## Contents

### Altera RTE for OpenCL Getting Started Guide ......................................................... 1-1
- Prerequisites for the Altera RTE for OpenCL.............................................................. 1-2
- Contents of the Altera RTE for OpenCL......................................................................... 1-2
- RTE Utility.................................................................................................................. 1-4
  - Displaying the Software Version.............................................................................. 1-5
  - Listing the RTE Utility Command Options............................................................. 1-5
- Managing an FPGA Board.......................................................................................... 1-5
- Managing Host Application......................................................................................... 1-5

### Getting Started with the Altera RTE for OpenCL for 64-Bit Windows............. 2-1
- Downloading the Altera RTE for OpenCL................................................................. 2-1
- Installing the Altera RTE for OpenCL......................................................................... 2-2
- Setting the Environment Variables for Windows....................................................... 2-2
- Verifying the RTE Installation.................................................................................... 2-3
- Licensing the Software............................................................................................... 2-4
- Installing an FPGA Board......................................................................................... 2-5
- Updating the Hardware Image on the FPGA.............................................................. 2-6
  - Querying the Device Name of Your FPGA Board.................................................. 2-6
  - Programming the Flash Memory of an FPGA on Windows..................................... 2-7
- Executing an OpenCL Kernel on an FPGA............................................................... 2-8
  - Building the Host Application............................................................................... 2-8
  - Running the Host Application............................................................................... 2-8
  - Output from Successful Kernel Execution.............................................................. 2-8
- Uninstalling the Software......................................................................................... 2-9
- Uninstalling the FPGA Board................................................................................... 2-10

### Getting Started with the Altera RTE for OpenCL for x86_64 Linux and 64-Bit Linux Big-Endian System ........................................................................................... 3-1
- Downloading the Altera RTE for OpenCL................................................................. 3-1
- Installing the Altera RTE for OpenCL......................................................................... 3-2
- Setting the Environment Variables for Linux............................................................ 3-2
- Installing the RTE on Big-Endian Systems............................................................... 3-3
  - Setting the Environment Variables on Big-Endian Systems.................................... 3-4
- Verifying the RTE Installation.................................................................................... 3-4
- Licensing the Software............................................................................................... 3-5
- Installing an FPGA Board......................................................................................... 3-6
- Updating the Hardware Image on the FPGA.............................................................. 3-7
  - Querying the Device Name of Your FPGA Board.................................................. 3-7
  - Programming the Flash Memory of an FPGA on Linux.......................................... 3-7
- Executing an OpenCL Kernel on an FPGA............................................................... 3-8
  - Building the Host Application............................................................................... 3-9
Getting Started with the Altera RTE for OpenCL for Altera ARMv7-A SoC...

4-1

Getting Started with the Altera RTE for OpenCL for SoC on Windows................................. 4-1
  Downloading the Altera SDK for OpenCL and the SoC EDS........................................ 4-2
  Installing the Altera SDK for OpenCL.......................................................................... 4-2
  Installing the SoC EDS ................................................................................................. 4-3
  Licensing the Software................................................................................................. 4-3
  Recompiling the Linux Kernel Driver........................................................................... 4-4
  Installing the RTE onto the SoC Board........................................................................ 4-4
  Installing the Cyclone V SoC Development Kit.......................................................... 4-5
  Executing an OpenCL Kernel on an SoC...................................................................... 4-10

Getting Started with the Altera RTE for OpenCL for SoC on Linux........................................ 4-12
  Downloading the Altera SDK for OpenCL and the SoC EDS........................................ 4-12
  Installing the Altera SDK for OpenCL.......................................................................... 4-13
  Installing the SoC EDS ................................................................................................. 4-14
  Licensing the Software................................................................................................. 4-14
  Recompiling the Linux Kernel Driver........................................................................... 4-15
  Installing the RTE onto the SoC Board........................................................................ 4-15
  Installing the Cyclone V SoC Development Kit.......................................................... 4-16
  Executing an OpenCL Kernel on an SoC...................................................................... 4-21

Uninstalling the Altera RTE for OpenCL.............................................................................. 4-23

Document Revision History.................................................................................................. A-1
The *Altera RTE for OpenCL Getting Started Guide* describes the procedures you follow to install the Altera® Runtime Environment (RTE) for OpenCL™. This document also contains instructions on how to deploy an OpenCL (1) application with the RTE.

The RTE is a subset of the Altera Software Development Kit (SDK) for OpenCL(2) (AOCL). Unlike the AOCL, which provides an environment that enables the development and deployment of OpenCL kernel programs, the RTE provides tools and runtime components that enable you to build and execute a host program, and execute precompiled OpenCL kernel programs on target accelerator boards.

OpenCL is a C-based open standard for the programming of heterogeneous parallel devices. For more information on the OpenCL Specification version 1.0, refer to the OpenCL Reference Pages. For detailed information on the OpenCL application programming interface (API) and programming language, refer to the *OpenCL Specification version 1.0*.

**Attention:** If you require OpenCL kernel development and deployment functionalities, download the AOCL. Refer to the *Altera SDK for OpenCL Getting Started Guide* for more information.

Do not install the RTE and the AOCL on the same host system.

**Attention:** If you require OpenCL kernel development and deployment functionalities that target the Cyclone® V SoC Development Kit, refer to the *Altera SDK for OpenCL Cyclone V SoC Getting Started Guide* for more information.

**Related Information**
- OpenCL Reference Pages
- OpenCL Specification version 1.0
- Altera SDK for OpenCL Getting Started Guide
- Altera SDK for OpenCL Cyclone V SoC Getting Started Guide

---

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

(2) The Altera 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](http://www.khronos.org/conformance).
Prerequisites for the Altera RTE for OpenCL

To install the RTE and deploy an OpenCL application on an Altera preferred accelerator board, your system must meet certain hardware, target platform, and software requirements.

Hardware Requirements

1. Accelerator boards requirements:
   - Acquire a Reference Platform from Altera, or a Custom Platform from an Altera preferred board vendor.
   For more information, refer to the OpenCL Reference Platforms page on the Altera website.

2. The host system must be running one of the following supported target platforms:
   - Microsoft 64-bit Windows 7 on the x86-64 architecture.
   - Red Hat Enterprise 64-bit Linux (RHEL) version 6 on the x86-64 architecture.
   - RHEL 6 on big-endian system
   - Linux versions as supported on Altera SoCs on the ARM® ARMv7-A architecture

   **Important:** For x86_64 Linux systems, 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`).

   To install the Linux kernel source or header package, invoke the `yum install <kernel_package_name>` command.

   You must have administrator privileges on the host system to install the necessary packages and drivers.

Software Prerequisites

1. Develop your host application using one of the following RTE-compatible C compiler or software development environment:
   - For Windows systems, use Microsoft Visual Studio version 2010 Professional.
   - For Linux systems, use the C compiler included with the GCC.

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

Related Information

*OpenCL Platforms page on the Altera website*

Contents of the Altera RTE for OpenCL

The Altera RTE for OpenCL provides utilities, host runtime libraries, drivers, and RTE-specific libraries and files.
Utilities and Host Runtime Libraries

- The RTE Utility includes commands you can invoke to perform high-level tasks. The RTE utilities are a subset of the Altera SDK for OpenCL (AOCL) utilities.
- The host runtime provides the OpenCL host platform application program interface (API) and runtime API for your OpenCL host application.

The host runtime consists of the following libraries:

- *Statically-linked libraries* provide OpenCL host APIs, hardware abstractions and helper libraries.
- *Dynamically-linked libraries* (DLLs) provide hardware abstractions and helper libraries.

Drivers, Libraries and Files

The RTE installation process installs the RTE into a directory that you own. The path to the software installation directory is referenced by the `ALTERAOCLSDKROOT` environment variable. The following table highlights the contents of this directory:

<table>
<thead>
<tr>
<th>Windows Folder</th>
<th>Linux Directory</th>
<th>Big-Endian System Directory</th>
<th>ARM Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bin</td>
<td>bin</td>
<td>bin</td>
<td>bin</td>
<td>High-level utilities. Include this directory in your <code>PATH</code> environment variable setting.</td>
</tr>
<tr>
<td>board</td>
<td>board</td>
<td>—</td>
<td>board</td>
<td>The Reference Platform available with the RTE. Important: The Reference Platform for the RTE does not include the <code>hardware</code> subdirectory.</td>
</tr>
<tr>
<td>host</td>
<td>host</td>
<td>host</td>
<td>host</td>
<td>Files necessary for compiling your host program.</td>
</tr>
<tr>
<td>host\include</td>
<td>host/ include</td>
<td>host/ include</td>
<td>host/include</td>
<td>OpenCL Specification version 1.0 header files and software interface files necessary for compiling and linking your host application.</td>
</tr>
</tbody>
</table>

The `host/ include/CL` subdirectory also includes the C++ header file `cl.hpp`. 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.

Important: The OpenCL version 1.1 C++ bindings are compatible with OpenCL Specification versions 1.0 and 1.1.

Add this path to the `include` file search path in your development environment.
Example OpenCL Applications

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

Related Information
OpenCL Design Examples page on the Altera website

RTE Utility

The Altera RTE for OpenCL utility is a subset of the Altera SDK for OpenCL utility. It provides you with tools and information to perform high-level tasks such as configuring the host application development flow.

**Displaying the Software Version** on page 1-5
To display the version of the Altera RTE for OpenCL, invoke the `version` utility command.

**Listing the RTE Utility Command Options** on page 1-5
To display information on the Altera RTE for OpenCL utility command options, invoke the `help` utility command.

**Managing an FPGA Board** on page 1-5
The Altera RTE for OpenCL includes utility commands you can invoke to install, uninstall, diagnose, and program your FPGA board.

**Managing Host Application** on page 1-5
The Altera RTE for OpenCL includes utility commands you can invoke to obtain information on flags and libraries necessary for compiling and linking your host application.
Displaying the Software Version

To display the version of the Altera RTE for OpenCL, invoke the `version` utility command.

**Note:** The ARM processor on the Cyclone V SoC Development Kit does not support this utility.

- At the command prompt, invoke the `aocl version` command.
  
  Example output:

  ```
  aocl <version> <build> (Altera Runtime Environment for OpenCL, Version <version> Build <build>, Copyright (C) <year> Altera Corporation)
  ```

Listing the RTE Utility Command Options

To display information on the Altera RTE for OpenCL utility command options, invoke the `help` utility command.

**Attention:** The ARM processor on the Cyclone V SoC Development Kit does not support this utility.

- At a command prompt, invoke the `aocl help` command.
  
  The RTE categorizes the utility command options based on their functions. It also provides a description for each option.

Managing an FPGA Board

The Altera RTE for OpenCL includes utility commands you can invoke to install, uninstall, diagnose, and program your FPGA board.

For more information on the `install`, `uninstall`, `diagnose`, `program` and `flash` utility commands, refer to the *Managing an FPGA Board* section of the *Altera SDK for OpenCL Programming Guide*.

**Related Information**

- *Managing an FPGA Board*

Managing Host Application

The Altera RTE for OpenCL includes utility commands you can invoke to obtain information on flags and libraries necessary for compiling and linking your host application.

**Attention:** To cross-compile your host application to an SoC board, include the `--arm` option in your utility command.

**Caution:** For Linux systems, if you debug your host application using the GNU Project Debugger (GDB), invoke the following command prior to running the host application:

```
handle SIG44 nostop
```

Without this command, the GDB debugging process terminates with the following error message:

```
Program received signal SIG44, Real-time event 44.
```
For information on the following utility command options, refer to the Managing Host Application section of the Altera SDK for OpenCL Programming Guide:

- example-makefile or makefile
- compile-config
- ldflags
- ldlibs
- link-config or linkflags

Related Information
Managing Host Application
Getting Started with the Altera RTE for OpenCL
for 64-Bit Windows

1. Downloading the Altera RTE for OpenCL on page 2-1
   Download the RTE for Windows from the Download Center on the Altera website.

2. Installing the Altera RTE for OpenCL on page 2-2
   Install the RTE in a folder that you own.

3. Setting the Environment Variables for Windows on page 2-2
   You have the option to set the RTE environment variables permanently or transiently.

4. Verifying the RTE Installation on page 2-3
   Invoke the version utility command and verify that the correct version of the AOCL is installed.

5. Licensing the Software on page 2-4
   Obtain the AOCL license from the Altera Self Service Licensing Center.

6. Installing an FPGA Board on page 2-5
   To install your board into the host system, invoke the install utility command.

7. Updating the Hardware Image on the FPGA on page 2-6
   If applicable, before you execute an OpenCL kernel program on the FPGA, ensure that the flash memory of the FPGA contains a hardware image created using a current version of the OpenCL software.

8. Executing an OpenCL Kernel on an FPGA on page 2-8
   Build your OpenCL host application in Microsoft Visual Studio version 2010 Professional, and run the application by invoking the hello_world.exe executable.

9. Uninstalling the Software on page 2-9
   To uninstall the RTE on Windows, delete the RTE folder and restore all modified environment variables to their previous settings.

10. Uninstalling the FPGA Board on page 2-10
    To uninstall an FPGA board for Windows, invoke the uninstall utility command, uninstall the Custom Platform, and unset the relevant environment variables.

---

**Downloading the Altera RTE for OpenCL**

Download the RTE for Windows from the Download Center on the Altera website.

1. In the main page of the Altera website, click MYALTERA and log into your account.
If you do not have a myAltera account, register for a new account.

2. Click **DOWNLOADS** to enter the Download Center.

3. Click **Altera SDK for OpenCL** to enter the download page for the subscription edition of the AOCL.

4. Select the software version. The default selection is the current version.

5. Select **Direct Download** as the download method.

6. Click the RTE tab and select **Altera Runtime Environment for OpenCL Windows x86-64**. Click **More** beside **Download and install instructions** to view the download and installation procedure.

7. Perform the steps outlined in the download and installation instructions on the download page.

Related Information

**Altera website**

## Installing the Altera RTE for OpenCL

Install the RTE in a folder that you own.

### Before you begin

You must have administrator privileges.

- To install the Altera Runtime Environment (RTE) for OpenCL as a stand-alone software, perform the following tasks:
  1. Run the **AOCLRTESetup-<version>-win64.exe** installer. Direct the installer to extract the software to an empty folder that you own (that is, not a system folder).

     The installation path must not contain any spaces (for example, `<home_directory>\altera\<version>\aclrt-windows64`).

     2. The installer sets the environment variable **ALTERAOCLSDKROOT**. Ensure that **ALTERAOCLSDKROOT** points to the current version of the software.

### Setting the Environment Variables for Windows

You have the option to set the RTE environment variables permanently or transiently. The environment variable settings describe the FPGA board and the host runtime to the software.
Attention: If you set the environment variables permanently, you apply the settings once during installation. If you set the environment variables transiently, you must apply the settings during installation and during every subsequent session you run.

- To apply permanent environment variable settings, perform the following tasks:
  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:

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Path to Include</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATH</td>
<td>1. %ALTERAOCLSDKROOT%\bin</td>
</tr>
<tr>
<td></td>
<td>2. %ALTERAOCLSDKROOT%\host\windows64\bin</td>
</tr>
</tbody>
</table>

where ALTERAOCLSDKROOT points to the path of the software installation.

- To apply transient environment variable settings, open a command window and run the %ALTERAOCLSDKROOT%\init_opencl.bat script.

Example script output:

```
AOCL_BOARD_PACKAGE_ROOT path is not set in environment
Setting to default s5_ref board.
If you want to target another board, do
  set AOCL_BOARD_PACKAGE_ROOT=board_pkg_dir
  and re-run this script
Adding %ALTERAOCLSDKROOT%\bin to PATH
Adding %ALTERAOCLSDKROOT%\host\windows64\bin to PATH
Adding %AOCL_BOARD_PACKAGE_ROOT%\windows64\bin to PATH
```

where AOCL_BOARD_PACKAGE_ROOT points to the path of the Custom or Reference Platform.

Running the init_opencl.bat script only affects the current command window. The script performs the following tasks:

- Finds the Microsoft Visual Studio installation
- Imports the Microsoft Visual Studio environment to properly set the LIB environment variable
- Ensures that the PATH environment variable includes the path to the Microsoft LINK.EXE file

Verifying the RTE Installation

Invoke the version utility command and verify that the correct version of the AOCL is installed.
Attention: The ARM processor on the Cyclone V SoC Development Kit does not support the AOCL version utility.

- At a command prompt, invoke the `aocl version` utility command.
  An output similar to the one below notifies you of a successful installation:

  ```
aocl <version>.<build> (Altera Runtime Environment for OpenCL, Version <version>
Build <build>, Copyright (C) <year> Altera Corporation)
```

- If installation was unsuccessful, reinstall the software. You can also refer to the Altera Software Installation and Licensing manual and the Knowledge Center on the Altera website for more information.

Related Information
- Altera Software Installation and Licensing
- Altera Knowledge Base

Licensing the Software

The RTE shares the same license as the AOCL. Obtain the AOCL license from the Altera Self Service Licensing Center.

Before you begin

For information on the licensing options and requirements, refer to the Licensing Altera Software section of the Altera Software Installation and Licensing manual, the LICENSE.txt file that accompanies each software, and the Altera Licensing page on the Altera website.

1. In the main page on the Altera website, click MYALTERA and log into your account.
2. Click the link to the Self-Service Licensing Center.
3. Perform the steps outlined in the Using the Self-Service Licensing Center section of the Altera Software Installation and Licensing manual to obtain and activate the license.
4. If you have a fixed license, append the `<path_to_license_file>/license_filename>` file to the `LM_LICENSE_FILE` environment variable in the following manner:
   a. Create a backup copy of the provided license file.
   b. Save the new license file on your local hard drive.
   c. Append the `<path_to_license_file>/license_filename>` file to the `LM_LICENSE_FILE` environment variable.
5. If you have a floating licence, append the `<path_to_license_file>/license_filename>` file to the `LM_LICENSE_FILE` environment variable in the following manner:
   a. Obtain the port number and host name from the network or system administrator. Alternatively, the information is in the license file line `SERVER <hostname> <8 to 12 character host or NIC ID> <port>`. The license location for the user is `<port>@<hostname>`. If a port is not listed in the license file, specify `@<hostname>`.
   b. Modify the license file to update the port number and host name.
   c. Append the `<path_to_license_file>/license_filename>` file to the `LM_LICENSE_FILE` environment variable setting.
Installing an FPGA Board

Before creating an OpenCL application for an FPGA board, you must first download and install the Custom Platform from your board vendor. Most Custom Platform installers require administrator privileges. To install your board into the host system, invoke the `install` utility command.

The steps below outline the board installation procedure. Some Custom Platforms require additional installation tasks. Consult your board vendor's documentation for further information on board installation.

**Attention:** If you are installing the Cyclone V SoC Development Kit for use with the Cyclone V SoC Development Kit Reference Platform, refer to *Installing the Cyclone V SoC Development Kit* in the *Altera SDK for OpenCL Cyclone V SoC Getting Started Guide* for more information.

1. Follow your board vendor’s instructions to connect the FPGA board to your system.
2. Download the Custom Platform for your FPGA board from your board vendor’s website.
   For more information, refer to the OpenCL Reference Platforms page on the Altera website.
3. Install the Custom Platform in a folder that you own (that is, not a system folder).
4. Set the environment variable `AOCL_BOARD_PACKAGE_ROOT` to point to the location of the Custom Platform subfolder containing the `board_env.xml` file.
   For example, for the Stratix® V Network Reference Platform (s5_net), set `AOCL_BOARD_PACKAGE_ROOT` to point to the `<path_to_s5_net>\s5_net` folder.
5. Add the Custom Platform library paths to the `PATH` environment variable setting. You may apply permanent settings manually by adding the path to the memory-mapped (MMD) library within the Custom Platform. Alternatively, you may apply transient settings to the current session by running the `%ALTERAOCLSDKROOT%\init_opencl.bat` script.
   For example, if you use s5_net, the Windows `PATH` environment variable setting is `%AOCL_BOARD_PACKAGE_ROOT%\windows64\bin`.
   For information on `init_opencl.bat`, refer to the *Setting the Environment Variables for Windows* section.
6. Invoke the command `aocl install` at a command prompt.
   Invoking `aocl install` also installs a board driver that allows communication between host applications and hardware kernel programs.
7. To query a list of FPGA devices installed in your machine, invoke the `aocl diagnose` command.
   The software generates an output that includes the `<device_name>`, which is an acl number that ranges from acl0 to acl15.
   For more information on querying the `<device_name>` of your accelerator board, refer to the *Querying the Device Name of Your FPGA Board* section.
8. To verify the successful installation of the FPGA board, invoke the command `aocl diagnose <device_name>` to run any board vendor-recommended diagnostic test.

**Related Information**
- Setting the Environment Variables for Windows on page 2-2
- Querying the Device Name of Your FPGA Board on page 2-6
- Installing the Cyclone V SoC Development Kit on page 4-5

**Updating the Hardware Image on the FPGA**

If applicable, before you execute an OpenCL kernel program on the FPGA, ensure that the flash memory of the FPGA contains a hardware image created using a current version of the OpenCL software.

**Remember:** If your Custom Platform requires that you preload a valid OpenCL image into the flash memory, for every major release of the Altera Complete Design Suite, program the flash memory of the FPGA with a hardware image compatible with the current version of the software.

**Querying the Device Name of Your FPGA Board**

Some AOCL utility commands require you to specify the device name (`<device_name>`). The `<device_name>` refers to the acl number (e.g. acl0 to acl15) that corresponds to the FPGA device. When you query a list of accelerator boards, the RTE produces a list of installed devices on your machine in the order of their device names.

- To query a list of installed devices on your machine, type `aocl diagnose <device_name>` at a command prompt. The software generates an output that resembles the example shown below:

```
aocl diagnose: Running diagnostic from ALTERAOCLSDKROOT/board/<board_name>/
<platform>/libexec

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

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

<table>
<thead>
<tr>
<th>device_name</th>
<th>Status</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl0</td>
<td>Passed</td>
<td>PCIe dev_id = &lt;device_ID&gt;, bus:slot.func = 02:00.00, at Gen 2 with 8 lanes. FPGA temperature = 43.0 degrees C.</td>
</tr>
<tr>
<td>acl1</td>
<td>Passed</td>
<td>PCIe dev_id = &lt;device_ID&gt;, bus:slot.func = 03:00.00, at Gen 2 with 8 lanes. FPGA temperature = 35.0 degrees C.</td>
</tr>
</tbody>
</table>

Found 2 active device(s) installed on the host machine, to perform a full diagnostic on a specific device, please run aocl diagnose <device_name>
```

DIAGNOSTIC_PASSED
Programming the Flash Memory of an FPGA on Windows

By default, you configure an FPGA using the hardware image stored in the flash memory of the device. When there is no power, the FPGA retains the hardware configuration file in the flash memory. When you power up the system, it configures the FPGA circuitry based on this hardware image in the flash memory. Therefore, it is imperative that an OpenCL-compatible hardware configuration file is loaded into the flash memory of your FPGA.

Preloading an OpenCL image into the flash memory is necessary for the proper functioning of many Custom Platforms. For example, most PCI Express® (PCIe®)-based boards require a valid OpenCL image in flash memory so that hardware on the board can use the image to configure the FPGA device when the host system powers up for the first time. If the FPGA is not configured with a valid OpenCL image, the system will fail to enumerate the PCIe endpoint, or the driver will not function.

Before running any designs, ensure that the flash memory of your board has a valid OpenCL image that is compatible with the current OpenCL software version. Consult your board vendor’s documentation for board-specific requirements.

Caution: When you load the hardware configuration file into the flash memory of the FPGA, maintain system power for the entire loading process, which might take a few minutes. Also, do not launch any host code that calls OpenCL kernels or might otherwise communicate with the FPGA board.

To load your hardware configuration file into the flash memory of your FPGA board, perform the following tasks:

1. Install any drivers or utilities that your Custom Platform requires.
   For example, some Custom Platforms require you to install the USB-Blaster™ driver to load your hardware configuration file into the flash memory. For installation instructions, refer to the USB-Blaster II Download Cable User Guide.

2. Ensure that you set the AOCL_BOARD_PACKAGE_ROOT environment variable to point to the subfolder in your Custom Platform that contains the board_env.xml file.

3. Download an example design for your Custom Platform.
   Remember: You can download example designs from the OpenCL Design Examples page, and extract the example to a location to which you have write access.

4. To load the hardware configuration file into the flash memory, invoke the aocl flash <device_name> <example_design_filename>.aocx command, where <device_name> refers to the acl number (e.g. acl0 to acl15) that corresponds to your FPGA device, and <example_design_filename>.aocx is the precompiled hardware configuration file you can find in the example design package.

5. Power down your device or computer and then power it up again.
   Power cycling ensures that the FPGA configuration device retrieves the hardware configuration file from the flash memory and configures it into the FPGA.

Warning: Some Custom Platforms require you to power cycle the entire host system after programming the flash memory. For example, PCIe-based Custom Platforms might require a host system reboot to reenumerate the PCIe endpoint. Altera recommends that you power cycle the complete host system after programming the flash memory.

Related Information
- USB-Blaster II Download Cable User Guide
Executing an OpenCL Kernel on an FPGA

Build your OpenCL host application in Microsoft Visual Studio version 2010 Professional, and run the application by invoking the `hello_world.exe` executable. The AOCL is compatible with 64-bit host binaries only.

Building the Host Application

The `<path_to_exm_opencl_hello_world_x64_windows_<version>>\hello_world\hello_world.sln` file contains the host solution. After you open this `.sln` file in Microsoft Visual Studio, you can build the host application in the `main.cpp` file.

To build the host application, perform the following tasks:

1. Open the `<path_to_exm_opencl_hello_world_x64_windows_<version>>\hello_world\hello_world.sln` file in Microsoft Visual Studio.
2. Verify that the build configuration is correct. The default build configuration is Debug, but you can use Release. You must select the appropriate option as the solution platform (for example, for x64 architecture, select x64).
3. Build the solution by selecting the Build > Build Solution menu option, or by pressing the F7 key. The `hello_world.exe` executable will be in the `<path_to_exm_opencl_hello_world_x64_windows_<version>>\hello_world\bin` folder.
4. Verify that the build is correct. An output ending with a message similar to the one shown below notifies you of a successful build:

```
1> Build succeeded.
1>
1> Time Elapsed 00:00:03:29
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

**Attention:** You can ignore the LNK4009: PDB 'vc90.pdb' was not found with... warnings because they have no effect on the build. The compiler might issue this type of warning messages if you have built your Windows libraries using a previous version of Microsoft Visual Studio.

Running the Host Application

To execute the OpenCL kernel on the FPGA, run the Windows host application that you built from the `.sln` file.

1. Add the path `%ALTERAOCLSDKROOT%\host\windows64\bin` to the PATH environment variable.
2. At a command prompt, navigate to the host executable within the `<path_to_exm_opencl_hello_world_x64_windows_<version>>\hello_world\bin` folder.
3. Invoke the `hello_world.exe` executable.

The `hello_world` executable executes the kernel code on the FPGA.

Output from Successful Kernel Execution

When you run the host application to execute your OpenCL kernel on the target FPGA, the RTE notifies you of a successful kernel execution.
Example output:

Found 1 OpenCL platforms.

Querying platform for info:
=================================
CL_PLATFORM_NAME = Altera SDK for OpenCL
CL_PLATFORM_VENDOR = Altera Corporation
CL_PLATFORM_VERSION = OpenCL 1.0 Altera SDK for OpenCL, Version <version>

Querying device for info:
=================================
CL_DEVICE_NAME = <board_name>: <descriptive_board_name>
CL_DEVICE_VENDOR = <board_vendor_name>
CL_DEVICE_VENDOR_ID = <board_vendor_ID>
CL_DEVICE_VERSION = OpenCL 1.0 Altera SDK for OpenCL, Version <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 = 0
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 = 1073741824
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 Altera's OpenCL Compiler!

Kernel execution is complete.

Uninstalling the Software

To uninstall the RTE on Windows, delete the RTE folder and restore all modified environment variables to their previous settings.

1. In Windows Explorer, navigate to the altera\<version> folder.
2. Delete the aclrte-windows64 folder.
3. Remove the following paths from the PATH environment variable:
   a. %ALTERAOCLSDKROOT%\bin
   b. %ALTERAOCLSDKROOT%\host\windows64\bin
4. Remove the ALTERAOCLSDKROOT environment variable.
Uninstalling the FPGA Board

To uninstall an FPGA board for Windows, invoke the `uninstall` utility command, uninstall the Custom Platform, and unset the relevant environment variables. You must uninstall the existing FPGA board if you migrate your OpenCL application to another FPGA board from a different Custom Platform.

To uninstall your FPGA board, perform the following tasks:

1. Following your board vendor's instructions to disconnect the board from your machine.
2. Invoke the `aocl uninstall` utility command to remove the current host computer drivers (for example, PCI Express (PCIe) drivers). The RTE uses these drivers to communicate with the FPGA board.
3. Uninstall the Custom Platform.
4. Unset the `PATH` environment variable.
5. Unset the `AOCL_BOARD_PACKAGE_ROOT` environment variable.
1. **Downloading the Altera RTE for OpenCL** on page 3-1
   Download the RTE for Linux from the Download Center on the Altera website.

2. **Installing the Altera RTE for OpenCL** on page 3-2
   Install the RTE in a directory that you own.

3. **Installing the RTE on Big-Endian Systems** on page 3-3
   Install the RTE in a directory that you own.

4. **Verifying the RTE Installation** on page 3-4
   Invoke the `version` utility command and verify that the correct version of the AOCL is installed.

5. **Licensing the Software** on page 3-5
   Obtain the AOCL license from the Altera Self Service Licensing Center.

6. **Installing an FPGA Board** on page 3-6
   To install your board into the host system, invoke the `install` utility command.

7. **Updating the Hardware Image on the FPGA** on page 3-7
   If applicable, before you execute an OpenCL kernel program on the FPGA, ensure that the flash memory of the FPGA contains a hardware image created using a current version of the OpenCL software.

8. **Executing an OpenCL Kernel on an FPGA** on page 3-8
   You must build your OpenCL host application with the `Makefile` file, and run the application by invoking the `hello_world` executable.

9. **Uninstalling the Software** on page 3-10
   To uninstall the RTE for Linux, remove the software package via the Red Hat Package Manager (RPM) or GUI uninstaller, then delete the software directory and restore all modified environment variables to their previous settings.

10. **Uninstalling the FPGA Board** on page 3-10
    To uninstall an FPGA board for Linux, invoke the `uninstall` utility command, uninstall the Custom Platform, and unset the relevant environment variables.

---

**Downloading the Altera RTE for OpenCL**

Download the RTE for Linux from the Download Center on the Altera website.

1. In the main page of the Altera website, click **MYALTERA** and log into your account.
If you do not have a myAltera account, register for a new account.

2. Click DOWNLOADS to enter the Download Center.

3. Click Altera SDK for OpenCL to enter the download page for the subscription edition of the AOCL.

4. Select the software version. The default selection is the current version.

5. Select Direct Download as the download method.

6. Click the RTE tab and select the installation package you want to download. Click More beside Download and install instructions to view the download and installation procedure.

   - For x86-64 Linux Red Hat Enterprise (RPM) installation package, select Altera Runtime Environment for OpenCL Linux x86-64 RPM
   - For big-endian systems, select Altera Runtime Environment for OpenCL PowerPC RPM

7. Perform the steps outlined in the download and installation instructions on the download page.

Related Information
Altera website

Installing the Altera RTE for OpenCL

Install the RTE in a directory that you own.

Before you begin

1. You must have sudo or root privileges.

2. 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).

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 ALTERAOCLSDKROOT 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 RTE as a stand-alone software, perform the following tasks:

     1. To install the software using the Red Hat Package Manager (RPM), at the command prompt, invoke one of the following commands:

        - At a command prompt, type `rpm -i aclr-rte-<version>.x86_64.rpm` to install the software in the default location (for example, opt/altera/aclrte-linux64).
        - To install the software in the default location with verbose progress reporting, type `rpm -ivh aclr-rte-<version>.x86_64.rpm`
        - To install the software in an alternate directory that you own (that is, not a system directory), type the `rpm -i --prefix <rte_destination_directory> aclr-rte-<version>.x86_64.rpm` command.

     2. The installer sets the environment variable ALTERAOCLSDKROOT to the path of the software installation. Ensure that ALTERAOCLSDKROOT points to the current version of the software.

Setting the Environment Variables for Linux

You have the option to set the RTE environment variables permanently or transiently. The environment variable settings describe the FPGA board and the host runtime to the software.
Attention: If you set the environment variables permanently, you apply the settings once during installa-
tion. If you set the environment variables transiently, you must apply the settings during
installation and during every subsequent session you run.

- To apply permanent environment variable settings, manually include the following paths in the
corresponding environment variable settings:

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Path to Include</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATH</td>
<td>$ALTERAOCLSDKROOT/bin</td>
</tr>
<tr>
<td></td>
<td>where $ALTERAOCLSDKROOT points to the path of the software installation</td>
</tr>
<tr>
<td>LD_LIBRARY_PATH</td>
<td>$ALTERAOCLSDKROOT/host/linux64/lib</td>
</tr>
<tr>
<td></td>
<td>$AOCL_BOARD_PACKAGE_ROOT/linux64/lib</td>
</tr>
<tr>
<td></td>
<td>where $AOCL_BOARD_PACKAGE_ROOT points to the path of the Custom or Reference Platform</td>
</tr>
</tbody>
</table>

- To apply transient environment variable settings, open a command-line terminal and run the source
$ALTERAOCLSDKROOT/init_opencl.sh command.

Example script output:

AOCL_BOARD_PACKAGE_ROOT path is not set in environment
Setting to default s5_ref board.
If you want to target another board, do
set AOCL_BOARD_PACKAGE_ROOT=board_pkg_dir
Adding $ALTERAOCLSDKROOT/bin to PATH
Adding $ALTERAOCLSDKROOT/host/linux64/lib to LD_LIBRARY_PATH
Adding $AOCL_BOARD_PACKAGE_ROOT/linux64/lib to LD_LIBRARY_PATH

Installing the RTE on Big-Endian Systems

Install the RTE in a directory that you own.

Before you begin

1. You must have sudo or root privileges.
2. 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).

1. Install the software by invoking one of the following commands:
   - At a command prompt, type rpm -i acl-rte-<version>.ppc64.rpm to install the RTE
     in the default location (for example, opt/altera/aocl-rte).
   - To install the RTE in the default location with verbose progress reporting, type the rpm -ivh
     acl-rte-<version>.ppc64.rpm command.
   - To install the RTE in an alternate location, type the rpm -i --prefix <rte_destination_directory>
     acl-rte-<version>.ppc64.rpm command.
2. The installer sets the environment variable $ALTERAOCLSDKROOT to the path of the software
installation. Ensure that $ALTERAOCLSDKROOT points to the current version of the software.
Setting the Environment Variables on Big-Endian Systems

You have the option to apply environment variables permanently or transiently. The environment variable settings describe the FPGA board and the host runtime to the software.

Before you begin

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 ALTERAOCLSDKROOT 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.

Attention: If you set the environment variables permanently, you apply the settings once during installation. If you set the environment variables transiently, you must apply the settings during installation and during every subsequent session you run.

- To apply permanent environment variable settings, manually include the following paths to the corresponding environment variables:

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Path to Include</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATH</td>
<td>$ALTERAOCLSDKROOT/bin</td>
</tr>
<tr>
<td></td>
<td>where ALTERAOCLSDKROOT points to the path of the software installation</td>
</tr>
<tr>
<td>LD_LIBRARY_PATH</td>
<td>$ALTERAOCLSDKROOT/host/ppc64/lib</td>
</tr>
<tr>
<td></td>
<td>$AOCL_BOARD_PACKAGE_ROOT/ppc64/lib</td>
</tr>
<tr>
<td></td>
<td>whereAOCL_BOARD_PACKAGE_ROOT points to the path of the Custom or Reference Platform</td>
</tr>
</tbody>
</table>

- To apply transient environment variable settings, open a command window and run the source $ALTERAOCLSDKROOT/init_opencl.sh command.

Example script output:

```
AOCL_BOARD_PACKAGE_ROOT path is not set in environment
Setting to default s5_ref board.
If you want to target another board, do
  set AOCL_BOARD_PACKAGE_ROOT=board_pkg_dir
Adding $ALTERAOCLSDKROOT/bin to PATH
Adding $ALTERAOCLSDKROOT/host/ppc64/lib to LD_LIBRARY_PATH
Adding $AOCL_BOARD_PACKAGE_ROOT/ppc64/lib to LD_LIBRARY_PATH
```

Verifying the RTE Installation

Invoke the version utility command and verify that the correct version of the AOCL is installed.
Attention: The ARM processor on the Cyclone V SoC Development Kit does not support the AOCL version utility.

- At a command prompt, invoke the aocl version utility command. An output similar to the one below notifies you of a successful installation:

  aocl <version>.<build> (Altera Runtime Environment for OpenCL, Version <version> Build <build>, Copyright (C) <year> Altera Corporation)

- If installation was unsuccessful, reinstall the software. You can also refer to the Altera Software Installation and Licensing manual and the Knowledge Center on the Altera website for more information.

Related Information
- Altera Software Installation and Licensing
- Altera Knowledge Base

Licensing the Software

The RTE shares the same license as the AOCL. Obtain the AOCL license from the Altera Self Service Licensing Center.

Before you begin

For information on the licensing options and requirements, refer to the Licensing Altera Software section of the Altera Software Installation and Licensing manual, the LICENSE.txt file that accompanies each software, and the Altera Licensing page on the Altera website.

1. In the main page on the Altera website, click MYALTERA and log into your account.
2. Click the link to the Self-Service Licensing Center.
3. Perform the steps outlined in the Using the Self-Service Licensing Center section of the Altera Software Installation and Licensing manual to obtain and activate the license.
4. If you have a fixed license, append the <path_to_license_file>/<license_filename> file to the LM_LICENSE_FILE environment variable in the following manner:
   a. Create a backup copy of the provided license file.
   b. Save the new license file on your local hard drive.
   c. Append the <path_to_license_file>/<license_filename> file to the LM_LICENSE_FILE environment variable.
5. If you have a floating licence, append the <path_to_license_file>/<license_filename> file to the LMLICENSEFILE environment variable in the following manner:
   a. Obtain the port number and host name from the network or system administrator. Alternatively, the information is in the license file line SERVER <hostname> <8 to 12 character host or NIC ID> <port>. The license location for the user is <port>@<hostname>. If a port is not listed in the license file, specify @<hostname>.
   b. Modify the license file to update the port number and host name.
   c. Append the <path_to_license_file>/<license_filename> file to the LM_LICENSE_FILE environment variable setting.
Installing an FPGA Board

Before creating an OpenCL application for an FPGA board on Linux, you must first download and install the Custom Platform from your board vendor. Most Custom Platform installers require administrator privileges. To install your board into the host system, invoke the install utility command.

The steps below outline the board installation procedure. Some Custom Platforms require additional installation tasks. Consult your board vendor’s documentation for further information on board installation.

**Attention:** If you are installing the Cyclone V SoC Development Kit for use with the Cyclone V SoC Development Kit Reference Platform, refer to *Installing the Cyclone V SoC Development Kit* in the *Altera SDK for OpenCL Cyclone V SoC Getting Started Guide* for more information.

1. Follow your board vendor’s instructions to connect the FPGA board to your system.
2. Download the Custom Platform for your FPGA board from your board vendor’s website.
   For more information, refer to the OpenCL Reference Platforms page within the Altera website.
3. Install the Custom Platform in a directory that you own (that is, not a system directory).
4. Set the environment variable `AOCL_BOARD_PACKAGE_ROOT` to point to the location of the Custom Platform subdirectory containing the `board_env.xml` file.
   For example, for the Stratix V Network Reference Platform (s5_net), set `AOCL_BOARD_PACKAGE_ROOT` to point to the `<path_to_s5_net>/s5_net` directory.
5. Add the Custom Platform library paths to the `LD_LIBRARY_PATH` environment variable. You may apply permanent settings manually by adding the path to the memory-mapped (MMD) library within the Custom Platform. Alternatively, you may apply transient settings to the current command window by running the `SALTERAOCLSDKROOT/init_opencl.sh` script.
   For example, if you use s5_net, the Linux `LD_LIBRARY_PATH` setting is `$AOCL_BOARD_PACKAGE_ROOT/linux64/lib`.
   For more information on the `init_opencl.sh` script, refer to the *Setting the Environment Variables for Linux* section.
6. Invoke the command `aocl install` at a command prompt.
   Invoking `aocl install` also installs a board driver that allows communication between host applications and hardware kernel programs.
7. To query a list of FPGA devices installed in your machine, invoke the `aocl diagnose` command.
   The software generates an output that includes the `<device_name>`, which is an acl number that ranges from acl0 to acl15.
   For more information on querying the `<device_name>` of your accelerator board, refer to the *Querying the Device Name of Your FPGA Board* section.
8. To verify the successful installation of the FPGA board, invoke the command `aocl diagnose <device_name>` to run any board vendor-recommended diagnostic test.
Related Information

- Installing the Cyclone V SoC Development Kit on page 4-16
- Setting the Environment Variables on Big-Endian Systems on page 3-4
- Querying the Device Name of Your FPGA Board on page 3-7
- Setting the Environment Variables for Linux on page 3-2

Updating the Hardware Image on the FPGA

If applicable, before you execute an OpenCL kernel program on the FPGA, ensure that the flash memory of the FPGA contains a hardware image created using a current version of the OpenCL software.

**Remember:** If your Custom Platform requires that you preload a valid OpenCL image into the flash memory, for every major release of the Altera Complete Design Suite, program the flash memory of the FPGA with a hardware image compatible with the current version of the software.

Querying the Device Name of Your FPGA Board

Some AOCL utility commands require you to specify the device name (`<device_name>`). The `<device_name>` refers to the acl number (e.g. acl0 to acl15) that corresponds to the FPGA device. When you query a list of accelerator boards, the RTE produces a list of installed devices on your machine in the order of their device names.

- To query a list of installed devices on your machine, type `aocl diagnose` at a command prompt. The software generates an output that resembles the example shown below:

```
   aocl diagnose: Running diagnostic from ALTERAOCLSDKROOT/board/<board_name>/
<platform>/libexec
   Verified that the kernel mode driver is installed on the host machine.
   Using board package from vendor: <board_vendor_name>
   Querying information for all supported devices that are installed on the host
   machine ...
   
   device_name  Status  Information
   acl0         Passed  <descriptive_board_name>
   PCIe dev_id = <device_ID>, bus:slot.func = 02:00.00,
at Gen 2 with 8 lanes.
   FPGA temperature=43.0 degrees C.
   acl1         Passed  <descriptive_board_name>
   PCIe dev_id = <device_ID>, bus:slot.func = 03:00.00,
at Gen 2 with 8 lanes.
   FPGA temperature = 35.0 degrees C.
   
   Found 2 active device(s) installed on the host machine, to perform a full
diagnostic on a specific device, please run aocl diagnose <device_name>
```

Programming the Flash Memory of an FPGA on Linux

By default, you configure an FPGA using the hardware image stored in the flash memory of the device. When there is no power, the FPGA retains the hardware configuration file in the flash memory. When
you power up the system, it configures the FPGA circuitry based on this hardware image in the flash memory. Therefore, it is imperative that an OpenCL-compatible hardware configuration file is loaded into the flash memory of your FPGA.

Preloading an OpenCL image into the flash memory is necessary for the proper functioning of many Custom Platforms. For example, most PCI Express (PCIe)-based boards require a valid OpenCL image in flash memory so that hardware on the board can use the image to configure the FPGA device when the host system powers up for the first time. If the FPGA is not configured with a valid OpenCL image, the system will fail to enumerate the PCIe endpoint, or the driver will not function.

Before running any designs, ensure that the flash memory of your board has an valid OpenCL image that is compatible with the current OpenCL software version. Consult your board vendor's documentation for board-specific requirements.

**Caution:** When you load the hardware configuration file into the flash memory of the FPGA, maintain system power for the entire loading process, which might take a few minutes. Also, do not launch any host code that calls OpenCL kernels or might otherwise communicate with the FPGA board.

To load your hardware configuration file into the flash memory of your FPGA board, perform the following tasks:

1. Install any drivers or utilities that your Custom Platform requires.
2. Ensure that you set the `AOCL_BOARD_PACKAGE_ROOT` environment variable to point to the subdirectory in your Custom Platform that contains the `board_env.xml` file.
3. Download an example design for your Custom Platform.
   
   **Remember:** You can download example designs from the OpenCL Design Examples page, and extract the example to a location to which you have write access.
4. To load the hardware configuration file into the flash memory, invoke the `aocl flash <device_name> <example_design_filename>.aocx` command, where `<device_name>` refers to the acl number (e.g. acl0 to acl15) that corresponds to your FPGA device, and `<example_design_filename>.aocx` is the precompiled hardware configuration file you can find in the example design package.
5. Power down your device or computer and then power it up again.

Power cycling ensures that the FPGA configuration device retrieves the hardware configuration file from the flash memory and configures it into the FPGA.

**Warning:** Some Custom Platforms require you to power cycle the entire host system after programming the flash memory. For example, PCIe-based Custom Platforms might require a host system reboot to reenumerate the PCIe endpoint. Altera recommends that you power cycle the complete host system after programming the flash memory.

Related Information

OpenCL Design Examples page on the Altera website

---

**Executing an OpenCL Kernel on an FPGA**

You must build your OpenCL host application with the `Makefile` file, and run the application by invoking the `hello_world` executable. You need GNU development tools such as `gcc` and `make` to build the OpenCL application.
Building the Host Application

Build the host executable with the `<path_to_exm_opencl_hello_world_x64_linux_<version>>/hello_world/Makefile` file.

To build the host application, perform the following tasks:

1. Navigate to the `hello_world` directory.
2. 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_x64_linux_<version>>/hello_world/bin` directory.

Running the Host Application

To execute the OpenCL kernel on the FPGA, run the Linux host application that you built from the `Makefile`.

1. Add the path `ALTERAOCLSDKROOT/host/linux64/lib` to the `LD_LIBRARY_PATH` environment variable.
2. At a command prompt, navigate to the host executable within the `<path_to_exm_opencl_hello_world_x64_linux_<version>>/hello_world/bin` directory.
3. Invoke the `hello_world` executable.
   The `hello_world` executable executes the kernel code on the FPGA.

Output from Successful Kernel Execution

When you run the host application to execute your OpenCL kernel on the target FPGA, the RTE notifies you of a successful kernel execution.

Example output:

```
Found 1 OpenCL platforms.
Querying platform for info: ========
CL_PLATFORM_NAME = Altera SDK for OpenCL
CL_PLATFORM_VENDOR = Altera Corporation
CL_PLATFORM_VERSION = OpenCL 1.0 Altera SDK for OpenCL, Version <version>
Querying device for info: ========
CL_DEVICE_NAME = <board_name> : <descriptive_board_name>
CL_DEVICE_VENDOR = <board_vendor_name>
CL_DEVICE_VENDOR_ID = <board_vendor_ID>
CL_DEVICE_VERSION = OpenCL 1.0 Altera SDK for OpenCL, Version <version>
```

```
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 Altera's OpenCL Compiler!

Kernel execution is complete.

Uninstalling the Software

To uninstall the RTE for Linux, remove the software package via the Red Hat Package Manager (RPM) or
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 RTE via RPM at the command line, type the rpm -e aocl-rte command.
   b. Delete the aclrte-linux64 directory.
2. Remove $ALTERAOCLSDKROOT/bin from the PATH environment variable.
3. Remove $ALTERAOCLSDKROOT/host/linux64/lib from the LD_LIBRARY_PATH environment variable.
4. Remove the $ALTERAOCLSDKROOT environment variable.

Uninstalling the FPGA Board

To uninstall an FPGA board for Linux, invoke the uninstall utility command, uninstall the Custom
Platform, and unset the relevant environment variables. You must uninstall the existing FPGA board if
you migrate your OpenCL application to another FPGA board from a different Custom Platform.

To uninstall your FPGA board, perform the following tasks:
1. Following your board vendor's instructions to disconnect the board from your machine.
2. Invoke the aocl uninstall utility command to remove the current host computer drivers (for
   example, PCI Express (PCIe) drivers). The RTE uses these drivers to communicate with the FPGA
   board.
3. Uninstall the Custom Platform.
4. Unset the LD_LIBRARY_PATH environment variable.
5. Unset the $AOCL_BOARD_PACKAGE_ROOT environment variable.
Getting Started with the Altera RTE for OpenCL for SoC on Windows on page 4-1
To execute an OpenCL kernel onto an SoC, install the Altera Runtime Environment (RTE) for OpenCL, the SoC Embedded Design Suite (EDS). You must also build your host application using an ARM-specific Makefile.

Getting Started with the Altera RTE for OpenCL for SoC on Linux on page 4-12
To execute an OpenCL kernel onto an SoC, install the Altera Runtime Environment (RTE) for OpenCL, the SoC Embedded Design Suite (EDS). You must also build your host application using an ARM-specific Makefile.

Uninstalling the Altera RTE for OpenCL on page 4-23
To uninstall the Altera Runtime Environment (RTE) for OpenCL for SoC, delete the RTE directory and restore all modified environment variables to their previous settings.

Getting Started with the Altera RTE for OpenCL for SoC on Windows

To execute an OpenCL kernel onto an SoC, install the Altera Runtime Environment (RTE) for OpenCL, the SoC Embedded Design Suite (EDS). You must also build your host application using an ARM-specific Makefile.

1. Downloading the Altera SDK for OpenCL and the SoC EDS on page 4-2
To get started with the Altera RTE for OpenCL on the Cyclone V SoC Development Kit, download the AOCL and the SoC Embedded Design Suite (EDS) for Windows from the Download Center within the Altera website.

2. Installing the Altera SDK for OpenCL on page 4-2
To get started with the Altera RTE for OpenCL on the Cyclone V SoC Development Kit, install the AOCL for Windows.

3. Installing the SoC EDS on page 4-3
Install the Altera SoC Embedded Design Suite (EDS) for Windows to build your host application for OpenCL kernel deployment on an SoC board.

4. Licensing the Software on page 4-3
Obtain the AOCL license from the Altera Self Service Licensing Center.

5. Recompiling the Linux Kernel Driver on page 4-4
If you need to rebuild the Linux kernel driver, recompile the aclsoc Linux kernel driver to the exact version of the Linux kernel running on the SoC.
6. **Installing the RTE onto the SoC Board** on page 4-4
   The Altera Runtime Environment (RTE) for OpenCL installation package for Altera SoCs with 32-bit ARM processor is available in tar format.

7. **Installing the Cyclone V SoC Development Kit** on page 4-5
   To execute an OpenCL kernel on a Cyclone V SoC, first install the Cyclone V SoC Development Kit and configure it as described in the Altera SDK for OpenCL (AOCL) documentation.

8. **Executing an OpenCL Kernel on an SoC** on page 4-10
   Build your host application using the GNU Compiler Collection (GCC) cross-compiler available with the SoC Embedded Design Suite (EDS).

---

**Downloading the Altera SDK for OpenCL and the SoC EDS**

To get started with the Altera RTE for OpenCL on the Cyclone V SoC Development Kit, download the AOCL and the SoC Embedded Design Suite (EDS) for Windows from the Download Center within the Altera website.

The following instructions are for downloading the AOCL. If you wish to recompile the Linux kernel driver and write the SD card image on your own, download the Altera RTE for OpenCL for SoC instead.

To download the Altera RTE for OpenCL, click the **RTE** tab and select **Altera Runtime Environment for OpenCL Linux Cyclone V SoC TGZ**.

**1.** In the main page of the Altera website, click **MYALTERA** and log into your account.
   If you do not have a myAltera account, register for a new account.

**2.** Click **DOWNLOADS** to enter the Download Center.

**3.** Click **Altera SDK for OpenCL** to enter the download page for the subscription edition of the AOCL.

**4.** Select the software version. The default selection is the current version.

**5.** Select **Direct Download** as the download method.

**6.** Click the **Windows SDK** tab and then select **Altera SDK for OpenCL**. Click **More** beside **Download and install instructions** to view the download and installation procedure.

**7.** Perform the steps outlined in the download and installation instructions on the download page.

**8.** Download the SoC EDS by performing the following steps:
   a. From the Download Center, click **SoC EDS** to enter the download page for the subscription edition of the SoC EDS.
   b. Select the software version.
   c. Select **Windows** as the operating system.
   d. Select **Akamai DLM3 Download Manager** or **Direct Download** as the download method.
   e. If you select Akamai DLM3 Download Manager as the download method, click **Download**.
   f. If you select Direct Download as the download method, click **SoC Embedded Design Suite (EDS)**.
   g. Perform the steps outlined in the download and installation instructions on the download page.

**Related Information**

**Altera website**

**Installing the Altera SDK for OpenCL**

The Altera SDK for OpenCL (AOCL) Cyclone V SoC Development Kit Reference Platform (c5soc) includes an SD flash card image necessary for running OpenCL applications on the board. The SD flash card image includes the recompiled Linux kernel driver, preinstalled version of the Altera Runtime Environment (RTE) for OpenCL, and a script for setting environment variables.
To get started with the Altera RTE for OpenCL on the Cyclone V SoC Development Kit, install the AOCL for Windows.

**Before you begin**

You must have administrator privileges.

1. Run the `AOCLSetup-<version>-windows.exe` installer. Direct the installer to extract the software to an empty folder that you own (that is, not a system folder).
   
   The installation path must not contain any spaces (for example, `<home_directory>\altera\<version>\hld`).

2. Verify that the installer sets the user environment variable `ALTERAOCLSDKROOT` to point to the current version of the software.

**Installing the SoC EDS**

Install the Altera SoC Embedded Design Suite (EDS) for Windows to build your host application for OpenCL kernel deployment on an SoC board.

1. Run the `SoCEDSSetup-<version>-windows.exe` installer. Follow the installation instructions in the *Installing the SoC EDS* section of the *Altera SoC Embedded Design Suite User Guide*.

   
   For more information on the ARM DS-5 Altera Edition Toolkit, refer to the ARM DS-5 Altera Edition page of the ARM website.


**Related Information**

- Installing the SoC EDS
- Installing the ARM DS-5 Altera Edition Toolkit
- SoC EDS Licensing

**Licensing the Software**

The RTE shares the same license as the AOCL. Obtain the AOCL license from the Altera Self Service Licensing Center. The AOCL license allows the software to access relevant Quartus® II software functionalities and device support without an additional Quartus II software license.

**Before you begin**

For information on the licensing options and requirements, refer to the *Licensing Altera Software* section of the *Altera Software Installation and Licensing* manual, the `LICENSE.txt` file that accompanies each software, and the Altera Licensing page within the Altera website.

1. In the main page within the Altera website, click **MYALTERA** and log into your account.

2. Click the link to the Self-Service Licensing Center.

3. Perform the steps outlined in the *Requesting a License from the Self-Service Licensing Center* section of the *Altera Software Installation and Licensing* manual to obtain and activate the license.
Recompiling the Linux Kernel Driver

You must obtain a separate license for the SoC Embedded Design Suite (EDS).

4. If you have a fixed license, append the `<path_to_license_file>/license_filename>` file to the `LM_LICENSE_FILE` environment variable in the following manner:
   a. Create a backup copy of the provided license file.
   b. Save the new license file on your local hard drive.
   c. Append the `<path_to_license_file>/license_filename>` file to the `LM_LICENSE_FILE` environment variable.

5. If you have a floating licence, append the `<path_to_license_file>/license_filename>` file to the `LM_LICENSE_FILE` environment variable in the following manner:
   a. Obtain the port number and host name from the network or system administrator. Alternatively, the information is in the license file line `SERVER <hostname> <8 to 12 character host or NIC ID> <port>`.
      The license location for the user is `<port>@<hostname>`. If a port is not listed in the license file, specify `@<hostname>`.
   b. Modify the license file to update the port number and host name.
   c. Append the `<path_to_license_file>/license_filename>` file to the `LM_LICENSE_FILE` environment variable.

Related Information

- Altera Licensing page on the Altera website
- Altera Software Installation and Licensing
- Altera website

Recompiling the Linux Kernel Driver

**Attention:** If you download and install the Altera SDK for OpenCL, the Cyclone V SoC Development Kit Reference Platform (`ALTERAOCLSDKROOT/board/c5soc`) includes an SD card image (`linux_sd_card_image.tgz`) that contains the recompiled Linux kernel driver.

If you need to rebuild the Linux kernel driver, recompile the `aclsoc` Linux kernel driver to the exact version of the Linux kernel running on the SoC.

**Important:** You must recompile the `aclsoc` Linux kernel driver on your Linux development machine.

1. Unpack the `aocl-rte-<version>.arm32.tgz` tarball to a temporary directory on your development machine by typing the `tar -xvfz aocl-rte-<version>.arm32.tgz` command.
2. Navigate to the `ALTERAOCLSDKROOT/board/c5soc/driver` subdirectory of the unpacked `aclrte-arm32` package.
3. Perform the tasks outlined in the *Recompiling the Linux Kernel and the OpenCL Linux Kernel Driver* section of the Altera Cyclone V SoC Development Kit Reference Platform Porting Guide.

Related Information

Recompiling the Linux Kernel and the OpenCL Linux Kernel Driver

Installing the RTE onto the SoC Board

The Altera Runtime Environment (RTE) for OpenCL installation package for Altera SoCs with 32-bit ARM processor is available in tar format. To install the software, you must install it in a directory that you own, and set all the necessary environment variables.
1. Create an RTE directory on the board’s file system by typing the `mkdir <rte_destination_directory>` command.

2. Move the downloaded installation package `aclrte-arm32.tgz` to the RTE directory by typing the `mv aclrte-arm32.tgz <rte_destination_directory>` command.

3. Type `cd <rte_destination_directory>` to navigate to the RTE directory.

4. To unpack the tarball, type `tar -xvfz aclrte-arm32.tgz` at the command prompt.

5. Transfer the `aclsoc_drv.ko` file you built on your development machine into the `<rte_destination_directory>/board/c5soc/driver` directory on the SoC board.

6. Set the environment variables, as shown below.
   Altera recommends that you consolidate the settings of the environment variables into a file called `init_opencl.sh`. Then, run the command `source ./init_opencl.sh` to load all the environment variables and the OpenCL Linux kernel driver simultaneously.

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

**Installing the Cyclone V SoC Development Kit**

To execute an OpenCL kernel on a Cyclone V SoC, first install the Cyclone V SoC Development Kit and configure it as described in the Altera SDK for OpenCL (AOCL) documentation.
1. **Writing an SD Card Image onto the Micro SD Flash Card on Windows** on page 4-6  
   To write an 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.

2. **Configuring the SW3 Switches** on page 4-7  
   Configure the SW3 dual in-line package (DIP) switches on the Cyclone V SoC Development Kit.

3. **Setting Up Terminal Connection in Windows** on page 4-8  
   To set up the terminal connection for the Cyclone V SoC Development Kit in Windows, specify the USB virtual COM port settings.

4. **Setting Environment Variables and Loading OpenCL Linux Kernel Driver** on page 4-8  
   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.

5. **Connecting the Board to Network via Ethernet** on page 4-9  
   Connecting the Cyclone V SoC Development Kit to the host network allows you to transfer files to and from your SoC.

**Writing an SD Card Image onto the Micro SD Flash Card on Windows**

The Altera SDK for OpenCL (AOCL) includes a Cyclone V SoC Development Kit Reference Platform. To write an 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.
Before you begin

The SD card image `linux_sd_card_image.tgz` is available in the Cyclone V SoC Development Kit Reference Platform. 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 administrator privileges.

1. Extract the files from the `%ALTERAOCLSDKROOT%\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.

**Configuring the SW3 Switches**

Configure the SW3 dual in-line package (DIP) switches on the Cyclone V SoC Development Kit. The switch bank is located next to the SD card slot.

1. Set the SW3 DIP switches to the following positions:

<table>
<thead>
<tr>
<th>Switch</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ON</td>
</tr>
<tr>
<td>2</td>
<td>OFF</td>
</tr>
<tr>
<td>3</td>
<td>ON</td>
</tr>
<tr>
<td>4</td>
<td>OFF</td>
</tr>
<tr>
<td>5</td>
<td>ON</td>
</tr>
<tr>
<td>6</td>
<td>ON</td>
</tr>
</tbody>
</table>

   The figure below illustrates the physical configuration of the SW switches on the Cyclone V SoC Development Kit:
Setting Up Terminal Connection in Windows

To set up the terminal connection for the Cyclone V SoC Development Kit in Windows, 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 FDTI 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.

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.

1. After logging into the SoC 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_openc1.sh` file is available in the SD card image that you write onto the micro SD flash card. It contains the commands shown below:

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

---

**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.

1. Connect the hard processor system (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.

---

**Ensuring IP Address Acquisition**

After you connect the hard processor system (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.

---

**Before you begin**

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/udhcpc.d/50default: Adding DNS 137.57.142.218
   /etc/udhcpc.d/50default: Adding DNS 137.57.109.10
   /etc/udhcpc.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...
      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 hard processor system (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.

**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 (SSH) connection and transfer files using a Secure Copy (SCP) 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.

**Executing an OpenCL Kernel on an SoC**

The procedures outlined in this document are for building and running the host application for the hello_world example design. To execute the hello_world OpenCL kernel on your SoC, you must first create an `hello_world.aocx` Altera Offline Compiler Executable file. For instructions on obtaining the hello_world example design and creating the `hello_world.aocx` file, refer to the `Creating the FPGA Hardware Configuration File of an OpenCL Kernel` section of the Altera SDK for OpenCL Cyclone V SoC Getting Started Guide.

Build your host application using the GNU Compiler Collection (GCC) cross-compiler available with the SoC Embedded Design Suite (EDS).

**Related Information**

*Creating the Hardware Configuration File of an OpenCL Kernel for SoC*

**Building the Host Application**

Build your SoC-specific OpenCL host application using the GNU Compiler Collection (GCC) cross-compiler available with the SoC Embedded Design Suite (EDS).

1. Perform the following tasks to download the hello_world design example.
   a. Download the SoC-specific hello_world design example (`<version>` Arm32 Linux package (.tgz)) from the Hello World Design Example page within the Altera website.
   b. Extract `exm_opencl_hello_world_arm32_linux_<version>.tar` to a location to which you have write access.
   c. Ensure that you set the environment variable `AOCL_BOARD_PACKAGE_ROOT` to point to the Cyclone V SoC Development Kit Reference Platform (that is, `%ALTERAOCLSDKROOT%oard\c5soc`).

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_opencl_hello_world_arm32_linux_<version>>\hello_world` directory.

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_opencl_hello_world_arm32_linux_<version>>\hello_world\bin` directory.

Related Information

OpenCL Design Examples page on the Altera website

Running the Host Application

To execute the `hello_world.aocx` Altera Offline Compiler Executable file on the SoC, run the host application you built from the ARM-specific Makefile.

1. Log into your SoC board.
2. Copy the `hello_world.aocx` hardware configuration file and the hello_world host executable from the `<exm_opencl_hello_world_arm32_linux_<version>>\hello_world\bin` to the board.
3. Ensure that the `LD_LIBRARY_PATH` environment variable includes the `%ALTERAOCLSDKROOT%\host\arm32\lib`.
4. To execute the kernel on the SoC, at a command prompt, navigate to the host executable directory and run the hello_world host executable.

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:

```
Found 1 OpenCL platforms.
Querying platform for info:
========================================================================
CL_PLATFORM_NAME                         = Altera SDK for OpenCL
CL_PLATFORM_VENDOR                       = Altera Corporation
CL_PLATFORM_VERSION                = OpenCL 1.0 Altera SDK for OpenCL, Version <version>
Querying device for info:
========================================================================
CL_DEVICE_NAME                    = c5soc : Cyclone V SoC Development Kit
CL_DEVICE_VENDOR                        = Altera
CL_DEVICE_VENDOR_ID                     = 4466
CL_DEVICE_VERSION                 = OpenCL 1.0 Altera SDK for OpenCL, Version <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_PREFERREDVECTOR_WIDTH_CHAR   = 4
CL_DEVICE_PREFERREDVECTOR_WIDTH_SHORT  = 2
CL_DEVICE_PREFERREDVECTOR_WIDTH_INT    = 1
CL_DEVICE_PREFERREDVECTOR_WIDTH_LONG   = 1
CL_DEVICE_PREFERREDVECTOR_WIDTH_FLOAT  = 1
```
Getting Started with the Altera RTE for OpenCL for SoC on Linux

To execute an OpenCL kernel onto an SoC, install the Altera Runtime Environment (RTE) for OpenCL, the SoC Embedded Design Suite (EDS). You must also build your host application using an ARM-specific Makefile.

1. **Downloading the Altera SDK for OpenCL and the SoC EDS** on page 4-12
   To get started with the Altera RTE for OpenCL on the Cyclone V SoC Development Kit, download the AOCL and the SoC Embedded Design Suite (EDS) for Linux from the Download Center within the Altera website.

2. **Installing the Altera SDK for OpenCL** on page 4-13
   To get started with the Altera RTE for OpenCL on the Cyclone V SoC Development Kit, install the AOCL for Linux.

3. **Installing the SoC EDS** on page 4-14
   Install the Altera SoC Embedded Design Suite (EDS) for Linux to build your host application for OpenCL kernel deployment on an SoC board.

4. **Licensing the Software** on page 4-14
   Obtain the AOCL license from the Altera Self Service Licensing Center.

5. **Recompiling the Linux Kernel Driver** on page 4-15
   If you need to rebuild the Linux kernel driver, recompile the `aclsoc` Linux kernel driver to the exact version of the Linux kernel running on the SoC.

6. **Installing the RTE onto the SoC Board** on page 4-15
   The Altera Runtime Environment (RTE) for OpenCL installation package for Altera SoCs with 32-bit ARM processor is available in tar format.

7. **Installing the Cyclone V SoC Development Kit** on page 4-16
   To execute an OpenCL kernel on a Cyclone V SoC, first install the Cyclone V SoC Development Kit and configure it as described in the Altera SDK for OpenCL (AOCL) documentation.

8. **Executing an OpenCL Kernel on an SoC** on page 4-21
   Build your host application using the GNU Compiler Collection (GCC) cross-compiler available with the SoC Embedded Design Suite (EDS).

**Getting Started with the Altera RTE for OpenCL for SoC on Linux**

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 Altera’s OpenCL Compiler!

Kernel execution is complete.

**Downloading the Altera SDK for OpenCL and the SoC EDS**

To get started with the Altera RTE for OpenCL on the Cyclone V SoC Development Kit, download the AOCL and the SoC Embedded Design Suite (EDS) for Linux from the Download Center within the Altera website.

The following instructions are for downloading the AOCL. If you wish to recompile the Linux kernel driver and write the SD card image on your own, download the Altera RTE for OpenCL for SoC instead.
To download the Altera RTE for OpenCL, click the RTE tab and select Altera Runtime Environment for OpenCL Linux Cyclone V SoC TGZ.

1. In the main page of the Altera website, click MYALTERA and log into your account.
   If you do not have a myAltera account, register for a new account.
2. Click DOWNLOADS to enter the Download Center.
3. Click Altera SDK for OpenCL to enter the download page for the subscription edition of the AOCL.
4. Select the software version. The default selection is the current version.
5. Select Direct Download as the download method.
6. Click the Linux SDK tab and then select Altera SDK for OpenCL or Altera SDK for OpenCL Linux x86-64 RPM. Click More beside Download and install instructions to view the download and installation procedure.
7. Download the SoC EDS by performing the following steps:
   a. From the Download Center, click SoC EDS to enter the download page for the subscription edition of the SoC EDS.
   b. Select the software version.
   c. Select Linux as the operating system.
   d. Select Direct Download as the download method.
   e. Click SoC Embedded Design Suite (EDS).
      Download will begin immediately.
   f. Perform the steps outlined in the download and installation instructions on the download page.

Related Information

Altera website

Installing the Altera SDK for OpenCL

The Altera SDK for OpenCL (AOCL) Cyclone V SoC Development Kit Reference Platform (c5soc) includes an SD flash card image necessary for running OpenCL applications on the board. The SD flash card image includes the recompiled Linux kernel driver, a preinstalled version of the Altera Runtime Environment (RTE) for OpenCL, and a script for setting environment variables.

To get started with the Altera RTE for OpenCL on the Cyclone V SoC Development Kit, install the AOCL for Linux.

Before you begin

You must have sudo or root privileges.

1. To install the AOCL using the RPM Package Manager (RPM), invoke one of the following commands:
   • At a command prompt, type rpm -i aocl-sdk-<version>.x86_64.rpm to install the software in the default location (for example, opt/altera/aocl-sdk).
   • To install the software in the default location with verbose progress reporting, type rpm -ivh aocl-sdk-<version>.x86_64.rpm
   • To install the software in an alternate directory that you own (that is, not a system directory), type the rpm -i --prefix <aocl_destination_directory> aocl-sdk-<version>.x86_64.rpm command.
2. To install the AOCL using the GUI installer, run the AOCLSetup-<version>-linux.run installer. Direct the installer to extract the software to an empty folder that you own (that is, not a system folder).
The installation path must not contain any spaces (for example, `<home_directory>/altera/<version>/hld`).

3. Verify that the installer sets the user environment variable \textsc{ALTERAOCLSDKROOT} to point to the current version of the software.

## Installing the SoC EDS

Install the Altera SoC Embedded Design Suite (EDS) for Linux to build your host application for OpenCL kernel deployment on an SoC board.

The GNU Compiler Collection (GCC) tool chain is part of the SoC EDS installation package.

1. Run the `SoCEDSSetup-<version>-linux.run` installer. Follow the installation instructions in the \textit{Installing the SoC EDS} section of the Altera SoC Embedded Design Suite User Guide.


   For more information on the ARM DS-5 Altera Edition Toolkit, refer to the ARM DS-5 Altera Edition page of the ARM website.


### Related Information

- Installing the SoC EDS
- Installing the ARM DS-5 Altera Edition Toolkit
- SoC EDS Licensing

## Licensing the Software

The RTE shares the same license as the AOCL. Obtain the AOCL license from the Altera Self Service Licensing Center. The AOCL license allows the software to access relevant Quartus II software functionalities and device support without an additional Quartus II software license.

### Before you begin

For information on the licensing options and requirements, refer to the \textit{Licensing Altera Software} section of the Altera Software Installation and Licensing manual, the \texttt{LICENSE.txt} file that accompanies each software, and the Altera Licensing page within the Altera website.

1. In the main page within the Altera website, click \texttt{MYALTERA} and log into your account.

2. Click the link to the Self-Service Licensing Center.

3. Perform the steps outlined in the \textit{Requesting a License from the Self-Service Licensing Center} section of the Altera Software Installation and Licensing manual to obtain and activate the license.

   You must obtain a separate license for the SoC Embedded Design Suite (EDS).

4. If you have a fixed license, append the `<path_to_license_file>/<license_filename>` file to the \texttt{LM_LICENSE_FILE} environment variable in the following manner:
a. Create a backup copy of the provided license file.
b. Save the new license file on your local hard drive.
c. Append the `<path_to_license_file>/<license_filename>` file to the `LM_LICENSE_FILE` environment variable.

5. If you have a floating licence, append the `<path_to_license_file>/<license_filename>` file to the `LM_LICENSE_FILE` environment variable in the following manner:
   a. Obtain the port number and host name from the network or system administrator. Alternatively, the information is in the license file line `SERVER <hostname> <8 to 12 character host or NIC ID> <port>.
   The license location for the user is `<port>@<hostname>`. If a port is not listed in the license file, specify `@<hostname>`.
   b. Modify the license file to update the port number and host name.
   c. Append the `<path_to_license_file>/<license_filename>` file to the `LM_LICENSE_FILE` environment variable.

Related Information
- Altera Licensing page on the Altera website
- Altera Software Installation and Licensing
- Altera website

Recompiling the Linux Kernel Driver

**Attention:** If you download and install the Altera SDK for OpenCL, the Cyclone V SoC Development Kit Reference Platform (`ALTERAOCLSDKROOT/board/c5soc`) includes an SD card image (`linux_sd_card_image.tgz`) that contains the recompiled Linux kernel driver.

If you need to rebuild the Linux kernel driver, recompile the `aclsoc` Linux kernel driver to the exact version of the Linux kernel running on the SoC.

**Important:** You must recompile the `aclsoc` Linux kernel driver on your Linux development machine.

1. Unpack the `aocl-rte-<version>.arm32.tgz` tarball to a temporary directory on your development machine by typing the `tar -xvfz aocl-rte-<version>.arm32.tgz` command.
2. Navigate to the `ALTERAOCLSDKROOT/board/c5soc/driver` subdirectory of the unpacked `aclrte-arm32` package.
3. Perform the tasks outlined in the Recompiling the Linux Kernel and the OpenCL Linux Kernel Driver section of the Altera Cyclone V SoC Development Kit Reference Platform Porting Guide.

Related Information
- Recompiling the Linux Kernel and the OpenCL Linux Kernel Driver

Installing the RTE onto the SoC Board

The Altera Runtime Environment (RTE) for OpenCL installation package for Altera SoCs with 32-bit ARM processor is available in tar format. To install the software, you must install it in a directory that you own, and set all the necessary environment variables.
1. Create an RTE directory on the board’s file system by typing the `mkdir <rte_destination_directory>` command.

2. Move the downloaded installation package `aclrte-arm32.tgz` to the RTE directory by typing the `mv aclrte-arm32.tgz <rte_destination_directory>` command.

3. Type `cd <rte_destination_directory>` to navigate to the RTE directory.

4. To unpack the tarball, type `tar -xvfz aclrte-arm32.tgz` at the command prompt.

5. Transfer the `aclsoc_drv.ko` file you built on your development machine into the `<rte_destination_directory>/board/c5soc/driver` directory on the SoC board.

6. Set the environment variables, as shown below.

   Altera recommends that you consolidate the settings of the environment variables into a file called `init_opencl.sh`. Then, run the command `source ./init_opencl.sh` to load all the environment variables and the OpenCL Linux kernel driver simultaneously.

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

### Installing the Cyclone V SoC Development Kit

To execute an OpenCL kernel on a Cyclone V SoC, first install the Cyclone V SoC Development Kit and configure it as described in the Altera SDK for OpenCL (AOCL) documentation.
1. **Writing an SD Card Image onto the Micro SD Flash Card on Linux** on page 4-17
   To write an 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.

2. **Configuring the SW3 Switches** on page 4-18
   Configure the SW3 dual in-line package (DIP) switches on the Cyclone V SoC Development Kit.

3. **Setting Up Terminal Connection in Linux** on page 4-19
   To set up the terminal connection for the Cyclone V SoC Development Kit in Linux, specify the USB virtual COM port settings.

4. **Setting Environment Variables and Loading OpenCL Linux Kernel Driver** on page 4-19
   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.

5. **Connecting the Board to Network via Ethernet** on page 4-20
   Connecting the Cyclone V SoC Development Kit to the host network allows you to transfer files to and from your SoC.

**Writing an SD Card Image onto the Micro SD Flash Card on Linux**

The Altera SDK for OpenCL (AOCL) includes a Cyclone V SoC Development Kit Reference Platform. To write an SD card image onto the micro SD flash card on Linux, extract the SD card image from the
Before you begin

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

You must have sudo or root privileges.

1. To decompress the `$ALTERAOCLSDKROOT/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.

Configuring the SW3 Switches

Configure the SW3 dual in-line package (DIP) switches on the Cyclone V SoC Development Kit. The switch bank is located next to the SD card slot.

1. Set the SW3 DIP switches to the following positions:

<table>
<thead>
<tr>
<th>Switch</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ON</td>
</tr>
<tr>
<td>2</td>
<td>OFF</td>
</tr>
<tr>
<td>3</td>
<td>ON</td>
</tr>
<tr>
<td>4</td>
<td>OFF</td>
</tr>
<tr>
<td>5</td>
<td>ON</td>
</tr>
<tr>
<td>6</td>
<td>ON</td>
</tr>
</tbody>
</table>

The figure below illustrates the physical configuration of the SW switches on the Cyclone V SoC Development Kit:
Setting Up Terminal Connection in Linux

To set up the terminal connection for the Cyclone V SoC Development Kit in Linux, 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. Run the command `dmesg | tail` to determine which device the Future Technology Devices International (FTDI) driver assigns for the connection (e.g. `/dev/ttyUSB0`).
4. Setup the minicom as follows:
   a. Ensure that minicom is installed on your system. If not, invoke the `yum install minicom` command.
   b. Run `minicom -s` as root to enter the minicom setup mode.
   c. 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.

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.

1. After logging into the SoC 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:

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

### 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.

1. Connect the hard processor system (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.

### Ensuring IP Address Acquisition

After you connect the hard processor system (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.

#### Before you begin

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/udhcpc.d/50default: Adding DNS 137.57.142.218
   /etc/udhcpc.d/50default: Adding DNS 137.57.109.10
   /etc/udhcpc.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...
   ```
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 hard processor system (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.

**Mounting a Shared Drive**

The most convenient way to share files between a development PC and the Cyclone V SoC 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,rsize=32768,wsize=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 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.

**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 (SSH) connection and transfer files using a Secure Copy (SCP) 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.

**Executing an OpenCL Kernel on an SoC**

The procedures outlined in this document are for building and running the host application for the hello_world example design. To execute the hello_world OpenCL kernel on your SoC, you must first create an `hello_world.aocx` Altera Offline Compiler Executable file. For instructions on obtaining the hello_world example design and creating the `hello_world.aocx` file, refer to the Creating the FPGA Hardware Configuration File of an OpenCL Kernel section of the Altera SDK for OpenCL Cyclone V SoC Getting Started Guide.
Build your host application using the GNU Compiler Collection (GCC) cross-compiler available with the SoC Embedded Design Suite (EDS).

**Related Information**

**Creating the Hardware Configuration File of an OpenCL Kernel for SoC**

**Building the Host Application**

Build your SoC-specific OpenCL host application using the GNU Compiler Collection (GCC) cross-compiler available with the SoC Embedded Design Suite (EDS).

1. Perform the following tasks to download the hello_world design example.
   a. Download the SoC-specific hello_world design example (<version> Arm32 Linux package (.tgz)) from the Hello World Design Example page within the Altera website.
   b. Extract `exm_opencl_hello_world_arm32_linux_<version>.tar` to a location to which you have write access.
   c. Ensure that you set the environment variable `AOCL_BOARD_PACKAGE_ROOT` to point to the Cyclone V SoC Development Kit Reference Platform (that is, `$ALTERAOCLSDKROOT/board/c5soc`).

2. 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
   ```

3. Navigate to the `<path_to_exm_opencl_hello_world_arm32_linux_<version>>/hello_world` directory.

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_opencl_hello_world_arm32_linux_<version>>/hello_world/bin` directory.

**Running the Host Application**

To execute the `hello_world.aocx` Altera Offline Compiler Executable file on the SoC, run the host application you built from the ARM-specific `Makefile`.

1. Log into your SoC board.
2. Copy the `hello_world.aocx` hardware configuration file and the hello_world host executable from the `<exm_opencl_hello_world_arm32_linux_<version>>/hello_world/bin` to the board.
3. Ensure that the `LD_LIBRARY_PATH` environment variable includes the `$ALTERAOCLSDKROOT/host/arm32/lib`.
4. To execute the kernel on the SoC, at a command prompt, navigate to the host executable directory and run the hello_world host executable.

**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:

```
Found 1 OpenCL platforms.
Querying platform for info:
========================
CL_PLATFORM_NAME                         = Altera SDK for OpenCL
CL_PLATFORM_VENDOR                       = Altera Corporation
CL_PLATFORM_VERSION                = OpenCL 1.0 Altera SDK for OpenCL, Version <version>
```
Querying device for info:

---

CL_DEVICE_NAME                    = c5soc : Cyclone V SoC Development Kit
CL_DEVICE_VENDOR                        = Altera
CL_DEVICE_VENDOR_ID                     = 4466
CL_DEVICE_VERSION                 = OpenCL 1.0 Altera SDK for OpenCL, Version <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_MEMCACHELINE_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_ITEMDIMENSIONS       = 3
CL_DEVICE_MAX_WORK_ITEMDIMENSIONS       = 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 Altera's OpenCL Compiler!

Kernel execution is complete.

---

### Uninstalling the Altera RTE for OpenCL

To uninstall the Altera Runtime Environment (RTE) for OpenCL for SoC, delete the RTE directory and restore all modified environment variables to their previous settings.

1. Navigate to the root directory in the SoC board’s file system that contains the `<rte_destination_directory>` directory.
2. Type `rm -rf <rte_destination_directory>` to remove the RTE directory.
3. Remove the environment variable settings by typing the following commands:

   ```bash
   unset AOCL_BOARD_PACKAGE_ROOT
   unset ALTERAOCLSDKROOT
   unset PATH
   unset LD_LIBRARY_PATH
   ```
### Document Revision History

#### May 2015

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>May 2015</td>
<td>15.0.0</td>
<td>- Reorganized instructions into the following sections:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Introduction to the RTE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Getting Started with the RTE for Windows</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Getting Started with the RTE for Linux and Big-Endian Systems</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Getting Started with the RTE for Altera ARMv7-A SoC, which is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>further divided into Windows and Linux instructions</td>
</tr>
</tbody>
</table>

#### December 2014

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>December</td>
<td>14.1.0</td>
<td>- Reorganized information flow.</td>
</tr>
<tr>
<td>2014</td>
<td></td>
<td>- Updated Red Hat Enterprise Linux (RHEL) version support.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Added licensing information in the Licensing the Software section.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Included information on the <code>init_opencl</code> script for setting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>environment variables.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Updated the board uninstallation instructions to include the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>invocation of the <code>aocl uninstall</code> utility command.</td>
</tr>
</tbody>
</table>

#### June 2014

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 2014</td>
<td>14.0.0</td>
<td>- Initial release.</td>
</tr>
</tbody>
</table>