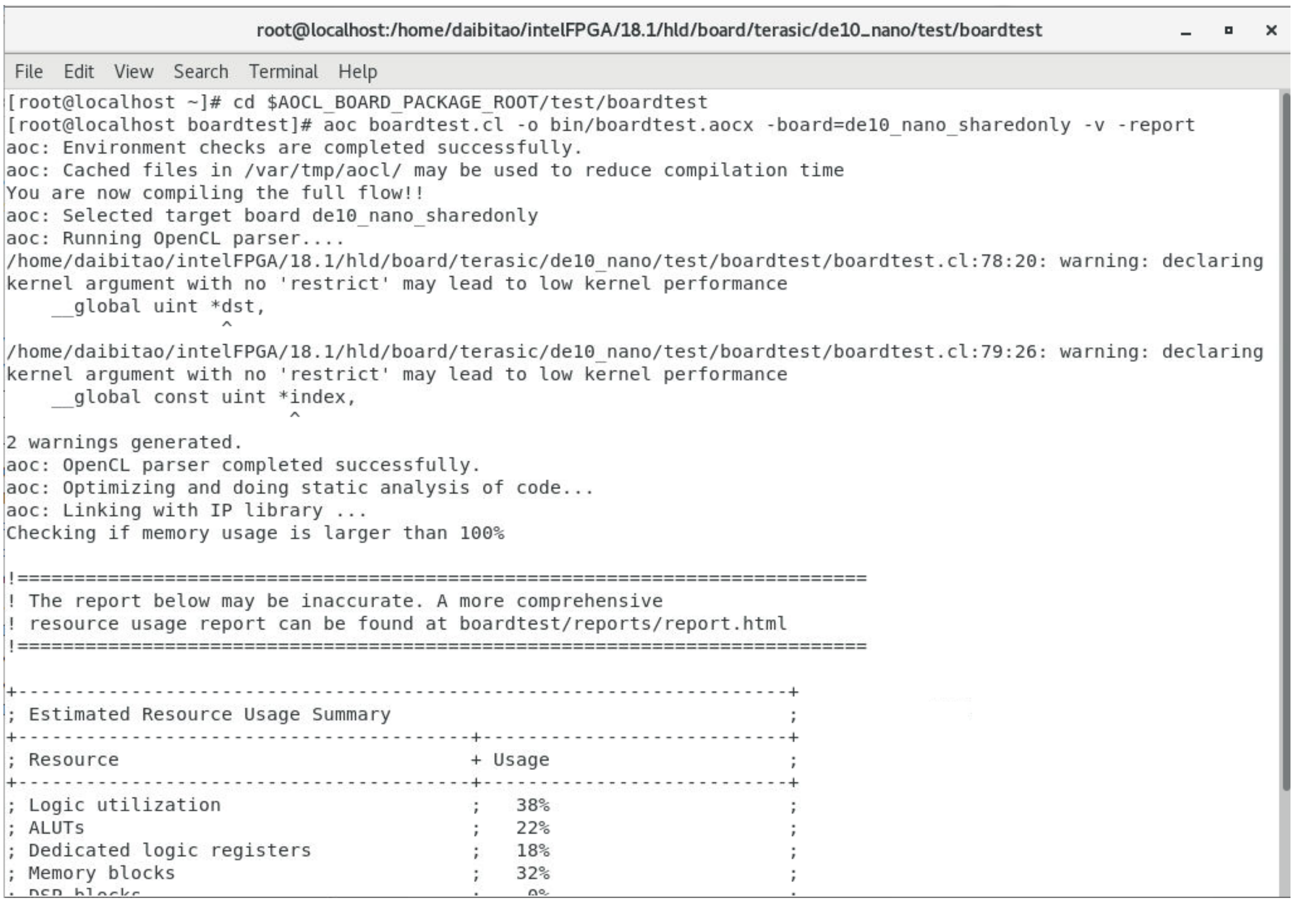
■ Compile OpenCL Kernel

The utility **aoc** (Intel SDK for OpenCL Kernel Compiler) is used to compile OpenCL kernel. Type "*cd $*AOCL_BOARD_PACKAGE_ROOT*/test/boardtest*" in the terminal to change the current directory to the **boardtest** project folder. Type

"aoc boardtest.cl -o bin/boardtest.aocx -board=de10_nano_sharedonly -v -report"

to compile the OpenCL kernel. The compilation process will take about one hour. When the compilation process is complete, an OpenCL image file **boardtest.aocx** is generated under the **bin** folder. **Figure 3-7** shows the screenshot after the OpenCL kernel is compiled successfully. For more information about the usage of **aoc**, please refer to the **Intel SDK for OpenCL Programming Guide**:

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

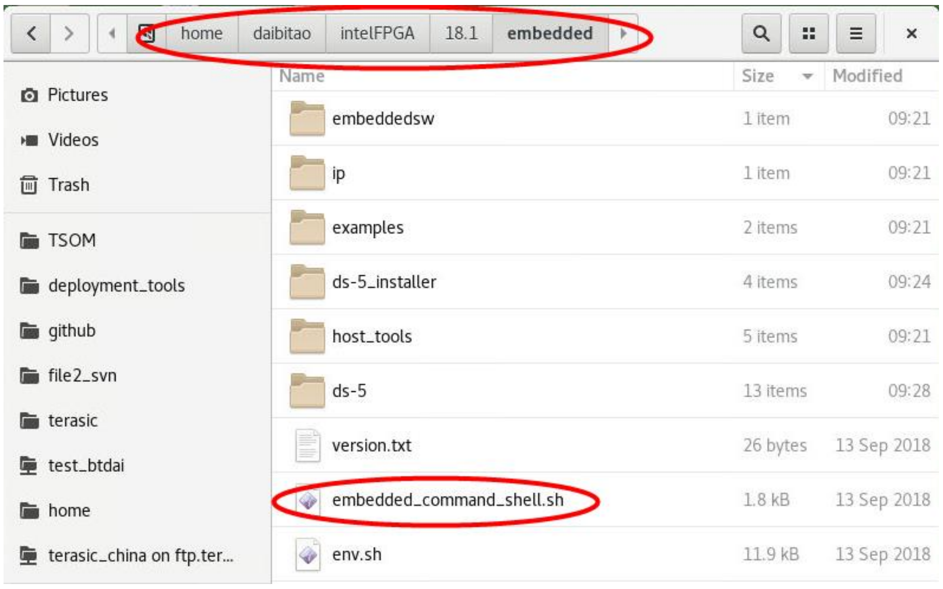

root@localhost:/home/daibitao/intelFPGA/18.1/hld/board/terasic/de10_nano/test/boardtest
File Edit View Search Terminal Help

```
[root@localhost ~]# cd $AOCL_BOARD_PACKAGE_ROOT/test/boardtest
[root@localhost boardtest]# aoc boardtest.cl -o bin/boardtest.aocx -board=de10_nano_sharedonly -v -report
aoc: Environment checks are completed successfully.
aoc: Cached files in /var/tmp/aocl/ may be used to reduce compilation time
You are now compiling the full flow!!
aoc: Selected target board de10_nano_sharedonly
aoc: Running OpenCL parser....
/home/daibitao/intelFPGA/18.1/hld/board/terasic/de10_nano/test/boardtest/boardtest.cl:78:20: warning: declaring
kernel argument with no 'restrict' may lead to low kernel performance
    __global uint *dst,
                   ^

/home/daibitao/intelFPGA/18.1/hld/board/terasic/de10_nano/test/boardtest/boardtest.cl:79:26: warning: declaring
kernel argument with no 'restrict' may lead to low kernel performance
    __global const uint *index,
                         ^

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%

!===========================================================================
! The report below may be inaccurate. A more comprehensive
! resource usage report can be found at boardtest/reports/report.html
!===========================================================================

+--------------------------------------------------------------------+
; Estimated Resource Usage Summary                                   ;
+----------------------------------------+---------------------------+
; Resource                               + Usage                     ;
+----------------------------------------+---------------------------+
; Logic utilization                      ;   38%                     ;
; ALUTs                                  ;   22%                     ;
; Dedicated logic registers              ;   18%                     ;
; Memory blocks                          ;   32%                     ;
```

Figure 3-7 OpenCL kernel is compiled successfully

■ Compile Host Program

Intel SoC EDS is used to compile the host program. Please launch the embedded command shell by executing the "Embedded_Command_Shell.sh", as shown in **Figure 3-8**, under the folder "/home/daibitao/intelFPGA/18.1/embedded", assuming Intel SoC EDS is installed under the directory "/home/daibitao/intelFPGA/18.1/".



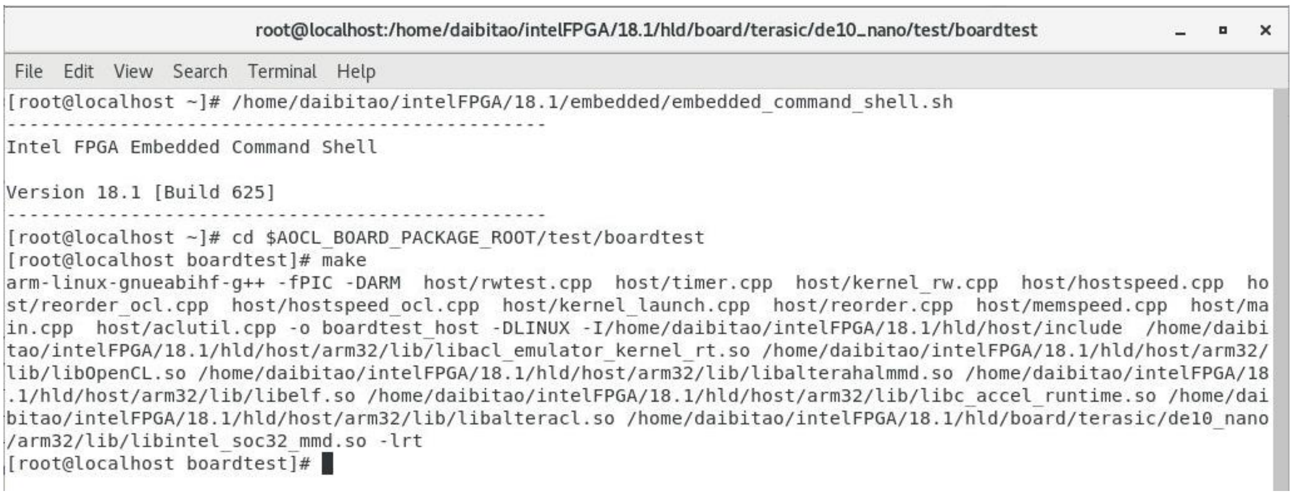
Name	Size	Modified
embeddedsw	1 item	09:21
ip	1 item	09:21
examples	2 items	09:21
ds-5_installer	4 items	09:24
host_tools	5 items	09:21
ds-5	13 items	09:28
version.txt	26 bytes	13 Sep 2018
embedded_command_shell.sh	1.8 kB	13 Sep 2018
env.sh	11.9 kB	13 Sep 2018

Figure 3-8 Location of Embedded_Command_Shell.sh

In the command shell, please type:

“cd $AOCL_BOARD_PACKAGE_ROOT/test/boardtest”

and the current directory will be changed to the folder where the boardtest project is located. Type “make” to build the host project, as shown in **Figure 3-9**. If the host project is compiled successfully, a host binary file **boardtest_host** will be generated under the boardtest folder.


root@localhost:/home/daibitao/intelFPGA/18.1/hld/board/terasic/de10_nano/test/boardtest
File Edit View Search Terminal Help

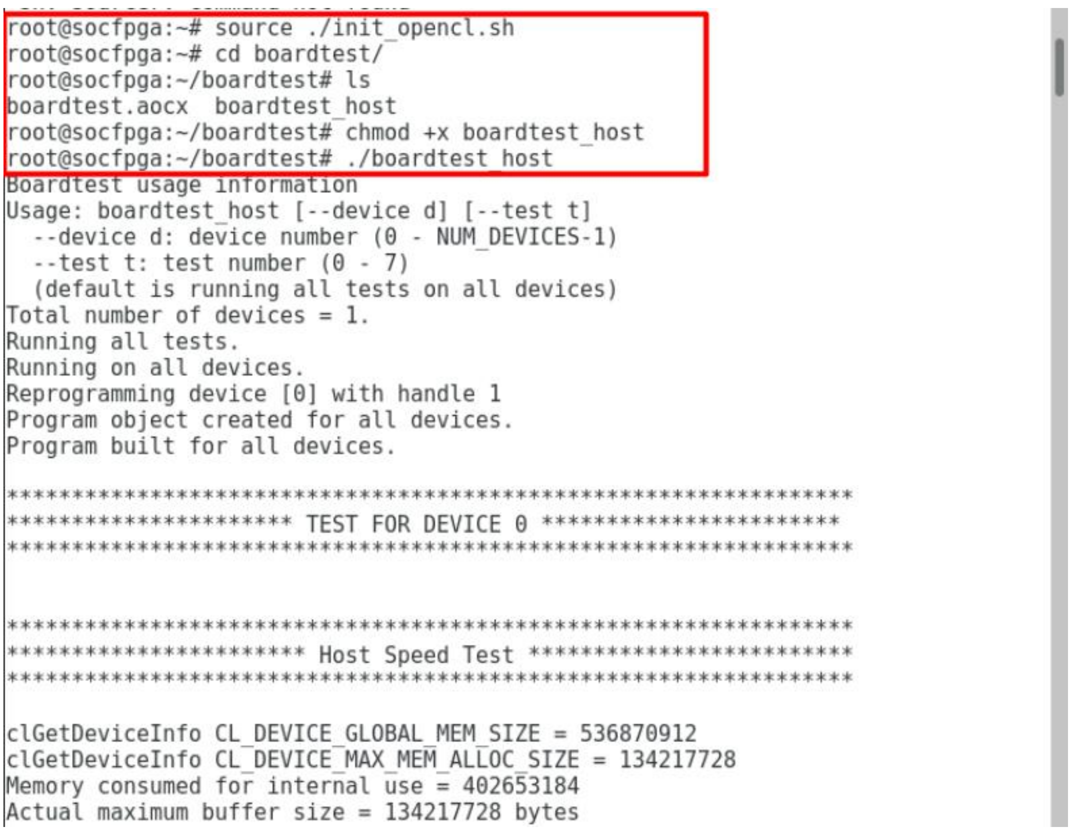
```
[root@localhost ~]# /home/daibitao/intelFPGA/18.1/embedded/embedded_command_shell.sh
------------------------------------------------
Intel FPGA Embedded Command Shell

Version 18.1 [Build 625]
------------------------------------------------
[root@localhost ~]# cd $AOCL_BOARD_PACKAGE_ROOT/test/boardtest
[root@localhost boardtest]# make
arm-linux-gnueabihf-g++ -fPIC -DARM  host/rwtest.cpp  host/timer.cpp  host/kernel_rw.cpp  host/hostspeed.cpp  ho
st/reorder_ocl.cpp  host/hostspeed_ocl.cpp  host/kernel_launch.cpp  host/reorder.cpp  host/memspeed.cpp  host/ma
in.cpp  host/aclutil.cpp -o boardtest_host -DLINUX -I/home/daibitao/intelFPGA/18.1/hld/host/include  /home/daibi
tao/intelFPGA/18.1/hld/host/arm32/lib/libacl_emulator_kernel_rt.so /home/daibitao/intelFPGA/18.1/hld/host/arm32/
lib/libOpenCL.so /home/daibitao/intelFPGA/18.1/hld/host/arm32/lib/libalterahalmmd.so /home/daibitao/intelFPGA/18
.1/hld/host/arm32/lib/libelf.so /home/daibitao/intelFPGA/18.1/hld/host/arm32/lib/libc_accel_runtime.so /home/dai
bitao/intelFPGA/18.1/hld/host/arm32/lib/libalteracl.so /home/daibitao/intelFPGA/18.1/hld/board/terasic/de10_nano
/arm32/lib/libintel_soc32_mmd.so -lrt
[root@localhost boardtest]# █
```

Figure 3-9 Type ‘make’ to build the boardtest host project

■ Test Boardtest project

Please boot DE10-Nano with the Linux image for DE10-Nano OpenCL described in Chapter 1.4. Users need to copy the generated kernel file **boardtest.aocx** and the host file **boardtest_host** from the host PC to the Linux system running on DE10-Nano. This can be done by establishing SSH connection via Ethernet with "scp" command or via usb-storage with "mount" command.

After these two files are copied to the Linux system running on DE10-Nano, please add executable attribution with "chmod +x boardtest_host" command and type "source ./init_opencl.sh" to setup OpneCL environment, then type "./boardtest_host" to launch the host application as shown in Figure 3-10.



```
root@socfpga:~# source ./init_opencl.sh
root@socfpga:~# cd boardtest/
root@socfpga:~/boardtest# ls
boardtest.aocx  boardtest_host
root@socfpga:~/boardtest# chmod +x boardtest_host
root@socfpga:~/boardtest# ./boardtest_host
Boardtest usage information
Usage: boardtest_host [--device d] [--test t]
  --device d: device number (0 - NUM_DEVICES-1)
  --test t: test number (0 - 7)
  (default is running all tests on all devices)
Total number of devices = 1.
Running all tests.
Running on all devices.
Reprogramming device [0] with handle 1
Program object created for all devices.
Program built for all devices.

*****************************************************************
********************** TEST FOR DEVICE 0 ***********************
*****************************************************************


*****************************************************************
*********************** Host Speed Test ************************
*****************************************************************

clGetDeviceInfo CL_DEVICE_GLOBAL_MEM_SIZE = 536870912
clGetDeviceInfo CL_DEVICE_MAX_MEM_ALLOC_SIZE = 134217728
Memory consumed for internal use = 402653184
Actual maximum buffer size = 134217728 bytes
```

Figure 3-10 Test BoardTest project

Revision History

Version	*Change Log*
V2.0	**Modify content for OpenCL 18.1**
V1.0	**Initial Version**

Copyright Statement