Altera RTE for OpenCL

Getting Started Guide

Subscribe

Send Feedback

OCL005-14.1.0
2014.12.15

101 Innovation Drive
San Jose, CA 95134
www.altera.com

Contents

Altera RTE for OpenCL Getting Started Guide

1

2014.12.15

OCL005-14.1.0 Subscribe Send Feedback

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 Altera Runtime Environment (RTE) for OpenCL 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 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 Altera Software Development Kit (SDK) for OpenCL (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

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

[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**.

Prerequisites

To install the Altera Runtime Environment (RTE) for OpenCL 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 of 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

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

Contents of the Altera RTE for OpenCL

The Altera Runtime Environment (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 of the Altera Software Development Kit (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 folder or directory that you own. The path to the software installation folder or directory is referenced by the *ALTERAOCLSDKROOT* environment variable. The following table highlights the contents of this folder or directory:

Windows Folder	Linux Directory	Big-Endian System Directory	ARM Directory	Description
bin	**bin**	**bin**	**bin**	High-level utilities. Include this folder or directory in your *PATH* environment variable.
board	**board**	—	**board**	The Reference Platform available with the RTE. **Important:** The Reference Platform for the RTE does not include the **hardware** subfolder or subdirectory.
host	**host**	**host**	**host**	Files necessary for compiling your host program.
host\include	**host/include**	**host/include**	**host/include**	OpenCL Specification version 1.0 header files and software interface files necessary for compiling and linking your host application. The **host/include/CL** subfolder or 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.

Windows Folder	Linux Directory	Big-Endian System Directory	ARM Directory	Description
host\windows64\lib	**host/linux64/lib**	**host/ppc64/lib**	**host/arm32/lib**	OpenCL host runtime libraries for the given target platform that provide the OpenCL platform and runtime APIs. These libraries are necessary for linking and running your host application. Prior to running your host application, include this folder or directory in the library search path. • For Linux and ARM, add the path to the *LD_LIBRARY_PATH* environment variable. • For Windows, add the path to the *PATH* environment variable.
host\windows64\bin	**host/linux64/bin**	**host/ppc64/bin**	**host/arm32/bin**	Platform-specific binary for the RTE Utility, runtime commands, and DLLs (for Windows) necessary for running your host application, wherever applicable. Include this folder in your *PATH* environment variable.
share\lib\perl	**share/lib/perl**	**share/lib/perl**	**share/lib/perl**	Perl scripts and support libraries for the RTE Utility.

Downloading the Software

Download the Altera Runtime Environment (RTE) for OpenCL from the Download Center of the Altera website.

1. On the main page of the Altera website, click **myAltera Account** to log on to your myAltera account.

   If you do not have a myAltera account, follow the instructions on the page to create one.
2. Click 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 the operating system (OS) (that is, Windows or Linux).
6. Select the download method.
   - For Windows systems, select the **Akamai DLM3 Download Manager** or **Direct Download**.
   - For Linux systems, select **Direct Download**.
7. Click the RTE tab and select the installation package you want to download.

   Note: For x86_64 Linux, you have to option to download the GUI installer (**Altera Runtime Environment for OpenCL Linux x86-64**) or the Red Hat Enterprise (RPM) installation package (**Altera Runtime Environment for OpenCL Linux x86-64 RPM**).
8. Click **Download Selected Files**.
9. **Attention:** If you use the RTE to deploy kernels on an SoC board, download the SoC Embedded Design Suite (EDS).

   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. Repeat steps 4 to 6.
c. Click **Download**.
The separate SoC EDS installation file installs the SoC EDS as a stand-alone software.

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

Installing the Software

The installation instructions for the Altera Runtime Environment (RTE) for OpenCL are platform-specific.

Installing the Software on 64-Bit Windows on page 1-5
Install the software in a folder that you own, and set all the necessary environment variables.

Installing the Software on x86_64 Linux on page 1-6
Install the software in a directory that you own, and set all the necessary environment variables.

Installing the Software on 64-Bit Linux Big-Endian System on page 1-8
Install the Altera Runtime Environment (RTE) for OpenCL on 64-bit big-endian system (for example, the IBM POWER system) in a directory that you own, and set all the necessary environment variables..

Installing the Software for Altera ARMv7-A SoC on page 1-10
You must recompile the Linux kernel driver before you install the RTE on your SoC board.

Verifying the Installation on page 1-11
To verify the successful installation of the software, invoke the `version` Altera Software Development Kit (SDK) for OpenCL (AOCL) utility command.

Installing the Software on 64-Bit Windows

Install the software in a folder that you own, and set all the necessary environment variables.

Before you begin

You must have administrator privileges.

Running the Installer on Windows

- 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 on Windows

You have the option to apply environment variables permanently or transiently.

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

Environment Variable	Path to Include
PATH	1. ***%ALTERAOCLSDKROOT*%\bin** 2. ***%ALTERAOCLSDKROOT*%\host\windows64\bin** 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.

Related Information

Verifying the Installation on page 1-11

Installing the Software on x86_64 Linux

Install the software in a directory that you own, and set all the necessary environment variables.

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**).

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

Running the Installer on Linux

- To install the Altera Runtime Environment (RTE) for OpenCL 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 aocl-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 aocl-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> aocl-rte-<version>.x86_64.rpm` command.
  2. To install the software using the GUI installer, run the **AOCLRTESetup-<*version*>-linux.run** installer. Direct the installer to extract the software to an empty folder that you own (that is, not a system directory).

     The installation path must not contain any spaces (for example, **/usr/altera/<*version*>/aclrte_linux64**).
  3. 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 Linux

You have the option to apply environment variables permanently or transiently.

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.

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

Environment Variable	Path to Include
PATH	***$ALTERAOCLSDKROOT*/bin** where *ALTERAOCLSDKROOT* points to the path of the software installation
LD_LIBRARY_PATH	***$ALTERAOCLSDKROOT*/host/linux64/lib** ***$AOCL_BOARD_PACKAGE_ROOT*/linux64/lib** where *AOCL_BOARD_PACKAGE_ROOT* points to the path of the Custom or Reference Platform

- 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/linux64/lib to LD_LIBRARY_PATH
Adding $AOCL_BOARD_PACKAGE_ROOT/linux64/lib to LD_LIBRARY_PATH
```

Related Information

Verifying the Installation on page 1-11

Installing the Software on 64-Bit Linux Big-Endian System

Install the Altera Runtime Environment (RTE) for OpenCL on 64-bit big-endian system (for example, the IBM POWER system) in a directory that you own, and set all the necessary environment variables.. you must install it

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**).

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

Running the Installer on Big-Endian Systems

1. Install the software by invoking one of the following commands:

- At a command prompt, type `rpm -i aocl-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 aocl-rte-<version>.ppc64.rpm` command.
- To install the RTE in an alternate location, type the `rpm -i --prefix <rte_destination_directory> aocl-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.

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.

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

Environment Variable	Path to Include
PATH	***$ALTERAOCLSDKROOT*/bin** where *ALTERAOCLSDKROOT* points to the path of the software installation
LD_LIBRARY_PATH	***$ALTERAOCLSDKROOT*/host/ppc64/lib** ***$AOCL_BOARD_PACKAGE_ROOT*/ppc64/lib** where*AOCL_BOARD_PACKAGE_ROOT* points to the path of the Custom or Reference Platform

- 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
```

Related Information

Verifying the Installation on page 1-11

Installing the Software for Altera ARMv7-A SoC

The Altera Cyclone V SoC Development Kit Reference Platform, available with the Altera Runtime Environment (RTE) for OpenCL, contains an SD flash card image necessary to run OpenCL applications on the board. The SD flash card image includes a preinstalled version of the RTE. You must recompile the Linux kernel driver before you install the RTE on your SoC board. Refer to the *Altera SDK for OpenCL Cyclone V SoC Getting Started Guide* for more information on host application development and kernel deployment.

Follow the instructions in this section to create a new SD flash card image or to install a different version of the RTE for Altera SoC at another location. The installation is a two-step process.

Recompiling the Linux Kernel Driver on page 1-10
Recompile the **aclsoc** Linux kernel driver to the exact version of the Linux kernel running on the SoC.

Installing the RTE onto the SoC Board on page 1-10
The Altera Runtime Environment (RTE) for OpenCL installation package for Altera SoCs with 32-bit ARM processor is available in tar format.

Related Information

Altera SDK for OpenCL Cyclone V SoC Getting Started Guide

Recompiling the Linux Kernel Driver

Recompile the **aclsoc** Linux kernel driver to the exact version of the Linux kernel running on the SoC.

Attention: 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

Altera Cyclone V SoC Development Kit Reference Platform Porting Guide

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
```

Verifying the Installation

To verify the successful installation of the software, invoke the `version` Altera Software Development Kit (SDK) for OpenCL (AOCL) utility command.

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 Center

Licensing the Software

The Altera Runtime Environment (RTE) for OpenCL shares the same license as the Altera Software Development Kit (SDK) for OpenCL. Obtain the Altera Software Development Kit (SDK) for OpenCL (AOCL) license from the Altera Self Service Licensing Center. The AOCL license allows the AOCL 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 *Altera Software Installation and Licensing* manual, the **LICENSE.txt** file that accompanies each software, and the Altera Licensing page of the Altera website.

1. Log on to your myAltera account from the Altera website.
2. Click the link to the Self-Service Licensing Center.
3. Perform the steps outlined in 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 *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
- Altera Software Installation and Licensing
- Altera website
- Self-Service Licensing Center

Installing an FPGA Board (install)

To target an FPGA board, you must first download and install the Custom Platform from your board vendor. Most Custom Platform installers require administrator privileges.

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 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 of the Altera website.
3. Install the Custom Platform to a folder or directory.
4. Set the environment variable *AOCL_BOARD_PACKAGE_ROOT* to the subfolder or subdirectory within the Custom Platform that contains 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 or directory.

5. Add the Custom Platform library paths to the *PATH* (Windows) or *LD_LIBRARY_PATH* (Linux) environment variable. You may apply permanent settings manually or apply transient settings to the current command window by running the **init_opencl** script.
   a. Apply permanent settings by adding the path to the memory-mapped (MMD) library within the Custom Platform.
   b. Apply transient settings by opening a new command window and running the ***%ALTERAOCLSDKROOT*%\init_opencl.bat** (Windows) or ***$ALTERAOCLSDKROOT*/init_opencl.sh** (Linux) script.

   For example, if you use s5_net, the Windows *PATH* environment variable setting is ***%AOCL_BOARD_PACKAGE_ROOT*%\windows64\bin**. The Linux *LD_LIBRARY_PATH* setting is ***$AOCL_BOARD_PACKAGE_ROOT*/linux64/lib**.

   For more information on the init_opencl script, refer to the *Setting the Environment Variables on Windows* and *Setting the Environment Variables on Linux* sections of the *Altera SDK for OpenCL Getting Started Guide*.
6. Invoke the command `aocl install` at a command prompt.

   Invoking the `aocl install` command 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

- **Querying the Device Name of Your FPGA Board (diagnose)** on page 1-15
- **Altera SDK for OpenCL Cyclone V SoC Getting Started Guide**

RTE Utility

The Altera Runtime Environment (RTE) for OpenCL utility is a subset of the Altera Software Development Kit (SDK) for OpenCL (AOCL) 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 (version)

To display the version of the Altera Runtime Environment (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 (help)

To display information on the Altera Runtime Environment (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 Runtime Environment 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

Altera SDK for OpenCL Programming Guide

Managing Host Application

The Altera Runtime Environment 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

Altera SDK for OpenCL Programming Guide

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 (diagnose)

Some Altera Software Development Kit (SDK) for OpenCL 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 software 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, perform the following task:

- 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>

DIAGNOSTIC_PASSED
```

Programming the Flash Memory of an FPGA

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 ahead of time so that hardware on the board can use the image to configure the FPGA device when the host system powers up. 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.

   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 Driver for Windows 7 and Windows Vista** page on the Altera website.
2. Ensure that you set the *AOCL_BOARD_PACKAGE_ROOT* environment variable to point to the subfolder or 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.

Setting Up the Host Application

The procedures to build and run the host application that executes the OpenCL kernel on your device are operating system (OS)-specific.

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 FPGA device, 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 Getting Started Guide*.

Attention: To create the **hello_world.aocx** file for big-endian systems, include the `--big-endian` option in the `aoc` command described in *Creating the FPGA Hardware Configuration File of an OpenCL Kernel*.

For instructions on obtaining the hello_world example design and creating the **hello_world.aocx** file for Cyclone V SoC, refer to the *Creating the FPGA Hardware Configuration File of an OpenCL Kernel for SoC* section of the *Altera SDK for OpenCL Cyclone V SoC Getting Started Guide*.

Related Information

- Altera SDK for OpenCL Getting Started Guide
- Altera SDK for OpenCL Cyclone V SoC Getting Started Guide

Building and Running the Host Application on Windows

Build your host application in Microsoft Visual Studio version 2010 Professional, and run the application by invoking the **hello_world.exe** executable. The Altera Software Development Kit (SDK) for OpenCL (AOCL) is compatible with 64-bit host binaries only.

Building the Host Application

The **hello_world.sln** file in the ***<path_to_exm_opencl_hello_world_x64_windows_<version>>*\hello_world** folder 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 **.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.

Building and Running the Host Application on Linux and Big-Endian Systems

You must build your 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 **Makefile** file, located in the ***<path_to_exm_opencl_hello_world_x64_linux_<version>>*/hello_world** directory.

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.

Building and Running the Host Application on Altera ARMv7-A SoC

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

1. To build the host application, perform the following tasks:
   a. 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
      ```
   b. Navigate to the ***<path_to_exm_opencl_hello_world_x32_linux_<version>>*/hello_world** directory.
   c. 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_x32_linux_<version>>*/hello_world/bin** directory.
2. To run the host application, perform the following tasks:

a. Log into your SoC board.
b. Copy the **hello_world.aocx** hardware configuration file and the **hello_world** host executable to the board.
c. Ensure that the *LD_LIBRARY_PATH* environment variable includes the path ***$ALTERAOCLSDKROOT/*****host/arm32/lib**.
d. At a command prompt, navigate to the host executable directory and run the **hello_world** executable.

   This command runs the host executable, which in turn 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 Altera Runtime Environment (RTE) for OpenCL 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 and the FPGA Board

Uninstalling the Software on 64-Bit Windows

To uninstall the Altera Runtime Environment (RTE) for OpenCL 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 Software on x86_64 Linux

To uninstall the Altera Runtime Environment (RTE) for OpenCL, 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 Software on 64-Bit Linux Big-Endian System

Use the Red Hat Package Manager (RPM) to uninstall the Altera Runtime Environment (RTE) for OpenCL at the command line. Delete the RTE directory and restore all modified environment variables to their previous settings.

1. To uninstall the RTE, type the `rpm -e aocl-rte` command.
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 Software for Altera ARMv7-A SoC

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

1. At a command prompt, type `cd` to navigate to the root directory.
2. Type `rm -rf <rte_destination_directory>` to remove the RTE directory.
3. Remove the environment variable settings by typing the following commands:

```
unset AOCL_BOARD_PACKAGE_ROOT

unset ALTERAOCLSDKROOT
unset PATH

unset LD_LIBRARY_PATH
```

Uninstalling the FPGA Board (uninstall)

To uninstall an FPGA board, 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. The Altera Runtime Environment (RTE) for OpenCL uses these drivers to communicate with the FPGA board.
   For example, PCI Express (PCIe) drivers
3. Uninstall the Custom Platform.
4. Unset the environment variable *LD_LIBRARY_PATH* (for Linux) or *PATH* (for Windows).
5. Unset the environment variable *AOCL_BOARD_PACKAGE_ROOT*.

Conclusion

You have performed successfully all the steps necessary to execute an OpenCL application on an Altera preferred FPGA-based accelerator board. You may follow the same steps to execute a different OpenCL example application using the Altera Runtime Environment (RTE) for OpenCL.

For more in-depth information on AOCL utilities and host application programming considerations, refer to the *Altera SDK for OpenCL Programming Guide*, available on the AOCL documentation page of the Altera website.

Related Information

Altera SDK for OpenCL Programming Guide

Document Revision History

Date	Version	Changes
December 2014	14.1.0	• Reorganized information flow. • Updated Red Hat Enterprise Linux (RHEL) version support. • Added licensing information in the *Licensing the Software* section. • Included information on the **init_opencl** script for setting environment variables. • Updated the board uninstallation instructions to include the invocation of the `aocl uninstall` utility command.
June 2014	14.0.0	• Initial release.