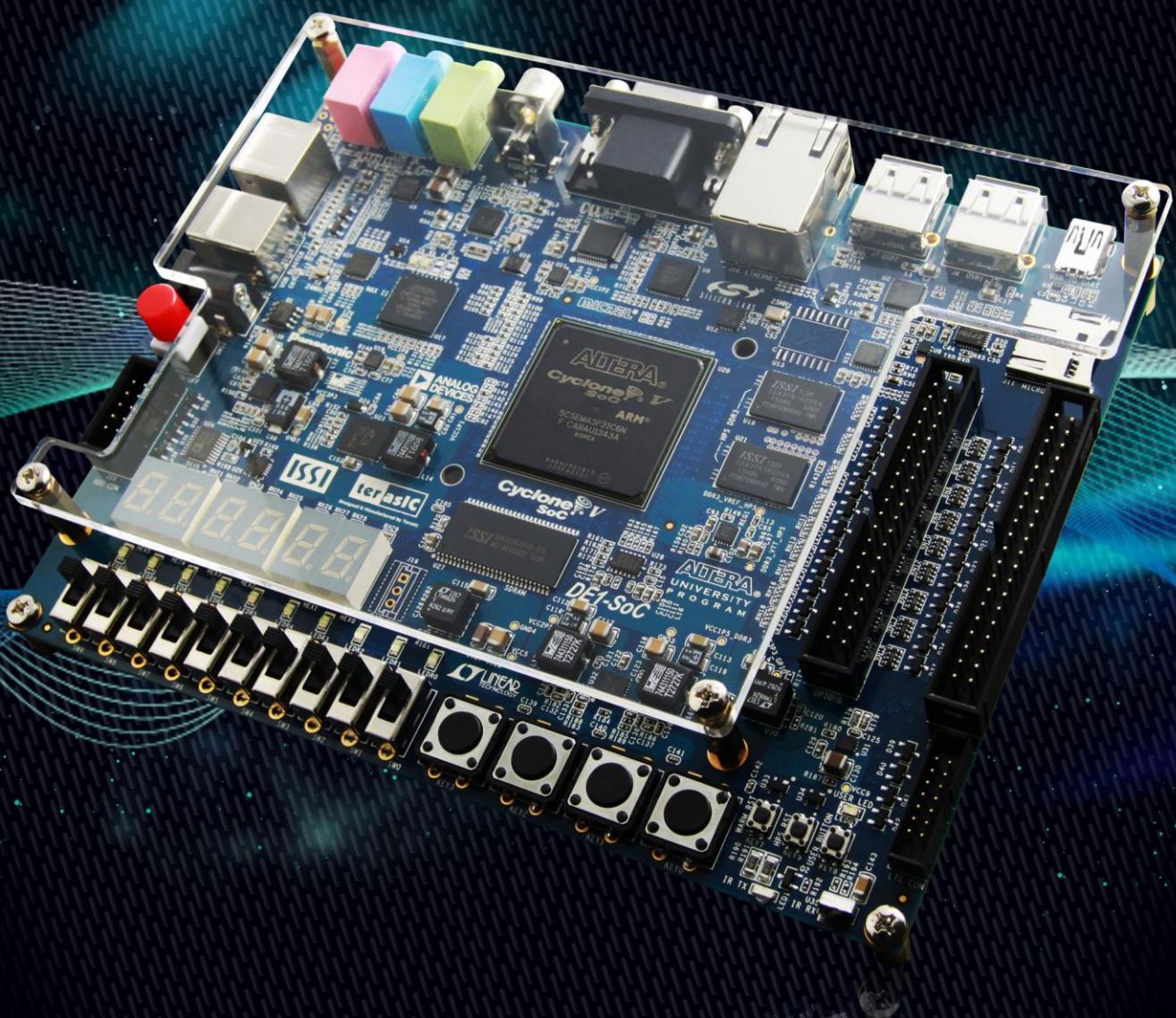


DE1-SoC



OpenCL

OpenCL



terasic
www.terasic.com

ALTERA
UNIVERSITY
PROGRAM

Copyright © 2003-2014 Terasic Technologies Inc. All Rights Reserved.

CONTENTS

CHAPTER 1 <i>INTRODUCTION</i>.....	3
1.1 DE1-SOC OPENCL BSP	3
1.2 SYSTEM REQUIREMENTS.....	4
1.3 OPENCL ARCHITECTURE	4
1.4 EXECUTE OPENCL DEMO ON DE1-SOC	5
CHAPTER 2 <i>OPENCL ON WINDOWS</i>.....	10
2.1 SOFTWARE INSTALLATION.....	10
2.2 LICENSE INSTALLATION	11
2.3 OPENCL ENVIRONMENT CONFIGURATION	12
2.4 OPENCL ENVIRONMENT VERIFICATION	14
2.5 COMPILE AND EXECUTE OPENCL PROJECT.....	15
CHAPTER 3 <i>OPENCL ON LINUX</i>.....	19
3.1 SOFTWARE INSTALLATION.....	19
3.2 OPENCL LICENSE INSTALLATION.....	20
3.3 CONFIGURATION OF ENVIRONMENT VARIABLES	20
3.4 VERIFICATION OF OPENCL ENVIRONMENT	21
3.5 BUILD AND EXECUTE OPENCL PROJECT.....	22

Chapter 1

Introduction

DE1-SoC 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 DE1-SoC. 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 DE1-SoC OpenCL BSP

The DE1-SoC OpenCL Board Support Package (BSP) contains required resources for users to develop OpenCL project based on DE1-SoC Board. The BSP is available from the website:

<http://de1-soc.terasic.com/cd>

For Windows Host, please download DE1-SoC_OpenCL_BSP.zip. For Linux, please download the DE1-SoC_OpenCL_BSP.tar.gz. The CentOS 7.7 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 DE1-SoC.

📁 arm32	2020/3/5 16:24
📁 hardware	2020/5/13 10:43
📁 tests	2020/5/13 10:42
📄 board_env.xml	2020/3/5 16:51
ZIP de1_soc_opencl.zip	2020/5/13 18:04

Figure 1-1 Contents of OpenCL BSP for DE1-SoC

1.2 System Requirements

The following items are required to setup OpenCL for DE1-SoC board:

- Terasic DE1-SoC 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.

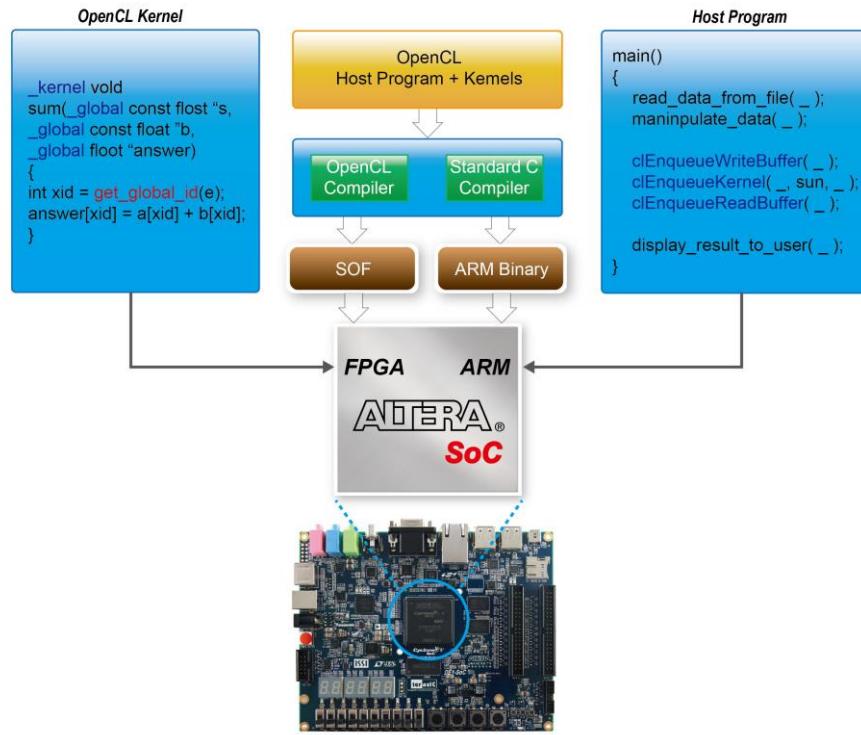


Figure 1-2 Intel SoC FPGA OpenCL architecture

1.4 Execute OpenCL Demo on DE1-SoC

This section describes how to execute OpenCL demo on DE1-SoC from the Linux image file included in DE1-SoC 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 DE1-SoC BSP from <http://de1-soc.terasic.com/cd> and extract the Linux image file **de1_soc_opencl.img** from **de1_soc_opencl.zip**
2. Write the Linux image file **de1_soc_opencl.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 DE1-SoC via an USB cable.
Users need to install the UART-to-USB device driver as described in the DE1-SoC Getting Started Guide.
6. Launch PuTTY utility on your host PC. Make sure the baud rate is set to 115200.
7. Power on DE1-SoC 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 hello_world/" to go to the hello_world folder.
 - Type "aocl program /dev/acl0 hello_world.aocx" to reconfigure the FPGA with the hello_world kernel.
 - Type "./host" to launch the hello_world host application, as shown in **Figure 1-3**.
10. Launch vector_add demo:
 - Type "cd .." to return to the home directory.
 - Type "cd vector_add" to go to the vector_add folder.
 - Type "aocl program /dev/acl0 vector_add.aocx" to reconfigure the FPGA with the vector_add kernel.
 - Type "./host" to launch the vector_add host application, as shown in **Figure 1-4**.

```
root@socfpga:~# source ./init_opencl.sh
root@socfpga:~# cd hello_world/
root@socfpga:~/hello_world# aocl program /dev/acl0 hello_world.aocx
aocl program: Running reprogram from /home/root/opencl_arm32_rte/board/c5soc/arm
32/bin
Reprogramming was successful!
root@socfpga:~/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 Open
CL(TM), Version 18.1

Querying device for info:
=====
CL_DEVICE_NAME              = de1soc_sharedonly : Cyclone V SoC Dev
elopment Kit
CL_DEVICE_VENDOR             = Intel(R) Corporation
CL_DEVICE_VENDOR_ID          = 4466
CL_DEVICE_VERSION            = OpenCL 1.0 Intel(R) FPGA SDK for Open
CL(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
```

Figure 1-3 Hello-world demo

```

kernel execution is complete.
root@socfpga:~/hello_world# cd ..
root@socfpga:~# cd vector_add
root@socfpga:~/vector_add# aocl program /dev/acl0 vector_add.aocx
aocl program: Running reprogram from /home/root/opencl_arm32_rte/board/c5soc/arm
32/bin
Reprogramming was successful!
root@socfpga:~/vector_add# ./host
Initializing OpenCL
Platform: Intel(R) FPGA SDK for OpenCL(TM)
Using 1 device(s)
    de1soc_sharedonly : Cyclone V SoC Development Kit
Using AOCX: vector_add.aocx
Reprogramming device [0] with handle 1
Launching for device 0 (1000000 elements)

Time: 114.366 ms
Kernel time (device 0): 7.191 ms

Verification: PASS
root@socfpga:~/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.

- Minicom – a terminal which can be installed via command "yum install minicom" or "apt-get install minicom"
1. Download DE1-SoC BSP from <http://de1-soc.terasic.com/cd> and extract the Linux image file **de1_soc_opencl.img** from **de1_soc_opencl.zip**.
 2. Write the Linux image file **de1_soc_opencl.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=de1_soc_opencl.img of=/dev/sdb bs=1M"
 - Run "sync"

3. Insert the microSD card into the microSD card slot (J11) of DE1-SoC.
4. The DIP switch (SW10) on DE1-SoC for MSEL[4:0] must be set to 01010.
5. Connect the host PC to the UART-to-USB port (J4) on DE1-SoC 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.htm>.
6. Launch Minicom utility ("minicom -s" for the configuration when it's launched the first time) on the host PC. The baud rate should be set to 115200. Shutdown the hardware flow control.
7. Power on DE1-SoC 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 hello_world/" to change the current directory to the hello_world folder.
 - Type "aoctl program /dev/acl0 hello_world.aocx" to reconfigure the FPGA with the hello_world kernel.
 - Type "./hello_world" to launch the hello_world host application, as shown in **Figure 1-3**.
10. Launch vector Add demo:
 - Type "cd .." to return to the home directory.
 - Type "cd vector_add" to change the directory to the vector_add folder.
 - Type "aoctl program /dev/acl0 vector_add.aocx" to reconfigure the FPGA with the vector_add kernel.
 - Type "./vector_add" to launch the vector_add host application, as shown in **Figure 1-4**.

Chapter 2

OpenCL on Windows

This chapter describes how to setup DE1-SoC OpenCL development environment in Windows 64-bit and then build and execute OpenCL project on DE1-SoC. 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 DE1-SoC.

■ Install Intel Quartus Prime and OpenCL SDK

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

<https://fpgasoftware.intel.com/18.1/?edition=standard&platform=windows>

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:

<https://fpgasoftware.intel.com/soceds/18.1/?edition=standard&platform=windows>

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

■ Install DE1-SoC 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\hdl\board", assuming Intel Quartus Prime is installed under the folder "D:\intelFPGA\18.1". Please download the DE1-SoC BSP file **DE1-SoC_OpenCL_BSP.zip** from

<http://de1-soc.terasic.com/cd>

Please decompress the zip file and copy the "de1_soc" folder to the "terasic" folder created previously, as shown in **Figure 2-1**.

Data (D:) > intelFPGA > 18.1 > hld > board		
Name	Date modified	Type
a10_ref	3/5/2020 4:23 PM	File folder
a10soc	3/5/2020 4:23 PM	File folder
c5soc	3/5/2020 5:35 PM	File folder
custom_platform_toolkit	3/5/2020 4:23 PM	File folder
s5_ref	3/5/2020 4:23 PM	File folder
terasic	5/13/2020 1:23 PM	File folder

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

2.2 License Installation

A 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".

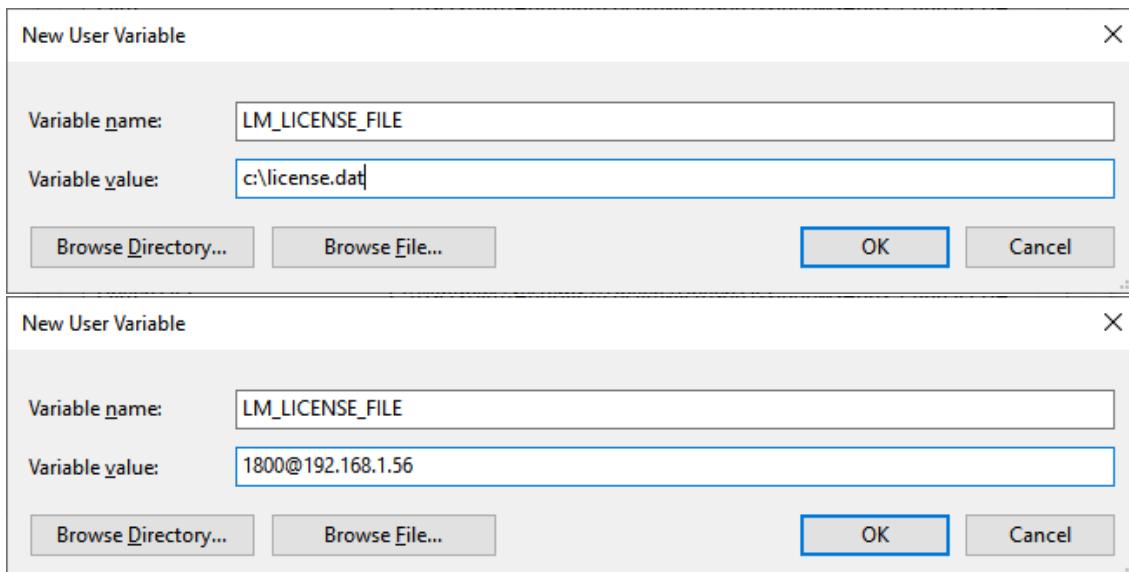


Figure 2-2 Setup LM_LICENSE_FILE environment variable

2.3 OpenCL Environment Configuration

Please add the following paths into the PATH environment variable:

■ OpenCL Configuration

For the operating system to find the OpenCL utilities correctly, users need to 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)

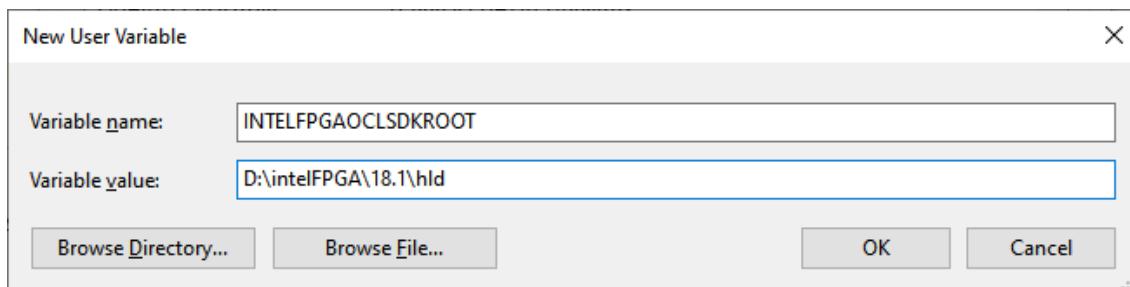


Figure 2-3 Add INTELFPGAOCLSDKROOT Environment Variable

7. Select **PATH** item 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.
 - %INTELFPGAOCLSDKROOT%\bin
 - %INTELFPGAOCLSDKROOT%\host\windows64\bin

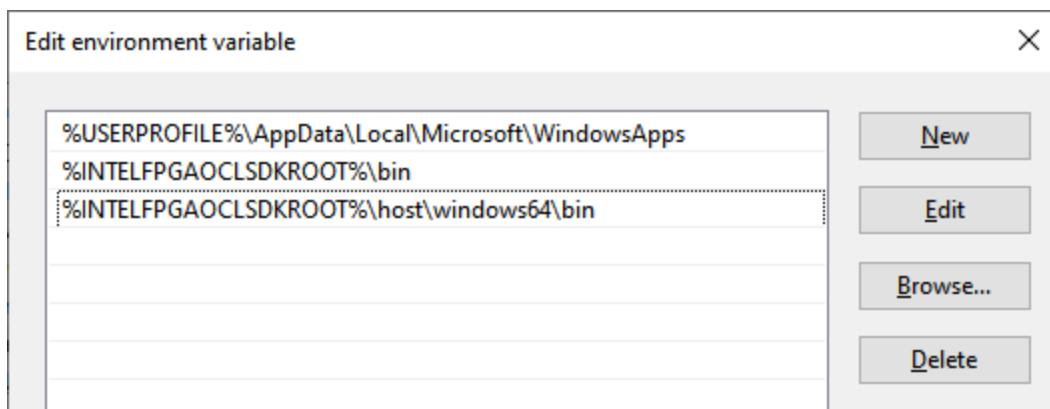


Figure 2-4 Modify PATH Environment Variable

■ DE1-SoC BSP Configuration

For Intel OpenCL SDK to find the kit location of DE1-SoC correctly, Users need to create an environment variable **AOCL_BOARD_PACKAGE_ROOT** and set

its value as:

```
"%INTELFPGAOCLSDKROOT%\board\terasic\de1_soc"
```

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\de1_soc"** in the **Variable value** field.

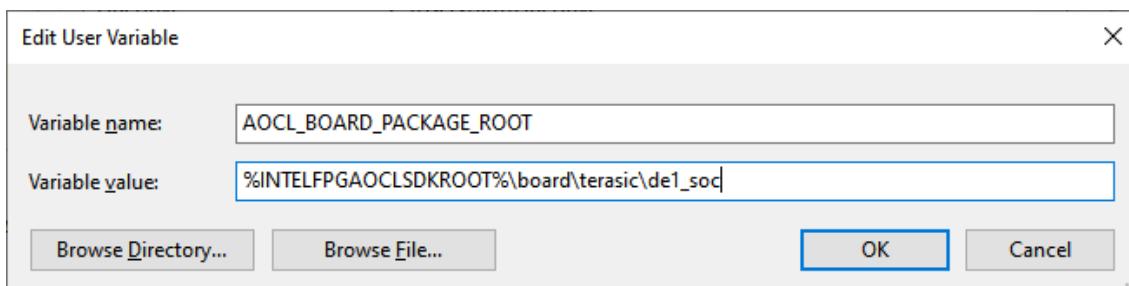


Figure 2-5 Setup AOCL_BOARD_PACKAGE_ROOT environment variable

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\User>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 Corporation)
C:\Users\User>
```

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 "de1_soc_sharedonly" is listed in the **Board list**, as shown in [Figure 2-7](#). If "de1_soc_sharedonly" is not listed, please check if the **AOCL_BOARD_PACKAGE_ROOT** environment variable is assigned correctly.

```
C:\Users\User>aoc -list-boards
Board list:
de1soc_sharedonly
    Board Package: D:\intelFPGA\18.1\hld\board\terasic\de1_soc
C:\Users\User>
```

Figure 2-7 "de1_soc_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**.

```
C:\Users\User>echo %AOCL_BOARD_PACKAGE_ROOT%
D:\intelFPGA\18.1\hld\board\terasic\de1_soc
C:\Users\User>
```

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 DE1-SoC BSP. Users can follow the same procedures to compile and execute other

OpenCL examples for DE1-SoC.

■ 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\de1_soc\tests\boardtest" in the command prompt window to change the current directory to the folder **boardtest** and then type:

```
"aoc device/boardtest.cl -o bin/boardtest.aocx -report -no-interleaving=default"
```

to compile the OpenCL kernel. It will take approximately 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 compiled successfully. 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

```
命令提示字元
ormance
__global uint *dst,
^
d:/intelFPGA/18.1/hld/board/terasic/de1_soc/tests/boardtest/device/boardtest.cl:79:2
6: warning: declaring kernel argument with no 'restrict' may lead to low kernel perf
ormance
__global const uint *index,
^
2 warnings generated.
aoc: Optimizing and doing static analysis of code...

! -----
! 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%     ;
+-----+
Compiling for FPGA. This process may take a long time, please be patient.

D:\intelFPGA\18.1\hld\board\terasic\de1_soc\tests\boardtest>
```

Figure 2-9 Screenshot of OpenCL kernel compiled successfully by typing "aoc boardtest.cl"

■ 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"

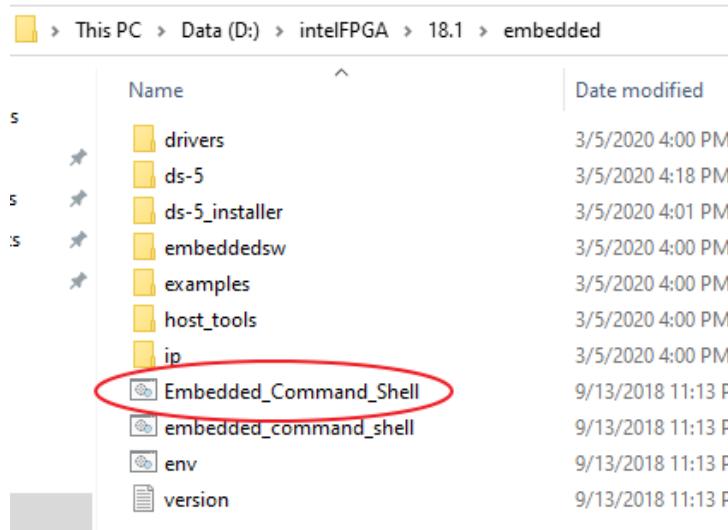


Figure 2-10 Location of embedded_command_Shell.bat

Type the following in the command shell:

```
"cd /cygdrive/D/intelFPGA/18.1/hld/board/terasic/de1_soc/tests/boardtest/"
```

to change the current directory to the folder where the board test 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.

```
Jim@DESKTOP-RMUP9R6 ~
$ cd /cygdrive/D/intelFPGA/18.1/hld/board/terasic/de1_soc/tests/boardtest/

Jim@DESKTOP-RMUP9R6 /cygdrive/D/intelFPGA/18.1/hld/board/terasic/de1_soc/tests/boardtest
$ make
arm-linux-gnueabihf-g++ -fPIC -DARM host/aclutil.cpp host/hostspeed.cpp host/
hostspeed_ocl.cpp host/kernel_launch.cpp host/kernel_rw.cpp host/main.
cpp host/memspeed.cpp host/reorder.cpp host/reorder_ocl.cpp host/rwtest.
cpp host/timer.cpp -o boardtest_host -DLINUX -ID:/intelFPGA/18.1/hld/host/
include D:\intelFPGA\18.1\hld/host/arm32/lib/libacl_emulator_kernel1_rt.so D
:\intelFPGA\18.1\hld/host/arm32/lib/libalteracl.so D:\intelFPGA\18.1\hld/hos
t/arm32/lib/libalterahalmmd.so D:\intelFPGA\18.1\hld/host/arm32/lib/libc_acc
el_runtime.so D:\intelFPGA\18.1\hld/host/arm32/lib/libelf.so D:\intelFPGA\18
.1\hld\board\terasic\de1_soc/arm32/lib/libintel_soc32_mmd.so -lrt
```

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

■ Execute Board Test Project

Please boot DE1-SoC with the Linux image generated from Chapter 1.4 for DE1-SoC 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 DE1-SoC. 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 DE1-SoC, please go to the terminal and type "source ./init_opencl.sh" to setup OpenCL environment. Type "chmod +x boardtest_host" to add execution attribute to the host file and type "aocl program /dev/acl0 boardtest.aocx" to configure the FPGA, as shown in **Figure 2-12**. Type "./boardtest_host" to launch the host application, as shown in **Figure 2-13**.

```
Poky 8.0 (Yocto Project 1.3 Reference Distro) 1.3 socfpga ttyS0

socfpga login: root
root@socfpga:~# source ./init_opencl.sh
root@socfpga:~# chmod +x boardtest_host
root@socfpga:~# aocl program /dev/acl0 boardtest.aocx
aocl program: Running reprogram from /home/root/opencl_arm32_rte/board/c5soc/arm
32/bin
Reprogramming was successful!
root@socfpga:~#
```

Figure 2-12 "aocl program /dev/acl0 boardtest.aocx" configures FPGA successfully

```
BOARD BANDWIDTH UTILIZATION = 38.17%
Warning : Board bandwidth utilization is less than 90%

Kernel mem bandwidth assuming ideal memory: 17572 MB/s
    * If this is lower than your board's peak memory
    * bandwidth then your kernel's clock isn't fast enough
    * to saturate memory
    * approx. fmax = 137

Kernel mem bandwidth assuming ideal memory is greater than board's peak memory b
andwidth. Success.

KERNEL-TO-MEMORY BANDWIDTH = 2443 MB/s/bank

*****
***** TEST COMPLETED FOR DEVICE 0 *****
*****
```

```
BOARDTEST PASSED
root@socfpga:~#
```

Figure 2-13 "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 Guide 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 DE1-SoC.

■ Install Intel Quartus Prime and OpenCL SDK

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

<https://fpgasoftware.intel.com/18.1/?edition=standard&platform=linux>

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:

<https://fpgasoftware.intel.com/soceds/18.1/?edition=standard&platform=linux>

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

■ Install DE1-SoC 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/user/intelFPGA/18.1/hld/board/", refer to your own path, and uncompressing the OpenCL BSP to this folder, as shown in [Figure 3-1](#).

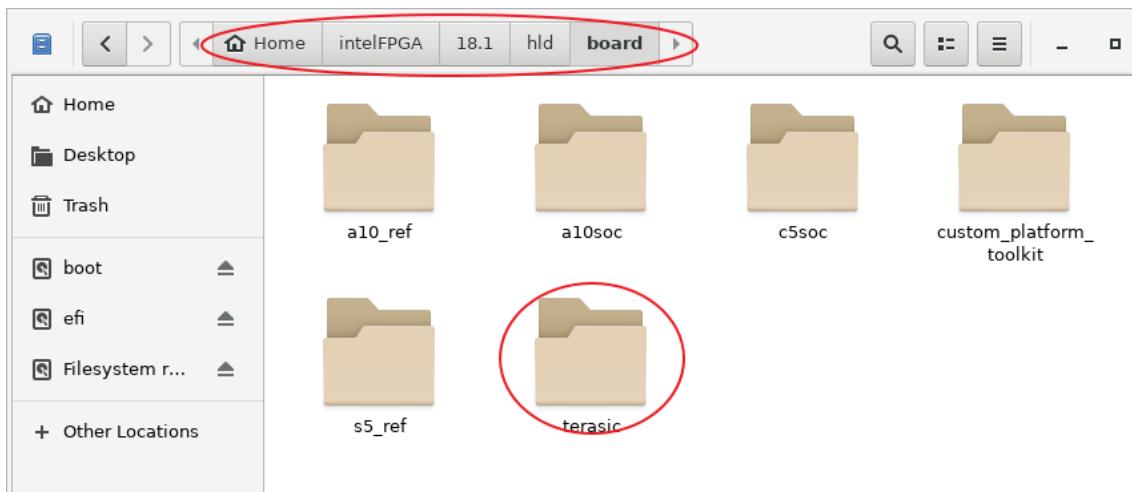


Figure 3-1 Copy the terasic folder to the intelFPGA/18.1/hld/board folder

3.2 OpenCL License Installation

A license for OpenCL is required to compile OpenCL project with Intel OpenCL SDK. Users can purchase the OpenCL 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/user/intelFPGA/18.1/hld/license.dat". Users also need to create an environment variable **LM_LICENSE_FILE** and set its value as "/home/user/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 DE1-SoC correctly. The value of this environment variable needs to be set as:

```
"$INTELFPGAOCLSDKROOT/board/terasic/de1_soc"
```

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. After the edit is complete, save the file and type "*source /etc/profile*" command in the terminal to apply the settings.

```
export QUARTUS_ROOTDIR=/home/user/intelFPGA/18.1/quartus
export INTELFPGAOCLSDKROOT=/home/user/intelFPGA/18.1/hld
export PATH=$PATH:$INTELFPGAOCLSDKROOT/bin:$INTELFPGAOCLSDKROOT/linux64/bin:
export LD_LIBRARY_PATH=$INTELFPGAOCLSDKROOT/linux64/lib
export AOCL_BOARD_PACKAGE_ROOT=$INTELFPGAOCLSDKROOT/board/terasic/de1_soc
export LM_LICENSE_FILE=/home/user/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 3-2**. If the "aocl" command cannot be found, please check if the "\$INTELFPGAOCLSDKROOT/bin" path is added to the **PATH** environment variable correctly.

```
[user@localhost ~]$ 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 Corporation)
[user@localhost ~]$
```

Figure 3-2 The information about aocl version

■ Verify Target Board

Type "aoc -list-boards" command in the terminal and make sure "de1_soc_sharedonly" is displayed

in the Board list, as shown in **Figure 3-3**. If "de1_soc_sharedonly" is not listed, please check if the **AOCL_BOARD_PACKAGE_ROOT** environment variable is assigned correctly.

```
[user@localhost ~]$ aoc -list-boards
Board list:
  delsoc_sharedonly
    Board Package: /home/user/intelFPGA/18.1/hld/board/terasic/de1_soc

```

Figure 3-3 "de1_soc_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 3-4**.

```
[user@localhost ~]$ echo $AOCL_BOARD_PACKAGE_ROOT
/home/user/intelFPGA/18.1/hld/board/terasic/de1_soc

```

Figure 3-4 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 DE1-SoC BSP. Users can follow the same procedures to compile and test other OpenCL projects for DE1-SoC.

■ Compile OpenCL Kernel

The utility **aoc** (Intel SDK for OpenCL Kernel Compiler) is used to compile OpenCL kernel. Type "cd /home/user/intelFPGA/18.1/hld/board/terasic/de1_soc/tests/boardtest" in the terminal to change the current directory to the **boardtest** project folder. Type

```
"aoc device/boardtest.cl -o bin/boardtest.aocx -report -no-interleaving=default"
```

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 3-5** 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

```
[user@localhost boardtest]$ aoc device/boardtest.cl -o bin/boardtest.aocx -report-no-interleaving=default
aoc: Selected default target board delsoc_sharedonly
aoc: Running OpenCL parser....
/home/user/intelFPGA/18.1/hld/board/terasic/delsoc/tests/boardtest/device/board
test.cl:78:20: warning: declaring kernel argument with no 'restrict' may lead to
    low kernel performance
    __global uint *dst,
        ^
/home/user/intelFPGA/18.1/hld/board/terasic/delsoc/tests/boardtest/device/board
test.cl:79:26: warning: declaring kernel argument with no 'restrict' may lead to
    low kernel performance
    __global const uint *index,
        ^
2 warnings generated.
aoc: Optimizing and doing static analysis of code...

=====
! 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%    ;
+-----+-----+-----+-----+-----+-----+-----+
Compiling for FPGA. This process may take a long time, please be patient.
[user@localhost boardtest]$
```

Figure 3-5 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 3-6**, under the folder "/home/user/intelFPGA/18.1/embedded", assuming Intel SoC EDS is installed under the directory "/home/user/intelFPGA/18.1/".

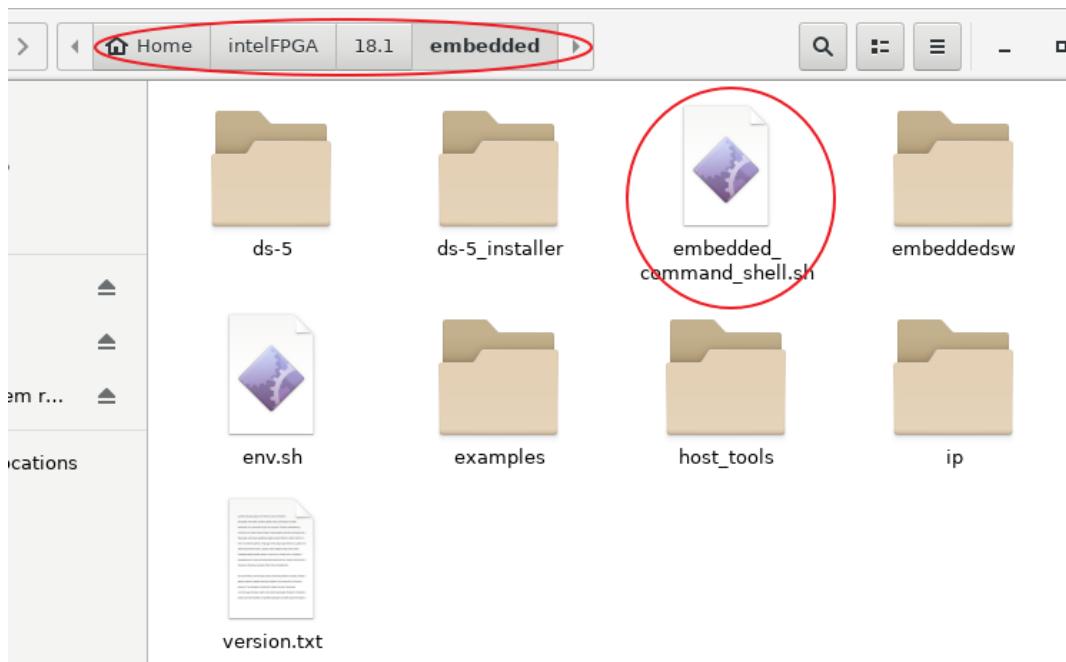


Figure 3-6 Location of Embedded_Command_Shell.sh

In the command shell, please type:

```
"cd $AOCL_BOARD_PACKAGE_ROOT/tests/boardtest/"
```

and the current directory will be changed to the folder where the board test project is located. Type "make" to build the host project, as shown in **Figure 3-7**. If the host project is compiled successfully, a host binary file **boardtes_host** will be generated under the boardtest folder.

```
[user@localhost boardtest]$ make
arm-linux-gnueabihf-g++ -fPIC -DARM host/rwtest.cpp host/timer.cpp host/reorder.cpp host/hostspeed.cpp host/reorder_ocl.cpp host/hostspeed_ocl.cpp host/kernel_launch.cpp host/kernel_rw.cpp host/aclutil.cpp host/memspeed.cpp host/main.cpp -o boardtest_host -DLINUX -I/home/user/intelFPGA/18.1/hld/host/include /home/user/intelFPGA/18.1/hld/host/arm32/lib/libacl_emulator_kernel_rt.so /home /user/intelFPGA/18.1/hld/host/arm32/lib/libelf.so /home/user/intelFPGA/18.1/hld/ host/arm32/lib/libOpenCL.so /home/user/intelFPGA/18.1/hld/host/arm32/lib/libaltera_hahalmm.so /home/user/intelFPGA/18.1/hld/host/arm32/lib/libc_accel_runtime.so /home/user/intelFPGA/18.1/hld/host/arm32/lib/libalteracl.so /home/user/intelFPGA/18.1/hld/ board/terasic/de1_soc/arm32/lib/libintel_soc32_mmd.so -lrt
[user@localhost boardtest]$
```

Figure 3-7 Type "make" to build the boardtest_host project

■ Test boardtest project

Please boot DE1-SoC with the Linux image for DE1-SoC 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 DE1-SoC. 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 DE1-SoC, please go to the terminal and type "source ./init_opencl.sh" to setup OpenCL environment. Type "chmod +x boardtest_host" to add execution attribute to the host file and type "aocl program /dev/acl0 boardtest.aocx" to configure the FPGA, as shown in **Figure 3-8**. Type "./boardtest_host" to launch the host application, as shown in **Figure 3-9**.

```
root@socfpga:~# source ./init_opencl.sh
root@socfpga:~# chmod +x boardtest_host
root@socfpga:~# aocl program /dev/acl0 boardtest.aocx
aocl program: Running reprogram from /home/root/opencl_arm32_rte/board/c5soc/arm
32/bin
Reprogramming was successful!
root@socfpga:~#
```

Figure 3-8 "aocl program /dev/acl0 boardtest.aocx" configures the FPGA successfully

It is assumed that all memory interfaces have equal widths.

```
BOARD BANDWIDTH UTILIZATION = 37.26%
Warning : Board bandwidth utilization is less than 90%
```

```
Kernel mem bandwidth assuming ideal memory: 17572 MB/s
    * If this is lower than your board's peak memory
    * bandwidth then your kernel's clock isn't fast enough
    * to saturate memory
    * approx. fmax = 137
```

```
Kernel mem bandwidth assuming ideal memory is greater than board's peak memory b
andwidth. Success.
```

```
KERNEL-T0-MEMORY BANDWIDTH = 2385 MB/s/bank
```

```
*****
***** TEST COMPLETED FOR DEVICE 0 *****
*****
```

```
BOARDTEST PASSED
root@socfpga:~#
```

Figure 3-9 "boardtest_host" test is successful