



Intel® FPGA SDK for OpenCL™ Standard Edition

Cyclone® V SoC Getting Started Guide

Updated for Intel® Quartus® Prime Design Suite: **18.1**



Online Version

Send Feedback

UG-OCL006

ID: **683030**

Version: **2018.09.24**

Contents

1. Intel® FPGA SDK for OpenCL™ Standard Edition Cyclone V SoC Getting Started Guide....	4
1.1. Prerequisites for the Intel FPGA SDK for OpenCL Standard Edition.....	5
1.2. Contents of the Intel FPGA SDK for OpenCL Standard Edition.....	5
1.3. Overview of the Intel FPGA SDK for OpenCL Standard Edition and Cyclone V SoC Development Kit Setup Processes.....	7
1.4. Overview of the Intel FPGA SDK for OpenCL Cyclone V SoC Programming Flow.....	8
1.5. Cyclone V SoC Development Kit Reference Platform Board Variants.....	9
1.6. Cyclone V SoC FPGA-Specific OpenCL Design Considerations.....	10
2. Setting Up the Intel FPGA SDK for OpenCL, Intel SoC FPGA Embedded Design Suite, and the Cyclone V SoC Development Kit for Windows.....	11
2.1. Upgrading to Current Version of Intel FPGA SDK for OpenCL for Cyclone V SoC FPGA....	11
2.2. Downloading the Intel FPGA SDK for OpenCL Standard Edition	12
2.3. Downloading the Intel SoC FPGA Embedded Development Suite.....	12
2.4. Installing the Intel FPGA SDK for OpenCL	13
2.5. Setting the Intel FPGA SDK for OpenCL User Environment Variables for SoC FPGA.....	13
2.6. Installing the Intel SoC FPGA Embedded Development Suite Standard Edition.....	14
2.7. Installing the Cyclone V SoC Development Kit.....	15
2.7.1. Writing an SD Card Image onto the Micro SD Flash Card	15
2.7.2. Configuring the SW3 Switches.....	16
2.7.3. Setting Up Terminal Connection	17
2.7.4. Setting Environment Variables and Loading OpenCL Linux Kernel Driver.....	17
2.7.5. Connecting the Board to Network via Ethernet.....	17
2.8. Downloading an OpenCL Design Example.....	19
2.9. Creating the Hardware Configuration File of an OpenCL Kernel for SoC FPGA.....	19
2.10. Executing an OpenCL Kernel on an SoC FPGA.....	21
2.10.1. Building the Host Application.....	21
2.10.2. Running the Host Application.....	22
2.10.3. Output from Successful Kernel Execution on the Cyclone V SoC Development Kit.....	22
2.11. Uninstalling the Software.....	23
3. Setting Up the Intel FPGA SDK for OpenCL, Intel SoC FPGA Embedded Design Suite, and the Cyclone V SoC Development Kit for Linux.....	25
3.1. Upgrading to Current Version of Intel FPGA SDK for OpenCL for SoC FPGA.....	25
3.2. Downloading the Intel FPGA SDK for OpenCL Standard Edition	26
3.3. Downloading the Intel SoC FPGA Embedded Development Suite.....	26
3.4. Installing the Intel FPGA SDK for OpenCL.....	27
3.5. Setting the Intel FPGA SDK for OpenCL User Environment Variables for SoC FPGA.....	28
3.6. Installing the Intel SoC FPGA Embedded Development Suite Pro Edition.....	28
3.7. Installing the Cyclone V SoC Development Kit.....	29
3.7.1. Writing an SD Card Image onto the Micro SD Flash Card.....	29
3.7.2. Configuring the SW3 Switches.....	30
3.7.3. Setting Up Terminal Connection.....	31
3.7.4. Setting Environment Variables and Loading OpenCL Linux Kernel Driver.....	32
3.7.5. Connecting the Board to Network via Ethernet.....	32
3.8. Verifying Host Runtime Functionality via Emulation.....	34
3.8.1. Downloading an OpenCL Design Example.....	35

3.8.2. Compiling a Kernel for Emulation (-march=emulator).....	35
3.8.3. Building the Host Application.....	36
3.8.4. Emulating Your OpenCL Kernel.....	37
3.9. Creating the Hardware Configuration File of an OpenCL Kernel for SoC FPGA.....	38
3.10. Executing an OpenCL Kernel on an SoC FPGA.....	39
3.10.1. Running the Host Application.....	39
3.10.2. Output from Successful Kernel Execution on the Cyclone V SoC Development Kit.....	40
3.11. Uninstalling the Software.....	41
A. Document Revision History of the Intel FPGA SDK for OpenCL Standard Edition Cyclone V SoC Getting Started Guide.....	42

1. Intel® FPGA SDK for OpenCL™ Standard Edition Cyclone V SoC Getting Started Guide

The *Intel® FPGA SDK for OpenCL™ Standard Edition Cyclone V SoC Getting Started Guide* describes the procedures to set up and use the Intel FPGA Software Development Kit (SDK) for OpenCL ⁽¹⁾ ⁽²⁾ Standard Edition to run an OpenCL application on the Cyclone V SoC Development Kit.

OpenCL is a C-based open standard for the programming of heterogeneous parallel devices. The SDK provides a compiler and tools for you to build and run OpenCL applications that target Intel FPGA products. The SDK supports the embedded profile of the OpenCL Specification version 1.0. For more information on the OpenCL Specification version 1.0, refer to the OpenCL Reference Pages. The *OpenCL Specification version 1.0* includes detailed information on the OpenCL application programming interface (API) and programming language.

This document assumes the following:

1. You are knowledgeable in OpenCL concepts and APIs, as described in the *OpenCL Specification version 1.0* by the Khronos Group.
2. You have experience in creating OpenCL applications, and are familiar with the contents of the OpenCL Specification.
3. You are familiar with the information available in the following Intel FPGA SDK for OpenCL Standard Edition documentation:
 - *Intel FPGA SDK for OpenCL Standard Edition Getting Started Guide.*
 - *Intel FPGA SDK for OpenCL Standard Edition Programming Guide.*
 - *Intel FPGA SDK for OpenCL Standard Edition Best Practices Guide.*

Related Information

- [OpenCL Reference Pages](#)
- [OpenCL Specification version 1.0](#)
- [Intel FPGA SDK for OpenCL Standard Edition Getting Started Guide](#)
- [Intel FPGA SDK for OpenCL Standard Edition Programming Guide](#)
- [Intel FPGA SDK for OpenCL Standard Edition Best Practices Guide](#)

⁽¹⁾ OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission of the Khronos Group™.

⁽²⁾ The Intel FPGA SDK for OpenCL is based on a published Khronos Specification, and has passed the Khronos Conformance Testing Process. Current conformance status can be found at www.khronos.org/conformance.

1.1. Prerequisites for the Intel FPGA SDK for OpenCL Standard Edition

To install the Intel FPGA SDK for OpenCL Standard Edition and create an OpenCL application for the Cyclone® V SoC Development Kit, your system must meet certain hardware, target platform, and software requirements.

Hardware Requirements

Ensure that your system meets the following minimum hardware and operating system requirements:

- The accelerator board is the Cyclone V SoC Development Kit.
- Development system requirements:
 - You must have administrator privileges on the development system to install the necessary packages and drivers.
 - The development system has at least 85 gigabytes (GB) of free disk space for software installation.
 - The development system has at least 8 GB of RAM.
 - The development system must be running on one of the following supported operating systems:
 - For a list of supported Windows and Linux operating systems, refer to the [Operating System Support](#) page.

You will use the development system to perform the following tasks:

- Compile OpenCL kernel source files to create hardware configuration files.
- Cross-compile the host executables to the ARM* processor.
- Connect your system to the SoC FPGA board so that they can communicate with each other using UART over USB connection.

Software Prerequisites

- Linux systems require the Perl command version 5 or later. Ensure that your *PATH* environment variable setting includes the path to the Perl command.

1.2. Contents of the Intel FPGA SDK for OpenCL Standard Edition

The Intel FPGA SDK for OpenCL Standard Edition provides programs, drivers, and SDK-specific libraries and files.

Logic Components

- The *Intel FPGA SDK for OpenCL Offline Compiler* translates your OpenCL device code into a hardware configuration file that the system loads onto an Intel FPGA product.
- The *Intel FPGA SDK for OpenCL Standard Edition* utility includes a set of commands you can invoke to perform high-level tasks such as running diagnostic tests.
- The *host runtime* provides the OpenCL host and runtime API for your OpenCL host application.

The host runtime consists of libraries that provide OpenCL APIs, hardware abstractions, and helper libraries.

Drivers, Libraries and Files

The software installation process installs the Intel FPGA SDK for OpenCL Standard Edition into a directory that you own. The `INTELFPGAOCSDKROOT` environment variable references the path to the SDK's installation directory.

Table 1. Select Contents of the Intel FPGA SDK for OpenCL Standard Edition Installation Directory

Windows Folder	Linux Directory	Description
bin	bin	User commands in the SDK. Include this directory in your <code>PATH</code> environment variable setting.
board	board	The Intel FPGA SDK for OpenCL Standard Edition Custom Platform Toolkit and Reference Platforms available with the software. <ul style="list-style-type: none"> • The path to the Custom Platform Toolkit is <code>INTELFPGAOCSDKROOT/board/custom_platform_toolkit</code> • The path to the s5_ref Reference Platform is <code>INTELFPGAOCSDKROOT/board/s5_ref</code> • The path to the Cyclone V SoC Development Kit Reference Platform, is <code>INTELFPGAOCSDKROOT/board/c5soc</code>
ip	ip	Intellectual property (IP) cores used to compile device kernels.
host	host	Files necessary for compiling and running your host application.
host\include	host/include	OpenCL Specification version 1.0 header files and software interface files necessary for compiling and linking your host application. The <code>host/include/CL</code> subdirectory also includes the C++ header file <code>cl.hpp</code> . The file contains an OpenCL version 1.1 C++ wrapper API. These C++ bindings enable a C++ host program to access the OpenCL runtime APIs using native C++ classes and methods. <i>Important:</i> The OpenCL version 1.1 C++ bindings are compatible with OpenCL Specification versions 1.0 and 1.1. Add this path to the <code>include</code> file search path in your development environment.
host \windows64\lib	host/ linux64/lib	OpenCL host runtime libraries that provide the OpenCL platform and runtime APIs. These libraries are necessary for linking your host application. To run an OpenCL application on Linux, include this directory in the <code>LD_LIBRARY_PATH</code> environment variable setting.
host \windows64\bin	host/ linux64/bin	Runtime commands and libraries necessary for running your host application, wherever applicable. For 64-bit Windows system, include this directory in your <code>PATH</code> environment variable setting. For Windows system, this folder contains runtime libraries.

continued...

Windows Folder	Linux Directory	Description
		For Linux system, this directory contains platform-specific binary for the aoc1 utility command.
share	share	Architecture-independent support files.

Example OpenCL Applications

You can download example OpenCL applications from the OpenCL Design Examples page.

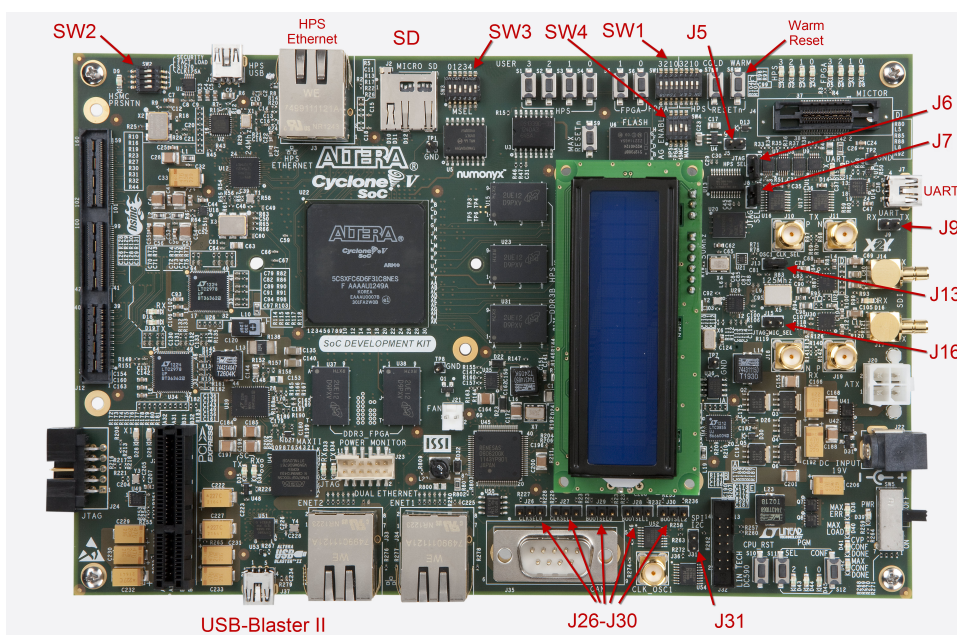
Related Information

[OpenCL Design Examples](#)

1.3. Overview of the Intel FPGA SDK for OpenCL Standard Edition and Cyclone V SoC Development Kit Setup Processes

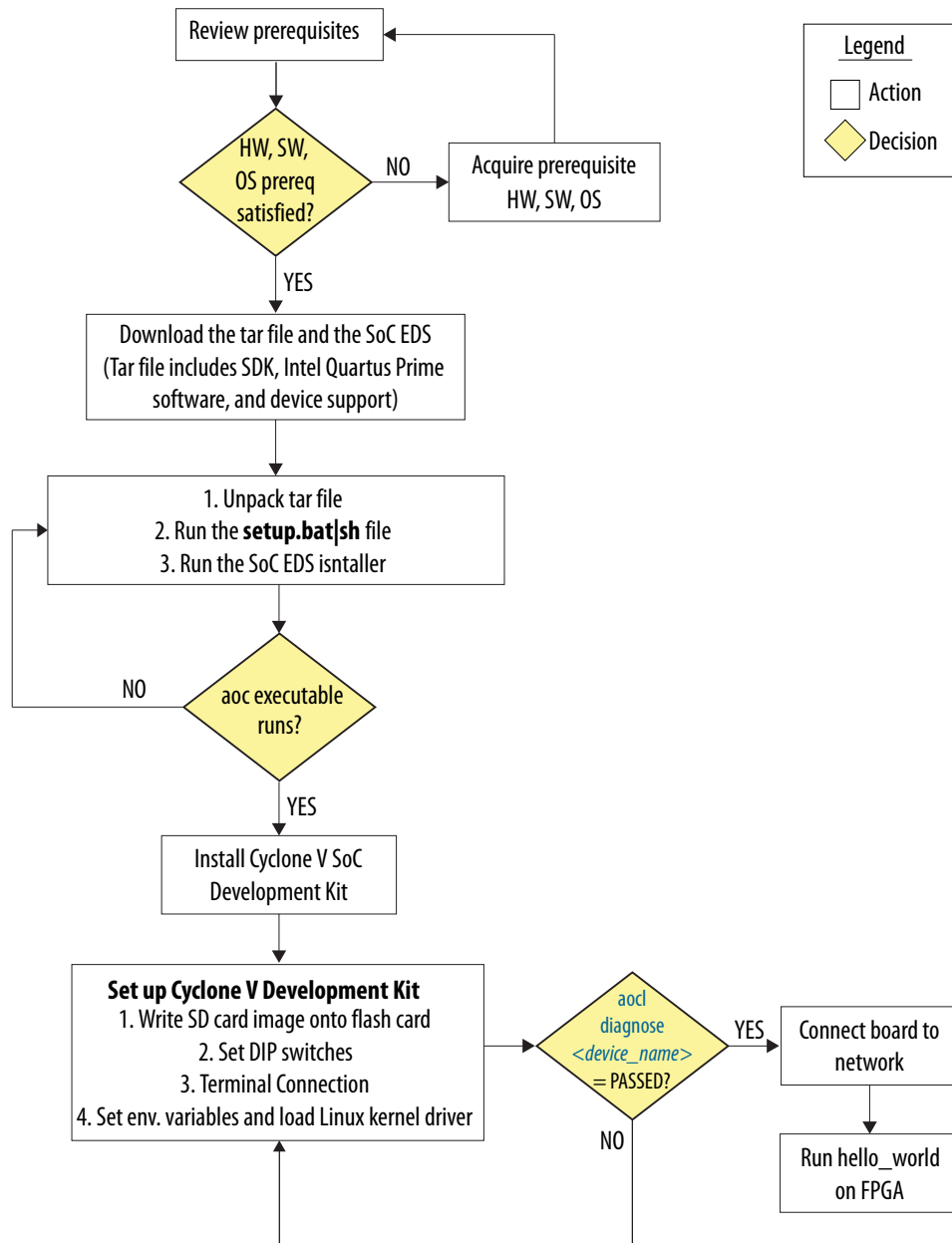
The *Intel FPGA SDK for OpenCL Standard Edition Cyclone V SoC Getting Started Guide* outlines the procedures for installing the Intel FPGA SDK for OpenCL Standard Edition and all related software. It also outlines the setup process for the Cyclone V SoC Development Kit.

Figure 1. Key Components of the Cyclone V SoC Development Kit



The figure below summarizes the steps for setting up the necessary software and installing the Cyclone V SoC Development Kit.

Figure 2. Overview of the SDK and Cyclone V SoC Development Kit Setup

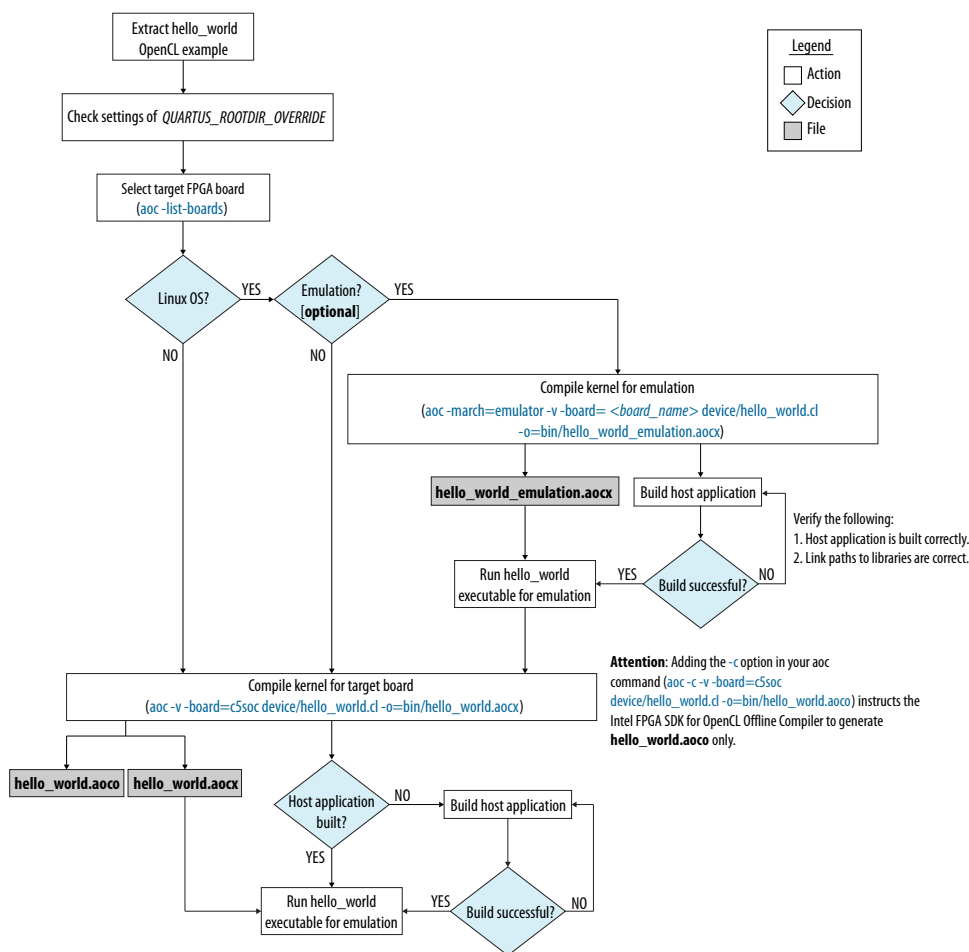


1.4. Overview of the Intel FPGA SDK for OpenCL Cyclone V SoC Programming Flow

The *Intel FPGA SDK for OpenCL Standard Edition Cyclone V SoC Getting Started Guide* outlines the procedures for programming the `hello_world` OpenCL design example onto the Cyclone V SoC Development Kit.

The figure below summarizes the steps you perform to program your Cyclone V SoC. Before you program the SoC, verify that the Cyclone V SoC Development kit micro SD flash card contains an image created using the current version of the SDK. Refer to *Upgrading to Current Version of Intel FPGA SDK for OpenCL Standard Edition for Cyclone V SoC* for more information.

Figure 3. Cyclone V SoC Development Kit Programming Overview



Related Information

- Upgrading to Current Version of Intel FPGA SDK for OpenCL for Cyclone V SoC FPGA on page 11
- Upgrading to Current Version of Intel FPGA SDK for OpenCL for Cyclone V SoC FPGA on page 25

1.5. Cyclone V SoC Development Kit Reference Platform Board Variants

The Intel FPGA SDK for OpenCL Cyclone V SoC Development Kit Reference Platform includes two board variants.

- c5soc board
This default board provides access to two DDR memory banks. The HPS DDR is accessible by both the FPGA and the CPU. The FPGA DDR is only accessible by the FPGA.
- c5soc_sharedonly board
This board variant contains only HPS DDR connectivity. The FPGA DDR is not accessible. This board variant is more area efficient because less hardware is necessary to support one DDR memory bank. The c5soc_sharedonly board is also a good prototyping platform for a final production board with a single DDR memory bank.

To target this board variant when compiling your OpenCL kernel, include the `-board=c5soc_sharedonly` option in your `aoc` command.

For more information about the `-board=<board_name>` option of the `aoc` command, refer to the *Intel FPGA SDK for OpenCL Standard Edition Programming Guide*.

Related Information

[Compiling a Kernel for a Specific FPGA Board \(-board=<board_name>\)](#)

1.6. Cyclone V SoC FPGA-Specific OpenCL Design Considerations

When designing your OpenCL kernel and host program for execution on the Cyclone V SoC Development Kit, consider implementing design recommendations such as shared memory usage and FPGA area optimization.

Shared memory is the preferred memory for FPGA kernels. For information on how to allocate and use shared memory, refer to the *Intel FPGA SDK for OpenCL Standard Edition Programming Guide*.

The Cyclone V SoC FPGA on the Cyclone V SoC Development Kit is not considered a large FPGA. However, if you structure your kernel code in a way that optimizes hardware usage, it can provide sufficient hardware resources to implement complex computations. Intel recommends that you consult the *Intel FPGA SDK for OpenCL Standard Edition Best Practices Guide* to obtain a good understanding of the Intel FPGA SDK for OpenCL Offline Compiler, and for strategies on area optimization.

Related Information

- [Allocating Shared Memory for OpenCL Kernels Targeting SoCs](#)
- [Strategies for Optimizing FPGA Area Usage](#)
- [Pipelines](#)



2. Setting Up the Intel FPGA SDK for OpenCL, Intel SoC FPGA Embedded Design Suite, and the Cyclone V SoC Development Kit for Windows

The Intel FPGA SDK for OpenCL setup process includes downloading and installing the software, installing the Cyclone V SoC Development Kit, and then executing an OpenCL kernel on the SoC FPGA.

1. [Upgrading to Current Version of Intel FPGA SDK for OpenCL for Cyclone V SoC FPGA](#) on page 11
2. [Downloading the Intel FPGA SDK for OpenCL Standard Edition](#) on page 12
3. [Downloading the Intel SoC FPGA Embedded Development Suite](#) on page 12
4. [Installing the Intel FPGA SDK for OpenCL](#) on page 13
5. [Setting the Intel FPGA SDK for OpenCL User Environment Variables for SoC FPGA](#) on page 13
6. [Installing the Intel SoC FPGA Embedded Development Suite Standard Edition](#) on page 14
7. [Installing the Cyclone V SoC Development Kit](#) on page 15
8. [Downloading an OpenCL Design Example](#) on page 19
9. [Creating the Hardware Configuration File of an OpenCL Kernel for SoC FPGA](#) on page 19
10. [Executing an OpenCL Kernel on an SoC FPGA](#) on page 21
11. [Uninstalling the Software](#) on page 23

2.1. Upgrading to Current Version of Intel FPGA SDK for OpenCL for Cyclone V SoC FPGA

If you have been using a previous version of the Intel FPGA SDK for OpenCL to program your Cyclone V SoC Development Kit, you must upgrade the software programs and files to the current version.

The SDK is incompatible with previous versions of the Intel Quartus® Prime software.

1. Upgrade the following software to the current version:
 - a. Intel Quartus Prime Standard Edition software
 - b. Intel FPGA SDK for OpenCL Standard Edition
 - c. Intel SoC FPGA Embedded Development Suite (EDS) Standard Edition
2. Reprogram the Cyclone V SoC Development Kit micro SD flash card with an image created using the current version of the SDK.

For instructions on programming the micro SD flash card, refer to the *Writing an SD Card Image File onto the Micro SD Flash Card* section.

3. Recompile your host application using the current version of the SDK.

Related Information

[Writing an SD Card Image onto the Micro SD Flash Card](#) on page 15

2.2. Downloading the Intel FPGA SDK for OpenCL Standard Edition

Download the installation package that contains the Intel FPGA SDK for OpenCL Standard Edition and all related software for Windows from the Intel FPGA SDK for OpenCL Download Center.

The Intel FPGA SDK for OpenCL Download Center provides a tar file that includes all of the following software and files:

- Intel FPGA SDK for OpenCL Standard Edition
 - Intel Quartus Prime Standard Edition software
 - Device support
1. Go to the Intel FPGA SDK for OpenCL Download Center at the following URL:
<http://fpgasoftware.intel.com/opencl/>
 2. Select the Standard edition.
 3. Select the software version. The default selection is the current version.
 4. Select one of the following download methods:
 - **Akamai DLM3 Download Manager**
 - **Direct Download**
 5. Click the **Windows SDK** tab.
 6. Click **More** beside **Download and install instructions** if you want to see the download and installation instructions.
 7. Click the download button to start the download process.
 8. Perform the steps outlined in the download and installation instructions on the download page.

Related Information

[Intel FPGA website](#)

2.3. Downloading the Intel SoC FPGA Embedded Development Suite

Download the Intel SoC FPGA Embedded Development Suite (EDS) for Windows from the Download Center on the Intel FPGA website.

1. Go to the SoC EDS Download Center at the following URL:
<http://dl.altera.com/soceds/>
2. Select the Standard edition.
3. Select the software version. The default selection is the current version.
4. Select **Windows** as the operating system.

5. Select **Akamai DLM3 Download Manager** or **Direct Download** as the download method.
6. If you select Akamai DLM3 Download Manager as the download method, click **Download**.
7. If you select Direct Download as the download method, click **Intel SoC FPGA Embedded Development Suite (EDS)**.
8. Perform the steps outlined in the download and installation instructions on the download page.

Related Information

[Intel FPGA website](#)

2.4. Installing the Intel FPGA SDK for OpenCL

In this section, you learn how to unpack the downloaded tar archive and run the installation files to install all the software and files.
Install the Windows version of the Intel FPGA SDK for OpenCL Standard Edition in a folder that you own.

You must have administrator privileges to execute these instructions.

To install the Intel FPGA SDK for OpenCL, Intel Quartus Prime Standard Edition software, and device support files, perform the following tasks:

1. Unpack the downloaded `AOCL-<version>-<build>-windows.tar` archive into a temporary folder.
2. Run the `setup.bat` file to install the SDK with the Intel Quartus Prime Standard Edition software.
3. *Note:* The installer sets the user environment variable `INTELFPGAOCCLSDKROOT` to point to the path of the software installation.

Verify that `INTELFPGAOCCLSDKROOT` points to the current version of the software.
Open a Windows command window and then type `echo %INTELFPGAOCCLSDKROOT%` at the command prompt.

If the returned path does not point to the location of the Intel FPGA SDK for OpenCL installation, edit the `INTELFPGAOCCLSDKROOT` setting.

Related Information

[Setting the Intel FPGA SDK for OpenCL User Environment Variables for SoC FPGA](#) on page 13

2.5. Setting the Intel FPGA SDK for OpenCL User Environment Variables for SoC FPGA

Set the `PATH` user environment variable to point to Intel FPGA SDK for OpenCL libraries.

1. In the Windows **Start** menu, click **Control Panel > System and Security > System**.
2. In the **System** window, click **Advanced system settings**. In the **Advanced** tab of the **System Properties** dialog box, click **Environment Variables**.
3. In the **Environment Variables** dialog box, include the following paths in the corresponding environment variable settings:

Environment Variable	Path to Include
<code>PATH</code>	a. <code>%INTELFPGAOCCLSDKROOT%\bin</code> b. <code>%INTELFPGAOCCLSDKROOT%\windows64\bin</code> c. <code>%INTELFPGAOCCLSDKROOT%\host\windows64\bin</code> where <code>INTELFPGAOCCLSDKROOT</code> points to the path of the software installation.
<code>AOCL_BOARD_PACKAGE_ROOT</code>	Path to the Custom or Reference Platform. For example, the path to the Cyclone V SoC Development Kit Reference Platform is <code>%INTELFPGAOCCLSDKROOT%\board\c5soc</code> .
<code>QUARTUS_ROOTDIR_OVERRIDE</code>	Path to the installation folder of the Intel Quartus Prime Standard Edition software.

2.6. Installing the Intel SoC FPGA Embedded Development Suite Standard Edition

Install the SoC EDS Standard Edition for Windows to build your host application for OpenCL kernel deployment on an SoC board.

1. Run the installer. Follow the installation instructions in the `SoCEDSSetup-<version>-windows.exe` executable. For more information, refer to the *Installing the SoC EDS* section of the *Intel SoC FPGA Embedded Development Suite User Guide*.
2. Perform the tasks outlined in the *Installing the Arm* DS-5* Intel SoC FPGA Edition Toolkit* section of the *Intel SoC FPGA Embedded Development Suite User Guide* to install the Arm Development Studio 5* (DS-5) Intel SoC FPGA Edition Toolkit for your operating system.

For more information on the Arm DS-5 Intel SoC FPGA Edition Toolkit, refer to the Arm DS-5 Intel SoC FPGA Edition page of the ARM website.
3. Consult the *Licensing* section of the *Intel SoC FPGA Embedded Development Suite User Guide* for licensing instructions for the SoC EDS and the Arm DS-5 Intel SoC FPGA Edition Toolkit.

Related Information

- [Installing the SoC EDS](#)
- [Installing the ARM DS-5 SoC FPGA Edition Toolkit](#)
- [ARM DS-5 Development Studio for Intel SoC FPGA Devices](#)
- [SoC EDS Licensing](#)

2.7. Installing the Cyclone V SoC Development Kit

To execute an OpenCL kernel on a Cyclone V SoC FPGA, first install the Cyclone V SoC Development Kit and then apply the Intel FPGA SDK for OpenCL-specific configurations.

For key components of a Cyclone V SoC Development Kit, refer to *Overview of the Intel FPGA SDK for OpenCL Standard Edition and Cyclone V SoC Development Kit Setup Processes*.

1. [Writing an SD Card Image onto the Micro SD Flash Card](#) on page 15
2. [Configuring the SW3 Switches](#) on page 16
3. [Setting Up Terminal Connection](#) on page 17
4. [Setting Environment Variables and Loading OpenCL Linux Kernel Driver](#) on page 17
5. [Connecting the Board to Network via Ethernet](#) on page 17

Related Information

[Overview of the Intel FPGA SDK for OpenCL Standard Edition and Cyclone V SoC Development Kit Setup Processes](#) on page 7

2.7.1. Writing an SD Card Image onto the Micro SD Flash Card

To write an Intel FPGA SDK for OpenCL-compatible SD card image onto the micro SD flash card on Windows, download and install the Win32 Disk Imager, and then write the SD card image onto the micro SD flash card. The SD card image contains everything you need to start using OpenCL on the board.

The SD card image `linux_sd_card_image.tgz` is included in the Cyclone V SoC Development Kit Reference Platform, available with the SDK.

You must have administrator privileges.

1. Extract the files from the `%INTELFPGAOCCLSDKROOT%\board\c5soc\linux_sd_card_image.tgz` archive.
You can use tools such as 7zip or WinZip to extract the SD card image file from the .tgz archive.
2. Download the Win32 Disk Imager from the [SourceForge](#) website.
3. Unzip the Win32 Disk Imager and the SD card image to a directory that you own.
4. Insert the micro SD card into the card reader and connect it to your PC.
5. Launch the Win32 Disk Imager. In the dialog box, under **Image File**, browse to the SD card image file.
6. From the **Device** pull-down menu, select the destination drive of the micro SD card.

Warning: Specifying the wrong device name might cause the SD card image to overwrite all existing data.

7. Click **Write**.
8. After you write the image onto the micro SD flash card, insert the card into the micro SD card slot on the Cyclone V SoC Development Kit.
9. Power up the board.
If the LEDs on the FPGA flash in a counter pattern, the image is written onto the micro SD card successfully. A section of OpenCL logic on the FPGA drives these LEDs.

Related Information

[Cyclone V SoC Development Kit Reference Platform Porting Guide](#)

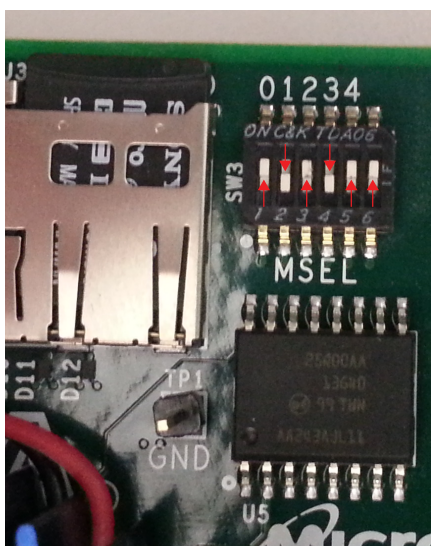
2.7.2. Configuring the SW3 Switches

Configure the SW3 dual in-line package (DIP) switches on the Cyclone V SoC Development Kit for use with the Intel FPGA SDK for OpenCL. The switch bank is located next to the micro SD card slot.

- Set the SW3 DIP switches to the following positions:

Switch	Configuration
1	ON
2	OFF
3	ON
4	OFF
5	ON
6	ON

The figure below illustrates the physical configuration of the SW switches on the Cyclone V SoC Development Kit:



2.7.3. Setting Up Terminal Connection

To set up the terminal connection on the Cyclone V SoC Development Kit for use with the Windows version of the Intel FPGA SDK for OpenCL, specify the USB virtual COM port settings.

1. Connect the board to your development machine via the micro-USB port that is closest to the power supply connector on the board.
2. Connect the board to the power supply and power it up.
3. Download the Virtual COM port (VCP) driver from the [VCP driver download page](#) on the Future Technology Devices International (FTDI) Ltd. website.
4. Determine the COM port in use.
 - a. From the Windows **Start** menu, click **Control Panel > Hardware and Sound**.
 - b. Under **Devices and Printers**, click **Device Manager**.
 - c. In the **Device Manager** window, under **Ports**, click **USB Serial Port (COM<X>)**.
5. Connect either the Tera Term or PuTTY open-source terminal emulator to the COM port that the FTDI driver creates.
6. Set the port settings to 115200, 8N1, with parity and control flow set to none.
7. For Tera Term, select **Setup > Terminal**, and then change Code Page to 1250.
8. Without powering down, restart the board.

2.7.4. Setting Environment Variables and Loading OpenCL Linux Kernel Driver

After you turn on the board and establish terminal connection, log into the Cyclone V SoC Development Kit as user `root` with no password. Then, before you run your host application, set the environment variables and load the OpenCL Linux kernel driver.

- After logging into the SoC FPGA board, run the `source ./init_opencl.sh` command, which performs the following tasks:
 - a. Set the `PATH`, `LD_LIBRARY_PATH`, and `AOCL_BOARD_PACKAGE_ROOT` environment variables.
 - b. Load the OpenCL Linux kernel driver.

The `init_opencl.sh` file is available in the SD card image that you write onto the micro SD flash card. It contains the commands shown below:

```
export INTELFGPAOCLSDKROOT=<aocl_destination_directory>
export AOCL_BOARD_PACKAGE_ROOT=$INTELFGPAOCLSDKROOT/board/c5soc
export PATH=$INTELFGPAOCLSDKROOT/bin:$PATH
export LD_LIBRARY_PATH=$INTELFGPAOCLSDKROOT/host/arm32/lib:$LD_LIBRARY_PATH
insmod $AOCL_BOARD_PACKAGE_ROOT/driver/aclsoc_drv.ko
```

2.7.5. Connecting the Board to Network via Ethernet

Connecting the Cyclone V SoC Development Kit to the host network allows you to transfer files to and from your SoC FPGA.

1. Connect the HPS Ethernet port of the board to your network.
2. Reboot the board.
The board acquires an IP address, allowing you to initiate a Secure Shell (SSH) connection and runs a Secure Copy (SCP) program to login and transfer files.

Alternatively, you can mount a network drive to your board and access the files directly.

2.7.5.1. Ensuring IP Address Acquisition

After you connect the HPS Ethernet port on the Cyclone V SoC Development Kit to your network and reboot the board, ensure that the board acquires an IP address successfully.

After you connect the HPS Ethernet port to your network and power up your board, you should see a solid orange light and a blinking green light. If not, check the connection of the Ethernet cable to the Ethernet port on your network.

1. To check if your board has an IP address, search for the IP address in boot messages such as the one shown below:

```
Sending discover...
libphy: stmmac-0:04 - Link is Up - 1000/Full
Sending discover...
Sending select for 137.57.175.148...
Lease of 137.57.175.148 obtained, lease time 86400
/etc/udhcpd.d/50default: Adding DNS 137.57.142.218
/etc/udhcpd.d/50default: Adding DNS 137.57.109.10
/etc/udhcpd.d/50default: Adding DNS 137.57.64.1
done.
```

The message Lease of <board_IP_address> obtained, lease time 86400 identifies the IP address of the board.

2. If you receive the following output, perform a warm reboot of the board by pressing the WARM button next to the LED lights.

```
Sending discover...
libphy: stmmac-0:04 - Link is Up - 1000/Full
Sending discover...
Sending discover...
No lease, failing
```

The board uses the dynamic host configuration protocol (DHCP) to acquire an IP address. If the session times out waiting for an IP assignment, reboot the CPU to restart the IP acquisition process. To reboot the CPU, press the Warm reset button next to the four HPS LEDs on the board.

3. If you are unable to acquire the IP address, ensure that the Ethernet cable is in good working condition and the Ethernet port on your network is enabled.

2.7.5.2. Using SSH and SCP

Instead of connecting the Cyclone V SoC Development Kit to the host system using UART over USB and transferring files using a network drive, you can initiate a Secure Shell connection and transfer files using a Secure Copy program.

1. To establish a connection between the Cyclone V SoC Development Kit and the host system via SSH, invoke the `ssh root@<board_ip_address>` command from your development machine.
For instructions on how to identify `<board_ip_address>`, refer to the *Ensuring IP Address Acquisition* section.
2. To transfer files, one at a time, from the host system to the board via SCP, invoke the `scp <source_filename> root@<board_ip_address>:<target_filename>` command from your development machine.

Related Information

[Ensuring IP Address Acquisition](#) on page 18

2.8. Downloading an OpenCL Design Example

The OpenCL Design Examples page contains sample applications of varying complexities that you can download and run on your FPGA.

The following instructions are for downloading the Hello World design.

1. In the OpenCL Design Examples page, under Basic Examples, click **Hello World**.
2. In the Hello World Design Example page, under Downloads, click **<version> x64 Windows package (.zip)** to download the compressed file for your platform.
3. In the Hello World Design Example page, under Downloads, click **<version> arm32 (Linux and Windows) package (.tgz)** to download the compressed file for your platform.
4. Unzip the `exm_openc1_hello_world_x64_windows_<version>.zip` file and save it in a folder that you have write access for.
Important: Ensure that the folder name does not contain spaces.
5. Unpack the `exm_openc1_hello_world_arm32_<version>.tgz` file and save it in a folder that you have write access for.

Important: Ensure that the folder name does not contain spaces.

Related Information

[OpenCL Design Examples](#)

2.9. Creating the Hardware Configuration File of an OpenCL Kernel for SoC FPGA

The hardware configuration file of an OpenCL kernel is in the form of a `.aocx` executable file. To create the `.aocx` file for the *hello_world* example OpenCL application, you must first download and extract the example design from the OpenCL Design Examples page on the Intel FPGA website. Then, compile the `hello_world.cl` kernel source file using the Intel FPGA SDK for OpenCL Offline Compiler.

After you successfully install your FPGA board, you can create a .aocx file that executes on the device. [Figure 3](#) on page 9 outlines the procedure for programming the hello_world design example onto the Cyclone V SoC FPGA. For more information on the OpenCL design examples, refer to the OpenCL Design Examples page.

Important: Before you program your Cyclone V SoC FPGA with the hardware image, ensure that your SoC FPGA contains an image created using the current version of the SDK.

1. Verify that you set the environment variable `AOCL_BOARD_PACKAGE_ROOT` to point to the Cyclone V SoC Development Kit Reference Platform (that is, `INTELFPGAOCCLSDKROOT/board/c5soc`).
The `init_openc1` script you ran when installing the Cyclone V SoC Development Kit should have set the `AOCL_BOARD_PACKAGE_ROOT` environment variable. If the returned path does not include `INTELFPGAOCCLSDKROOT/board/c5soc`, modify the `AOCL_BOARD_PACKAGE_ROOT` setting.
2. Verify that you set the environment variable `QUARTUS_ROOTDIR_OVERRIDE` to point to the installation directory of the correct Intel Quartus Prime Standard Edition software.
3. At a command prompt, navigate to the hello_world design.
4. To list the SoC FPGA boards available in the Cyclone V SoC Development Kit Reference Platform, invoke the `aoc -list-boards` command.
You should see an output similar to the one below:

```
Board list:
c5soc
c5soc_sharedonly
```

5. To compile the kernel for your Cyclone V SoC Development Kit, invoke the following command:

```
aoc -board=c5soc device/hello_world.cl -o=bin/
hello_world.aocx
```

This command performs the following tasks:

- Generates the Intel Quartus Prime design project files from the OpenCL source code.
- Checks for initial syntax errors.
- Performs basic optimizations.
- Creates a `hello_world` subfolder or subdirectory containing necessary intermediate files.
- Creates the .aoco object file.
- Creates the .aocx hardware configuration file and saves it in the `bin` subfolder or subdirectory.

Attention: The .aocx file might take hours to build, depending on the complexity of the kernel. To view the progress of the compilation on-screen, include the `-v` flag in your `aoc` command (that is, `aoc -v <your_kernel_filename>.cl`).

The offline compiler displays the line `aoc: Hardware generation completed successfully.` to signify the completion of the compilation process.

For more information on the `-list-boards`, `-board=<board_name>`, `-v`, and `-o=<filename>` options of the `aoc` command, refer to the *Intel FPGA SDK for OpenCL Standard Edition Programming Guide*.

Related Information

- [OpenCL Design Examples](#)
- [Listing the Available FPGA Boards in Your Custom Platform \(-list-boards\)](#)
- [Compiling a Kernel for a Specific FPGA Board \(-board=<board_name>\)](#)
- [Generating Compilation Progress Report \(-v\)](#)
- [Specifying the Name of an Intel FPGA SDK Offline Compiler Output File \(-o=<filename>\)](#)
- [Setting the Intel FPGA SDK for OpenCL User Environment Variables for SoC FPGA](#) on page 13
- [Setting Environment Variables and Loading OpenCL Linux Kernel Driver](#) on page 17

2.10. Executing an OpenCL Kernel on an SoC FPGA

Build your host application using the GCC cross-compiler available with the SoC EDS.

2.10.1. Building the Host Application

Build your SoC FPGA-specific OpenCL host application using the GCC cross-compiler available with the Windows version of the SoC EDS.

1. Perform the following tasks to download the `hello_world` design example.
 - a. Download the SoC FPGA-specific `hello_world` design example (**<version> Arm32 Linux package (.tgz)**) from the Hello World Design Example page.
 - b. Extract `exm_openc1_hello_world_arm32_linux_<version>.tar` to a location to which you have write access.
Important: Ensure that the location name does not contain spaces.
 - c. Verify that the `AOCL_BOARD_PACKAGE_ROOT` environment variable setting points to the Intel Arria® 10 SoC Development Kit Reference Platform. Open a Windows command window and then type `echo %AOCL_BOARD_PACKAGE_ROOT%` at the command prompt.

If the returned path is not `%INTELFPGAOCCLSDKROOT%\board\al0soc`, or if `AOCL_BOARD_PACKAGE_ROOT` is not set, modify the environment variable setting.

2. At a command prompt, invoke the following command to set the `PATH` environment variable:

```
SET PATH=%PATH%;<path_to_SoCEDs_installation_dir>\ds-5\sw\gcc
\bin
```

3. Navigate to the
`<path_to_exm_openc1_hello_world_arm32_linux_<version>>`
`\hello_world` folder.
4. Invoke the `make -f Makefile` command. Alternatively, you can simply invoke the `make` command.
 The `hello_world` executable will be in the
`<path_to_exm_openc1_hello_world_arm32_linux_<version>>`
`\hello_world\bin` folder.

Related Information

- [Hello World Design Example](#)
- [Setting the Intel FPGA SDK for OpenCL User Environment Variables for SoC FPGA on page 13](#)
- [Setting Environment Variables and Loading OpenCL Linux Kernel Driver on page 17](#)

2.10.2. Running the Host Application

For Windows systems, execute the `hello_world.aocx` executable file on the SoC FPGA by running the host application you built from the ARM®-specific Makefile.

1. Log into your SoC FPGA board.
2. Copy the `hello_world.aocx` hardware configuration file and the `hello_world` host executable from their current folders to the board.
3. Verify that the `LD_LIBRARY_PATH` environment variable setting includes `%INTELFPGAOCCLSDKROOT%\host\arm32\lib`. Run the command `echo $LD_LIBRARY_PATH`.
 If you ran the `init_openc1.sh` script, the `LD_LIBRARY_PATH` setting should point to `%INTELFPGAOCCLSDKROOT%\host\arm32\lib`.
4. To execute the kernel on the SoC FPGA, at a command prompt, navigate to the host executable folder and run the `hello_world` host executable.

Related Information

- [Setting the Intel FPGA SDK for OpenCL User Environment Variables for SoC FPGA on page 13](#)
- [Setting Environment Variables and Loading OpenCL Linux Kernel Driver on page 17](#)

2.10.3. Output from Successful Kernel Execution on the Cyclone V SoC Development Kit

When you run the host application to execute your OpenCL kernel on the Cyclone V SoC Development Kit, the software notifies you of a successful kernel execution.

Example output:

```
Reprogramming device [0] with handle 1
Querying platform for info:
=====
CL_PLATFORM_NAME           = Intel(R) FPGA SDK for OpenCL(TM)
CL_PLATFORM_VENDOR         = Intel Corporation
CL_PLATFORM_VERSION        = OpenCL 1.0 Intel(R) FPGA SDK for OpenCL(TM), <version>

Querying device for info:
=====
CL_DEVICE_NAME              = c5soc : 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>
CL_DRIVER_VERSION           = <version>
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   = 2147483648
CL_DEVICE_IMAGE_SUPPORT     = false
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 = 3758096384
CL_DEVICE_MAX_WORK_ITEM_DIMENSIONS = 3
CL_DEVICE_MAX_WORK_ITEM_DIMENSIONS = 1024
CL_DEVICE_MIN_DATA_TYPE_ALIGN_SIZE = 128
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

Kernel initialization is complete.
Launching the kernel...

Thread #2: Hello from the Intel(R) FPGA SDK for OpenCL(TM) Compiler!

Kernel execution is complete.
```

2.11. Uninstalling the Software

To uninstall the Intel FPGA SDK for OpenCL Standard Edition for Windows, run the uninstaller, and restore all modified environment variables to their previous settings.

1. From the Windows Start Menu shortcut, navigate to the Altera <version> Standard Edition installation folder.
2. Select **Uninstall Intel Quartus Prime Standard Edition**.
The uninstallation wizard appears.
3. In the uninstallation wizard, perform the following tasks:
 - a. Select **Individual components** and then click **Next**.
 - b. Select the SDK and then click **Next**.
The uninstaller uninstalls the Intel FPGA SDK for OpenCL.
4. Remove the following paths from the *PATH* environment variable:
 - a. %INTELFPGAOCSDKROOT%\bin



2. Setting Up the Intel FPGA SDK for OpenCL, Intel SoC FPGA Embedded Design Suite, and the Cyclone V SoC Development Kit for Windows

683030 | 2018.09.24

- b. `%INTELFPGAOCSDKROOT%\host\windows64\bin`
- 5. Remove the `INTELFPGAOCSDKROOT` environment variable.
- 6. Remove the `QUARTUS_ROOTDIR_OVERRIDE` environment variable.



3. Setting Up the Intel FPGA SDK for OpenCL, Intel SoC FPGA Embedded Design Suite, and the Cyclone V SoC Development Kit for Linux

The Intel FPGA SDK for OpenCL setup process includes downloading and installing the software, installing the Cyclone V SoC Development Kit, and then executing an OpenCL kernel on the SoC.

1. [Upgrading to Current Version of Intel FPGA SDK for OpenCL for Cyclone V SoC FPGA](#) on page 25
2. [Downloading the Intel FPGA SDK for OpenCL Standard Edition](#) on page 26
3. [Downloading the Intel SoC FPGA Embedded Development Suite](#) on page 26
4. [Installing the Intel FPGA SDK for OpenCL](#) on page 27
5. [Setting the Intel FPGA SDK for OpenCL User Environment Variables for SoC FPGA](#) on page 28
6. [Installing the Intel SoC FPGA Embedded Development Suite Pro Edition](#) on page 28
7. [Installing the Cyclone V SoC Development Kit](#) on page 29
8. [Verifying Host Runtime Functionality via Emulation](#) on page 34
9. [Creating the Hardware Configuration File of an OpenCL Kernel for SoC FPGA](#) on page 38
10. [Executing an OpenCL Kernel on an SoC FPGA](#) on page 39
11. [Uninstalling the Software](#) on page 41

3.1. Upgrading to Current Version of Intel FPGA SDK for OpenCL for SoC FPGA

If you have been using a previous version of the Intel FPGA SDK for OpenCL to program your Cyclone V SoC Development Kit, you must upgrade the software programs and files to the current version.

The SDK is incompatible with previous versions of the software.

1. Upgrade the following software to the current version:
 - a. software
 - b. Standard Edition
 - c. SoC FPGA Embedded Development Suite (EDS) Standard Edition
2. Reprogram the SoC Development Kit micro SD flash card with an image created using the current version of the SDK.

For instructions on programming the micro SD flash card, refer to the *Writing an SD Card Image File onto the Micro SD Flash Card* section.

3. Recompile your host application using the current version of the SDK.

Related Information

[Writing an SD Card Image onto the Micro SD Flash Card](#) on page 29

3.2. Downloading the Intel FPGA SDK for OpenCL Standard Edition

Download the installation package that contains the Intel FPGA SDK for OpenCL Standard Edition and all related software for Linux from the Intel FPGA SDK for OpenCL Download Center.

The Download Center provides a tar file that includes all of the following software and files:

- Intel FPGA SDK for OpenCL Standard Edition
 - Intel Quartus Prime Standard Edition software
 - Device support
1. Go to the Intel FPGA SDK for OpenCL Download Center at the following URL:
<http://fpgasoftware.intel.com/opencl/>
 2. Select the Standard edition.
 3. Select the software version. The default selection is the current version.
 4. Select one of the following download methods:
 - **Akamai DLM3 Download Manager**
 - **Direct Download**
 5. On the **Linux SDK** tab.
 6. Click **More** beside **Download and install instructions** if you want to see the download and installation instructions.
 7. Perform the steps outlined in the download and installation instructions on the download page.

Related Information

[Intel FPGA website](#)

3.3. Downloading the Intel SoC FPGA Embedded Development Suite

Download the Intel SoC FPGA Embedded Development Suite (EDS) for Linux from the Download Center on the Intel FPGA website.

1. Go to the SoC EDS Download Center at the following URL:
<http://dl.altera.com/soceds/>
2. Select the Standard edition.
3. Select the software version. The default selection is the current version.
4. Select **Linux** as the operating system.

5. Select **Akamai DLM3 Download Manager** or **Direct Download** as the download method.
6. If you select Akamai DLM3 Download Manager as the download method, click **Download**.
7. If you select Direct Download as the download method, click **Intel SoC FPGA Embedded Development Suite (EDS)**.
8. Perform the steps outlined in the download and installation instructions on the download page.

Related Information

[Intel FPGA website](#)

3.4. Installing the Intel FPGA SDK for OpenCL

Unpack the downloaded tar file and run the installation files to install all the software and files.

Install the Linux version of the Intel FPGA SDK for OpenCL Standard Edition in a directory that you own.

- You must have `sudo` or `root` privileges.
- You must install the Linux OS kernel source and headers (for example, `kernel-devel.x86_64` and `kernel-headers.x86_64`), and the GNU Compiler Collection (GCC) (`gcc.x86_64`).
- If you are installing a package that includes Intel Code Builder, you must have Java SE 1.8.71 or later installed to run Intel Code Builder. If you have an earlier version of Java SE installed, you can still complete the installation of Intel Code Builder. However, you must meet the Java version prerequisite to run Intel Code Builder.

Attention: If you install the software on a system that does not contain any C Shell Run Commands file (`.cshrc`) or Bash Run Commands file (`.bashrc`) in your directory, you must set the environment variables `INTELFPGAOCSDKROOT` and `PATH` manually. Alternatively, you may create the `.cshrc` and `.bashrc` files, and then append the environment variables to them. To ensure that the updates take effect, restart your terminal after you set the environment variables.

To install the Intel FPGA SDK for OpenCL, Intel Quartus Prime Standard Edition software, and device support files simultaneously, perform the following tasks:

1. Unpack the downloaded tar file into a temporary directory.
2. Run the `setup.sh` file to install the SDK with the Intel Quartus Prime Standard Edition software.
3. *Note:* The installer sets the environment variable `INTELFPGAOCSDKROOT` to point to the path of the software installation.

Verify that `INTELFPGAOCSDKROOT` points to the current version of the software. Open a shell and then type `echo $INTELFPGAOCSDKROOT` at the command prompt.

If the returned path does not point to the location of the Intel FPGA SDK for OpenCL installation, edit the `INTELFPGAOCSDKROOT` setting.

Related Information

Setting the Intel FPGA SDK for OpenCL User Environment Variables for SoC FPGA on page 28

3.5. Setting the Intel FPGA SDK for OpenCL User Environment Variables for SoC FPGA

Set the *PATH* and *LD_LIBRARY_PATH* user environment variables to point to the Intel FPGA SDK for OpenCL and board libraries.

Attention: If you install the software on a system that does not contain any C Shell Run Commands file (*.cshrc*) or Bash Run Commands file (*.bashrc*) in your directory, you must set the environment variables *INTELFPGAOCSDKROOT* and *PATH* manually. Alternatively, you may create the *.cshrc* and *.bashrc* files, and then append the environment variables to them. To ensure that the updates take effect, restart your terminal after you set the environment variables.

1. At a command prompt, type `export PATH=$INTELFPGAOCSDKROOT/bin:$PATH` to add the path to the SDK bin directory to *PATH*, where *INTELFPGAOCSDKROOT* points to the path to the software installation.
2. At a command prompt, type `export AOCL_BOARD_PACKAGE_ROOT=$INTELFPGAOCSDKROOT/board/c5soc` to set *AOCL_BOARD_PACKAGE_ROOT* to point to the Cyclone V SoC Development Kit Reference Platform.
3. At a command prompt, specify the following environment variable settings:
 - a. Invoke the `export LD_LIBRARY_PATH=$INTELFPGAOCSDKROOT/host/linux64/lib:$LD_LIBRARY_PATH` command to add the path to the SDK host runtime libraries to *LD_LIBRARY_PATH*.
 - b. Invoke the `export LD_LIBRARY_PATH=$AOCL_BOARD_PACKAGE_ROOT/linux64/lib:$LD_LIBRARY_PATH` command to add the path to the board libraries to *LD_LIBRARY_PATH*, where *AOCL_BOARD_PACKAGE_ROOT* points to the path to the Custom or Reference Platform.
4. At a command prompt, type `export QUARTUS_ROOTDIR_OVERRIDE=<Quartus Prime software installation directory>`, where *<Quartus Prime software installation directory>* is the path to the installation directory of the Intel Quartus Prime Standard Edition software.

3.6. Installing the Intel SoC FPGA Embedded Development Suite Pro Edition

Install the Intel SoC EDS for Linux to build your host application for OpenCL kernel deployment on an SoC FPGA board.

The GCC tool chain is part of the SoC EDS installation package.

1. Run the `SoCEDSSetup-<version>-linux.run` installer. For more information, refer to the *Installing the SoC EDS* section of the *Intel SoC FPGA Embedded Development Suite User Guide*.
2. Perform the tasks outlined in the *Installing the Arm DS-5 Intel SoC FPGA Edition Toolkit* section of the *Intel SoC FPGA Embedded Development Suite User Guide* to install the Arm Development Studio 5 (DS-5) Intel SoC FPGA Edition Toolkit for your operating system.

For more information on the Arm DS-5 Intel SoC FPGA Edition Toolkit, refer to the Arm DS-5 Intel SoC FPGA Edition page of the ARM website.

3. Consult the *Licensing* section of the *Intel SoC FPGA Embedded Development Suite User Guide* for licensing instructions for the SoC EDS and the Arm DS-5 Intel SoC FPGA Edition Toolkit.

Related Information

- [Installing the SoC EDS](#)
- [Installing the ARM DS-5 SoC FPGA Edition Toolkit](#)
- [ARM DS-5 Development Studio for Intel SoC FPGA Devices](#)
- [SoC EDS Licensing](#)

3.7. Installing the Cyclone V SoC Development Kit

To execute an OpenCL kernel on a Cyclone V SoC FPGA, first install the Cyclone V SoC Development Kit and then apply the Intel FPGA SDK for OpenCL-specific configurations.

For key components of a Cyclone V SoC Development Kit, refer to *Overview of the Intel FPGA SDK for OpenCL Standard Edition and Cyclone V SoC Development Kit Setup Processes*.

1. [Writing an SD Card Image onto the Micro SD Flash Card](#) on page 29
2. [Configuring the SW3 Switches](#) on page 30
3. [Setting Up Terminal Connection](#) on page 31
4. [Setting Environment Variables and Loading OpenCL Linux Kernel Driver](#) on page 32
5. [Connecting the Board to Network via Ethernet](#) on page 32

Related Information

[Overview of the Intel FPGA SDK for OpenCL Standard Edition and Cyclone V SoC Development Kit Setup Processes](#) on page 7

3.7.1. Writing an SD Card Image onto the Micro SD Flash Card

To write an Intel FPGA SDK for OpenCL-compatible SD card image onto the micro SD flash card on Linux, extract the SD card image from the Cyclone V SoC Development Kit Reference Platform, and then write the image onto the micro SD flash card. The SD card image contains everything you need to start using OpenCL on the board.

The SD card image `linux_sd_card_image.tgz` is included in the Cyclone V SoC Development Kit Reference Platform, available with the SDK. Ensure that the environment variable `AOCL_BOARD_PACKAGE_ROOT` points to the location of the `board_env.xml` file in the Reference Platform.

You must have `sudo` or root privileges.

1. To decompress the `$INTELFPGAOCCLSDKROOT/board/c5soc/linux_sd_card_image.tgz` file, run the `tar xvfz linux_sd_card_image.tgz` command.
2. Insert the micro SD flash card into a card reader, and connect the reader to your PC.
 - a. If the flash card already contains an image, partitions will exist automatically in the micro SD card. Unmount or eject all these partitions.
3. Run the `dmesg | tail` command to verify the device name of the flash card (for example, `/dev/sde`).
4. Write the SD card image onto the micro SD flash card by running the following commands:

```
sudo dd if=linux_sd_card_image of=/dev/sde bs=1M
sync
```

Attention: If the device name of your micro SD flash card is not `/dev/sde`, replace `/dev/sde` in the above command with the device name you obtain from Step 3.

Warning: Specifying the wrong device name might cause the SD card image to overwrite all existing data.

5. After you write the image onto the micro SD flash card, insert the card into the micro SD card slot on the Cyclone V SoC Development Kit.

Related Information

[Cyclone V SoC Development Kit Reference Platform Porting Guide](#)

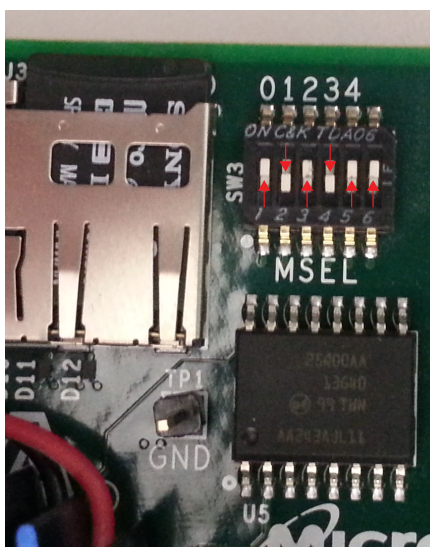
3.7.2. Configuring the SW3 Switches

Configure the SW3 dual in-line package (DIP) switches on the Cyclone V SoC Development Kit for use with the Intel FPGA SDK for OpenCL. The switch bank is located next to the micro SD card slot.

- Set the SW3 DIP switches to the following positions:

Switch	Configuration
1	ON
2	OFF
3	ON
4	OFF
5	ON
6	ON

The figure below illustrates the physical configuration of the SW switches on the Cyclone V SoC Development Kit:



3.7.3. Setting Up Terminal Connection

To set up the terminal connection on the Cyclone V SoC Development Kit for use with the Linux version of the Intel FPGA SDK for OpenCL, specify the USB virtual COM port settings.

- Connect the board to your development machine via the micro-USB port that is closest to the power supply connector on the board.
- Connect the board to the power supply and power it up.
- Run the command `dmesg | tail` to determine which device the Future Technology Devices International (FTDI) driver assigns for the connection (e.g. `/dev/ttyUSB0`).
- Setup the minicom as follows:
 - Ensure that minicom is installed on your system. If not, invoke the `yum install minicom` command.
 - Run `minicom -s` as root to enter the minicom setup mode.
 - Select **Serial port setup** and then press Enter.

- d. Press A to change **Serial Device** to `/dev/ttyUSB0` and then press Enter.
 - e. Press E to change the port settings. Press E again to select **115200** for **Speed**, and then press Q to set **Data/Parity/Stopbits** to **8-N-1** configuration.
 - f. Press Enter twice to return to the main minicom setup menu.
 - g. Select **Save setup as dfl** and then press Enter to save the minicom settings as defaults.
 - h. Select **Exit**.
5. Without powering down, restart the board.
You should see Linux boot messages appear on the terminal command of your choice.

3.7.4. Setting Environment Variables and Loading OpenCL Linux Kernel Driver

After you turn on the board and establish terminal connection, log into the Cyclone V SoC Development Kit as user `root` with no password. Then, before you run your host application, set the environment variables and load the OpenCL Linux kernel driver.

- After logging into the SoC FPGA board, run the source `./init_opencl.sh` command, which performs the following tasks:
 - a. Set the `PATH`, `LD_LIBRARY_PATH`, and `AOCL_BOARD_PACKAGE_ROOT` environment variables.
 - b. Load the OpenCL Linux kernel driver.

The `init_opencl.sh` file is available in the SD card image that you write onto the micro SD flash card. It contains the commands shown below:

```
export INTELFGAOCLSDKROOT=<aocl_destination_directory>
export AOCL_BOARD_PACKAGE_ROOT=$INTELFGAOCLSDKROOT/board/c5soc
export PATH=$INTELFGAOCLSDKROOT/bin:$PATH
export LD_LIBRARY_PATH=$INTELFGAOCLSDKROOT/host/arm32/lib:$LD_LIBRARY_PATH
insmod $AOCL_BOARD_PACKAGE_ROOT/driver/aclsoc_drv.ko
```

3.7.5. Connecting the Board to Network via Ethernet

Connecting the Cyclone V SoC Development Kit to the host network allows you to transfer files to and from your SoC FPGA.

1. Connect the HPS Ethernet port of the board to your network.
2. Reboot the board.
The board acquires an IP address, allowing you to initiate a Secure Shell (SSH) connection and runs a Secure Copy (SCP) program to login and transfer files.

Alternatively, you can mount a network drive to your board and access the files directly.

3.7.5.1. Ensuring IP Address Acquisition

After you connect the HPS Ethernet port on the Cyclone V SoC Development Kit to your network and reboot the board, ensure that the board acquires an IP address successfully.

After you connect the HPS Ethernet port to your network and power up your board, you should see a solid orange light and a blinking green light. If not, check the connection of the Ethernet cable to the Ethernet port on your network.

1. To check if your board has an IP address, search for the IP address in boot messages such as the one shown below:

```
Sending discover...
libphy: stmmac-0:04 - Link is Up - 1000/Full
Sending discover...
Sending select for 137.57.175.148...
Lease of 137.57.175.148 obtained, lease time 86400
/etc/udhcpd.d/50default: Adding DNS 137.57.142.218
/etc/udhcpd.d/50default: Adding DNS 137.57.109.10
/etc/udhcpd.d/50default: Adding DNS 137.57.64.1
done.
```

The message Lease of <board_IP_address> obtained, lease time 86400 identifies the IP address of the board.

2. If you receive the following output, perform a warm reboot of the board by pressing the WARM button next to the LED lights.

```
Sending discover...
libphy: stmmac-0:04 - Link is Up - 1000/Full
Sending discover...
Sending discover...
No lease, failing
```

The board uses the dynamic host configuration protocol (DHCP) to acquire an IP address. If the session times out waiting for an IP assignment, reboot the CPU to restart the IP acquisition process. To reboot the CPU, press the Warm reset button next to the four HPS LEDs on the board.

3. If you are unable to acquire the IP address, ensure that the Ethernet cable is in good working condition and the Ethernet port on your network is enabled.

3.7.5.2. Mounting a Shared Drive

The most convenient way to share files between a development PC and the Cyclone V SoC FPGA development board is to mount a network drive.

1. Check the `/etc/fstab` file systems table file on your development PC for the line that describes the mounting of the drive you want to use on the board.

The following example `/etc/fstab` entry indicates that the `/data` folder on the `my_nas` server is mounted to the `/data` folder on the development PC:

```
my_nas:/data /data nfs
exec,dev,suid,rw,tcp,hard,intr,vers=3,rsz=32768,wsz=32768,
timeo=600,retrans=200
```

2. Add the `/etc/fstab` entry described above to the `/etc/fstab` file on the Cyclone V SoC development board.
3. Run the `sync` command to save the `/etc/fstab` file to the micro SD flash card.
4. Create an empty folder on the board that serves as the mounting point for the network drive.

For example: type `mkdir /data`, where `/data` is the name of the folder.

5. Invoke the `busybox mount -a` command.
If the mounting operation fails, rerun the command.

3.7.5.3. Using SSH and SCP

Instead of connecting the Cyclone V SoC Development Kit to the host system using UART over USB and transferring files using a network drive, you can initiate a Secure Shell connection and transfer files using a Secure Copy program.

1. To establish a connection between the Cyclone V SoC Development Kit and the host system via SSH, invoke the `ssh root@<board_ip_address>` command from your development machine.
For instructions on how to identify `<board_ip_address>`, refer to the *Ensuring IP Address Acquisition* section.
2. To transfer files, one at a time, from the host system to the board via SCP, invoke the `scp <source_filename> root@<board_ip_address>:<target_filename>` command from your development machine.

Related Information

[Ensuring IP Address Acquisition](#) on page 32

3.8. Verifying Host Runtime Functionality via Emulation

Test the functionality of the host runtime by emulating an OpenCL design example using the Intel FPGA SDK for OpenCL Emulator.

Attention: Emulation is a Linux-only optional step when setting up the SDK for use with Intel SoC FPGAs. Specifically, you must emulate your OpenCL kernel on a non-SoC FPGA board before executing your kernel on the SoC FPGA.

1. Install a Custom or Reference Platform because emulation targets a specific FPGA board.
2. Verify that the `QUARTUS_ROOTDIR_OVERRIDE` environment variable points to the Intel Quartus Prime Standard Edition software. Open a shell and then type `echo $QUARTUS_ROOTDIR_OVERRIDE` at the command prompt.
If the path to the installation directory of the Intel Quartus Prime Standard Edition software is not returned, add it to the `QUARTUS_ROOTDIR_OVERRIDE` setting.
3. Verify that the `LD_LIBRARY_PATH` environment variable setting includes the paths identified in *Setting the Intel FPGA SDK for OpenCL User Environment Variables for SoC FPGA*. Open a shell and then type `echo $LD_LIBRARY_PATH` at the command prompt.
If the returned path do not include `$INTELFPGAOCCLSDKROOT/host/linux64/lib`, add it to the `LD_LIBRARY_PATH` setting.
4. Verify that the `LD_LIBRARY_PATH` environment variable setting includes the paths identified in *Setting the Intel FPGA SDK for OpenCL User Environment Variables for SoC FPGA*. Open a shell and then type `echo $LD_LIBRARY_PATH` at the command prompt.

If the returned path do not include `$INTELFPGAOCCLSDKROOT/host/linux64/lib`, add it to the `LD_LIBRARY_PATH` setting.

Each invocation of the emulated kernel creates a shared library copy called `<process_ID>-libkernel.so` in a default temporary directory, where `<process_ID>` is a unique numerical value assigned to each emulation run. You may override the default directory by setting the `TMPDIR` environment variable.

Related Information

[Setting the Intel FPGA SDK for OpenCL User Environment Variables for SoC FPGA](#) on page 28

3.8.1. Downloading an OpenCL Design Example

The OpenCL Design Examples page contains sample applications of varying complexities that you can download and run on your FPGA.

The following instructions are for downloading the Hello World design.

1. In the OpenCL Design Examples page, under Basic Examples, click **Hello World**.
2. In the Hello World Design Example page, under Downloads, click **<version> x64 Linux package (.tgz)** to download the compressed file for your platform.
3. In the Hello World Design Example page, under Downloads, click **<version> arm32 (Linux and Windows) package (.tgz)** to download the compressed file for your platform.
4. Unpack the `.tgz` file and save it in a directory to which you have write access.

Important: Ensure that the directory name does not contain spaces.

Related Information

[OpenCL Design Examples](#)

3.8.2. Compiling a Kernel for Emulation (-march=emulator)

To compile an OpenCL kernel for emulation, include the `-march=emulator` option in your `aoc` command.

- Before you perform kernel emulation, perform the following tasks:
 - Install a Custom Platform from your board vendor for your FPGA accelerator boards.
 - Verify that the environment variable `QUARTUS_ROOTDIR_OVERRIDE` points to Intel Quartus Prime Pro Edition software installation folder.
- To emulate your kernels on Windows systems, you need the Microsoft linker and additional compilation time libraries. Verify that the `PATH` environment variable setting includes all the paths described in the *Setting the Intel FPGA SDK for OpenCL Pro Edition User Environment Variables* section of the *Intel FPGA SDK for OpenCL Pro Edition Getting Started Guide*.
 The `PATH` environment variable setting must include the path to the `LINK.EXE` file in Microsoft Visual Studio.
- Ensure that your `LIB` environment variable setting includes the path to the Microsoft compilation time libraries.
 The compilation time libraries are available with Microsoft Visual Studio.
- Verify that the `LD_LIBRARY_PATH` environment variable setting includes all the paths described in the *Setting the Intel FPGA SDK for OpenCL Pro Edition User Environment Variables* section in the *Intel FPGA SDK for OpenCL Pro Edition Getting Started Guide*.
- To create kernel programs that are executable on x86-64 host systems, invoke the `aoc -march=emulator <your_kernel_filename>.cl` command.
- To compile a kernel for emulation that targets a specific board, invoke the `aoc -march=emulator -board=<board_name> <your_kernel_filename>.cl` command.
- For Linux systems, the Intel FPGA SDK for OpenCL Offline Compiler offers symbolic debug support for the debugger.
 The offline compiler's debug support allows you to pinpoint the origins of functional errors in your kernel source code.

3.8.3. Building the Host Application

Build your SoC FPGA-specific OpenCL host application using the GCC cross-compiler available with the Linux version of the SoC EDS.

- To build your host application for emulation, modify the `AOCL_BOARD_PACKAGE_ROOT` environment variable setting to point to a non-SoC Reference or Custom Platform. Verify the setting by opening a shell and then typing `echo $AOCL_BOARD_PACKAGE_ROOT` at the command prompt.
 - To build your host application for kernel execution, verify that the `AOCL_BOARD_PACKAGE_ROOT` environment variable setting points to the Intel Arria 10 SoC Development Kit Reference Platform.
1. At a command prompt, invoke the following command to set the `PATH` environment variable:

```
export PATH=<path_to_SoCEDS_installation_dir>/ds-5/sw/gcc/  
bin:$PATH
```

2. Navigate to the
`<path_to_exm_opencl_hello_world_arm32_linux_<version>>/hello_world` directory.
3. Invoke the `make -f Makefile` command. Alternatively, you can simply invoke the `make` command.
The `hello_world` executable will be in the
`<path_to_exm_opencl_hello_world_arm32_linux_<version>>/hello_world/bin` directory.

Related Information

- [Hello World Design Example](#)
- [Setting the Intel FPGA SDK for OpenCL User Environment Variables for SoC FPGA](#) on page 28
- [Setting Environment Variables and Loading OpenCL Linux Kernel Driver](#) on page 32

3.8.4. Emulating Your OpenCL Kernel

To emulate your OpenCL kernel, run the emulation `.aocx` file on the platform on which you build your kernel.

To emulate your kernel, perform the following steps:

1. Run the utility command `aocl linkflags` to find out which libraries are necessary for building a host application. The software lists the libraries for both emulation and regular kernel compilation flows.
2. Build a host application and link it to the libraries from Step 1.
Attention: To emulate multiple devices alongside other OpenCL SDKs, link your host application to the Khronos ICD Loader Library *before* linking it to the host runtime libraries. Link the host application to the ICD Loader Library by modifying the `Makefile` for the host application. Refer to *Linking Your Host Application to the Khronos ICD Loader Library* for more information.
3. If necessary, move the `<your_kernel_filename>.aocx` file to a location where the host can find easily, preferably the current working directory.
4. To run the host application for emulation:
 - For Windows, first define the number of emulated devices by invoking the `set CL_CONTEXT_EMULATOR_DEVICE_INTELFPGA=<number_of_devices>` command and then run the host application.

After you run the host application, invoke `set CL_CONTEXT_EMULATOR_DEVICE_INTELFPGA=` to unset the variable.
 - For Linux, invoke the `env`
`CL_CONTEXT_EMULATOR_DEVICE_INTELFPGA=<number_of_devices>`
`<host_application_filename>` command.

This command specifies the number of identical emulation devices that the Emulator needs to provide.

Remember: When the environment variable `CL_CONTEXT_EMULATOR_DEVICE_INTELFPGA` is set, only the emulated devices are available, i.e., access to all physical boards is disabled.

5. If you change your host or kernel program and you want to test it, only recompile the modified host or kernel program and then rerun emulation.

Each invocation of the emulated kernel creates a shared library copy called `<process_ID>-libkernel.so` in a default temporary directory, where `<process_ID>` is a unique numerical value assigned to each emulation run. You may override the default directory by setting the `TMP` or `TEMP` environment variable on Windows, or setting `TMPDIR` on Linux.

3.9. Creating the Hardware Configuration File of an OpenCL Kernel for SoC FPGA

The hardware configuration file of an OpenCL kernel is in the form of a `.aocx` executable file. To create the `.aocx` file for the `hello_world` example OpenCL application, you must first download and extract the example design from the OpenCL Design Examples page on the FPGA website. Then, compile the `hello_world.cl` kernel source file using the `aoc` command.

After you successfully install your FPGA board, you can create a `.aocx` file that executes on the device. Figure 3 on page 9 outlines the procedure for programming the `hello_world` design example onto the Cyclone V SoC FPGA. For more information on the OpenCL design examples, refer to the OpenCL Design Examples page.

Important: Before you program your Cyclone V SoC FPGA with the hardware image, ensure that your SoC FPGA contains an image created using the current version of the SDK.

1. Verify that you set the environment variable `AOCL_BOARD_PACKAGE_ROOT` to point to the Cyclone V SoC Development Kit Reference Platform (that is, `INTELFPGAOCCLSDKROOT/board/c5soc`).
The `init_openc1` script you ran when installing the Cyclone V SoC Development Kit should have set the `AOCL_BOARD_PACKAGE_ROOT` environment variable. If the returned path does not include `INTELFPGAOCCLSDKROOT/board/c5soc`, modify the `AOCL_BOARD_PACKAGE_ROOT` setting.
2. Verify that you set the environment variable `QUARTUS_ROOTDIR_OVERRIDE` to point to the installation directory of the correct software.
3. At a command prompt, navigate to the `hello_world` design.
4. To list the SoC FPGA boards available in the Cyclone V SoC Development Kit Reference Platform, invoke the `aoc -list-boards` command.
You should see an output similar to the one below:

```
Board list:
c5soc
c5soc_sharedonly
```

5. To compile the kernel for your Cyclone V SoC Development Kit, invoke the following command:

```
aoc -board=c5soc device/hello_world.cl -o=bin/
hello_world.aocx
```

This command performs the following tasks:

- Generates the design project files from the OpenCL source code.
- Checks for initial syntax errors.
- Performs basic optimizations.
- Creates a `hello_world` subfolder or subdirectory containing necessary intermediate files.
- Creates the `.aoco` object file.
- Creates the `.aocx` hardware configuration file and saves it in the `bin` subfolder or subdirectory.

Attention: The `.aocx` file might take hours to build, depending on the complexity of the kernel. To view the progress of the compilation on-screen, include the `-v` flag in your `aoc` command (that is, `aoc -v <your_kernel_filename>.cl`).

The offline compiler displays the line `aoc: Hardware generation completed successfully.` to signify the completion of the compilation process.

For more information on the `-list-boards`, `-board=<board_name>`, `-v`, and `-o=<filename>` options of the `aoc` command, refer to the *Intel FPGA SDK for OpenCL Standard Edition Programming Guide*.

Related Information

- [OpenCL Design Examples](#)
- [Listing the Available FPGA Boards in Your Custom Platform \(-list-boards\)](#)
- [Compiling a Kernel for a Specific FPGA Board \(-board=<board_name>\)](#)
- [Generating Compilation Progress Report \(-v\)](#)
- [Specifying the Name of an Intel FPGA SDK Offline Compiler Output File \(-o=<filename>\)](#)
- [Setting the Intel FPGA SDK for OpenCL User Environment Variables for SoC FPGA on page 28](#)
- [Setting Environment Variables and Loading OpenCL Linux Kernel Driver on page 32](#)

3.10. Executing an OpenCL Kernel on an SoC FPGA

Build your host application using the GCC cross-compiler available with the SoC EDS.

Related Information

[Building the Host Application](#) on page 36

3.10.1. Running the Host Application

For Linux systems, execute the `hello_world.aocx` file on the SoC FPGA by running the host application you built from the ARM-specific `Makefile`.

1. Log into your SoC FPGA board.
2. Copy the `hello_world.aocx` hardware configuration file and the `hello_world` host executable from their current directories to the board.
3. Verify that the `LD_LIBRARY_PATH` environment variable setting includes `$INTELFPGAOCSDKROOT/host/arm32/lib`. Run the command `echo $LD_LIBRARY_PATH`.
If you ran the `init_opencl.sh` script, the `LD_LIBRARY_PATH` setting should point to `$INTELFPGAOCSDKROOT/host/arm32/lib`.
4. To execute the kernel on the SoC FPGA, at a command prompt, navigate to the host executable directory and run the `hello_world` host executable.

Related Information

- [Setting the Intel FPGA SDK for OpenCL User Environment Variables for SoC FPGA on page 28](#)
- [Setting Environment Variables and Loading OpenCL Linux Kernel Driver on page 32](#)

3.10.2. Output from Successful Kernel Execution on the Cyclone V SoC Development Kit

When you run the host application to execute your OpenCL kernel on the Cyclone V SoC Development Kit, the software notifies you of a successful kernel execution.

Example output:

```
Reprogramming device [0] with handle 1
Querying platform for info:
=====
CL_PLATFORM_NAME           = Intel(R) FPGA SDK for OpenCL(TM)
CL_PLATFORM_VENDOR         = Intel Corporation
CL_PLATFORM_VERSION        = OpenCL 1.0 Intel(R) FPGA SDK for OpenCL(TM), <version>

Querying device for info:
=====
CL_DEVICE_NAME              = c5soc : 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>
CL_DRIVER_VERSION           = <version>
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   = 2147483648
CL_DEVICE_IMAGE_SUPPORT     = false
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 = 3758096384
CL_DEVICE_MAX_WORK_ITEM_DIMENSIONS = 3
CL_DEVICE_MAX_WORK_ITEM_DIMENSIONS = 1024
CL_DEVICE_MIN_DATA_TYPE_ALIGN_SIZE = 128
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

Kernel initialization is complete.
Launching the kernel...

Thread #2: Hello from the Intel(R) FPGA SDK for OpenCL(TM) Compiler!

Kernel execution is complete.
```

3.11. Uninstalling the Software

To uninstall the Intel FPGA SDK for OpenCL Standard Edition for Linux, remove the software package via the GUI uninstaller, then delete the software directory and restore all modified environment variables to their previous settings.

1. Remove the software package by performing one of the following tasks:
 - a. To uninstall the SDK, run the `aocl-<version>-uninstall.run` program located in the `<install directory>/uninstall` directory of the Intel Quartus Prime Standard Edition software.
2. Remove `$INTELFPGAOCCLSDKROOT/bin` from the `PATH` environment variable.
3. Remove `$INTELFPGAOCCLSDKROOT/host/linux64/lib` from the `LD_LIBRARY_PATH` environment variable.
4. Remove the `INTELFPGAOCCLSDKROOT` environment variable.
5. Remove the `QUARTUS_ROOTDIR_OVERRIDE` environment variable.

A. Document Revision History of the Intel FPGA SDK for OpenCL Standard Edition Cyclone V SoC Getting Started Guide

Document Version	Intel Quartus Prime Version	Changes
2018.09.24	18.1	Maintenance release.
2018.05.04	18.0	Maintenance release.

Date	Version	Changes
November 2017	2017.11.04	<ul style="list-style-type: none"> Rebranded the environment variable ALTERAOCLSDKROOT to INTELFPGAOCSDKROOT. Rebranded CL_CONTEXT_EMULATOR_DEVICE_ALTERA to CL_CONTEXT_EMULATOR_DEVICE_ALTERA. Removed topics on Licensing the Intel FPGA SDK for OpenCL in both SoC for Linux and SoC for Windows section. In the following topics, implemented the single dash and -option=<value> conventions: <ul style="list-style-type: none"> Overview of the Intel FPGA SDK for OpenCL Cyclone V SoC Programming Flow on page 8 Cyclone V SoC Development Kit Reference Platform Board Variants on page 9 Creating the Hardware Configuration File of an OpenCL Kernel for SoC FPGA on page 19 Compiling a Kernel for Emulation (-march=emulator) on page 35 In Contents of the Intel FPGA SDK for OpenCL Standard Edition on page 5: <ul style="list-style-type: none"> In the sentence "The Intel FPGA SDK for OpenCL provides logic components, drivers, and SDK-specific libraries and files.", changed logical components to programs. Under the Logic Components section, changed "host platform API and runtime API" as "host and runtime API". Removed topics on Licensing the Intel FPGA SDK for OpenCL in both OpenCL for Linux and Windows sections since it no longer is required.
May 2017	2017.05.08	<ul style="list-style-type: none"> Updated download instructions
October 2016	2016.10.31	<ul style="list-style-type: none"> Rebranded the Altera SDK for OpenCL to Intel FPGA SDK for OpenCL.
May 2016	2016.05.02	<ul style="list-style-type: none"> Added the %ALTERAOCLSDKROOT%\windows64\bin setting to the list of Windows environment variables.
November 2015	2015.11.02	<ul style="list-style-type: none"> Changed instances of <i>Quartus II</i> to <i>Quartus Prime</i>. Added Windows 8.1 to supported Windows versions. Updated download and installation instructions for the tar file that includes the AOCL, the Intel Quartus Prime software, and device support.
continued...		

Date	Version	Changes
		<ul style="list-style-type: none"> Added emulation as a optional step in the AOCL start up process on Linux. Updated the figure <i>Cyclone V SoC Development Kit Programming Overview</i> to include steps for checking <code>QUARTUS_ROOTDIR_OVERRIDE</code> setting and an optional emulation step. Updated software uninstallation instructions.
May 2015	15.0.0	<ul style="list-style-type: none"> Organized instructions into the following sections: <ul style="list-style-type: none"> Setting Up the Altera SDK for OpenCL on SoC for Windows Setting Up the Altera SDK for OpenCL on SoC for Linux
December 2014	14.1.0	<ul style="list-style-type: none"> Reorganized information flow. Updated Red Hat Enterprise Linux (RHEL) version support. Included information on the <code>ALTERAOCLSDKROOT/init_openc1</code> script for setting environment variables. Added the <i>Contents of the AOCL</i> section. Consolidated software uninstallation instructions and moved them to the end of the document. Updated the kernel compilation command in the figure <i>Cyclone V SoC Development Kit Programming Overview</i>. Updated licensing instructions for the new Altera® Software Development Kit (SDK) for OpenCL (AOCL) license. Updated instructions for building and running the host application for SoC.
June 2014	14.0.0	<ul style="list-style-type: none"> Initial Release.