

Altera SDK for OpenCL

Getting Started Guide



Subscribe



Send Feedback

Last updated for Quartus Prime Design Suite: 16.0

UG-OCL001
2016.05.02

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

ALTERA
now part of Intel

Contents

Altera SDK for OpenCL Getting Started Guide.....	1-1
Prerequisites for the Altera SDK for OpenCL.....	1-1
Contents of the Altera SDK for OpenCL.....	1-2
Overview of the Altera SDK for OpenCL Setup Process.....	1-4
Getting Started with the Altera SDK for OpenCL for Windows.....	2-1
Downloading the Altera SDK for OpenCL.....	2-2
Installing the Altera SDK for OpenCL	2-2
Setting the Altera SDK for OpenCL User Environment Variables.....	2-3
Verifying Software Installation.....	2-5
Licensing the Software.....	2-5
Installing an FPGA Board.....	2-6
Verifying Host Runtime Functionality via Emulation.....	2-7
Downloading an OpenCL Design Example.....	2-8
Compiling a Kernel for Emulation.....	2-8
Building the Host Application.....	2-9
Emulating Your OpenCL Kernel.....	2-9
Creating the FPGA Hardware Configuration File of an OpenCL Kernel.....	2-10
Updating the Hardware Image on the FPGA.....	2-11
Querying the Device Name of Your FPGA Board.....	2-12
Programming the Flash Memory of an FPGA.....	2-12
Executing an OpenCL Kernel on an FPGA.....	2-13
Running the Host Application.....	2-14
Output from Successful Kernel Execution.....	2-14
Uninstalling the Software.....	2-15
Uninstalling the FPGA Board	2-15
Getting Started with the Altera SDK for OpenCL for Linux.....	3-1
Downloading the Altera SDK for OpenCL.....	3-2
Installing the Altera SDK for OpenCL.....	3-2
Setting the Altera SDK for OpenCL User Environment Variables.....	3-3
Verifying Software Installation.....	3-4
Licensing the Software.....	3-4
Installing an FPGA Board.....	3-5
Verifying Host Runtime Functionality via Emulation.....	3-7
Downloading an OpenCL Design Example.....	3-7
Compiling a Kernel for Emulation.....	3-8
Building the Host Application.....	3-8
Emulating Your OpenCL Kernel.....	3-8
Creating the FPGA Hardware Configuration File of an OpenCL Kernel.....	3-9
Updating the Hardware Image on the FPGA.....	3-11

Querying the Device Name of Your FPGA Board.....	3-11
Programming the Flash Memory of an FPGA.....	3-11
Executing an OpenCL Kernel on an FPGA.....	3-12
Running the Host Application.....	3-13
Output from Successful Kernel Execution.....	3-13
Uninstalling the Software.....	3-14
Uninstalling the FPGA Board.....	3-14
Document Revision History.....	A-1

2016.05.02

UG-OCL001



Subscribe



Send Feedback

The *Altera SDK for OpenCL Getting Started Guide* describes the procedures to install the Altera® Software Development Kit (SDK) for OpenCL™ (AOCL)⁽¹⁾. This document also contains instructions on how to compile an example OpenCL⁽²⁾ application with the AOCL.

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*.

The AOCL provides a compiler and tools for you to build and run OpenCL applications that target Altera FPGAs. The AOCL supports the embedded profile of the OpenCL Specification version 1.0.

Attention: If you only require the AOCL kernel deployment functionality, download and install the Altera Runtime Environment (RTE) for OpenCL. Refer to the *Altera RTE for OpenCL Getting Started Guide* for more information.

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

Attention: If you want to use the AOCL with 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 RTE for OpenCL Getting Started Guide](#)
- [Altera SDK for OpenCL Cyclone V SoC Getting Started Guide](#)

Prerequisites for the Altera SDK for OpenCL

To install the Altera SDK for OpenCL and create an OpenCL application for an Altera preferred accelerator board, your system must meet certain hardware, target platform, and software requirements.

⁽¹⁾ 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.

⁽²⁾ OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission of the Khronos Group™.

Hardware Requirements

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 [Altera SDK for OpenCL FPGA Platforms page](#) on the Altera website.

Development system requirements:

- You must have administrator privileges on the development system to install the necessary packages and drivers.
- The development system has at least 85 gigabytes (GB) of free disk space for software installation.
- The development system has at least 24 GB of RAM.

Tip: Refer to board vendor's documentation on the recommended system storage size.

- For PCI Express® (PCIe®) accelerator boards, the host machine motherboard must have an available PCIe port slot that is at least the same width (that is, the same number of PCIe lanes) as the board.

The host system must be running one of the following supported operating systems:

- For a list of supported Windows and Linux operating systems, refer to the [Operating System Support](#) page on the Altera website.

Software Prerequisites

Develop your host application using one of the following AOCL- and Quartus® Prime software-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.

Linux systems require the Perl command version 5 or later. Include the path to the Perl command in your *PATH* system environment variable setting.

Related Information

[Altera SDK for OpenCL FPGA Platforms page on the Altera website](#)

Contents of the Altera SDK for OpenCL

The Altera SDK for OpenCL provides logic components, drivers, and AOCL-specific libraries and files.

Logic Components

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

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

Drivers, Libraries and Files

The software installation process installs the AOCL into a directory that you own. The `ALTERAOCLSDK-ROOT` environment variable references the path to the AOCL installation directory.

Table 1-1: Select Contents of the AOCL Installation Directory

Windows Folder	Linux Directory	Description
bin	bin	User commands in the AOCL. Include this directory in your <i>PATH</i> environment variable setting.
board	board	The AOCL Custom Platform Toolkit and Reference Platforms available with the software. <ul style="list-style-type: none"> • The path to the Custom Platform Toolkit is <code>ALTERAOCLSDKROOT/board/custom_platform_toolkit</code>
ip	ip	Intellectual property (IP) cores used to compile device kernels.
host	host	Files necessary for compiling and running your host application.
host\include	host/include	OpenCL Specification version 1.0 header files and software interface files necessary for compiling and linking your host application. The 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.
host\ windows64\ lib	host/linux64/lib	OpenCL host runtime libraries that provide the OpenCL platform and runtime APIs. These libraries are necessary for linking your host application. To run an OpenCL application on Linux, include this directory in the <code>LD_LIBRARY_PATH</code> environment variable setting.

Windows Folder	Linux Directory	Description
host\ windows64\ bin	host/linux64/bin	Runtime commands and libraries necessary for running your host application, wherever applicable. For 64-bit Windows system, include this directory in your <i>PATH</i> environment variable setting. For Windows system, this folder contains runtime libraries. For Linux system, this directory contains platform-specific binary for the <code>aocl</code> utility command.
share	share	Architecture-independent support files.

Example OpenCL Applications

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

Related Information

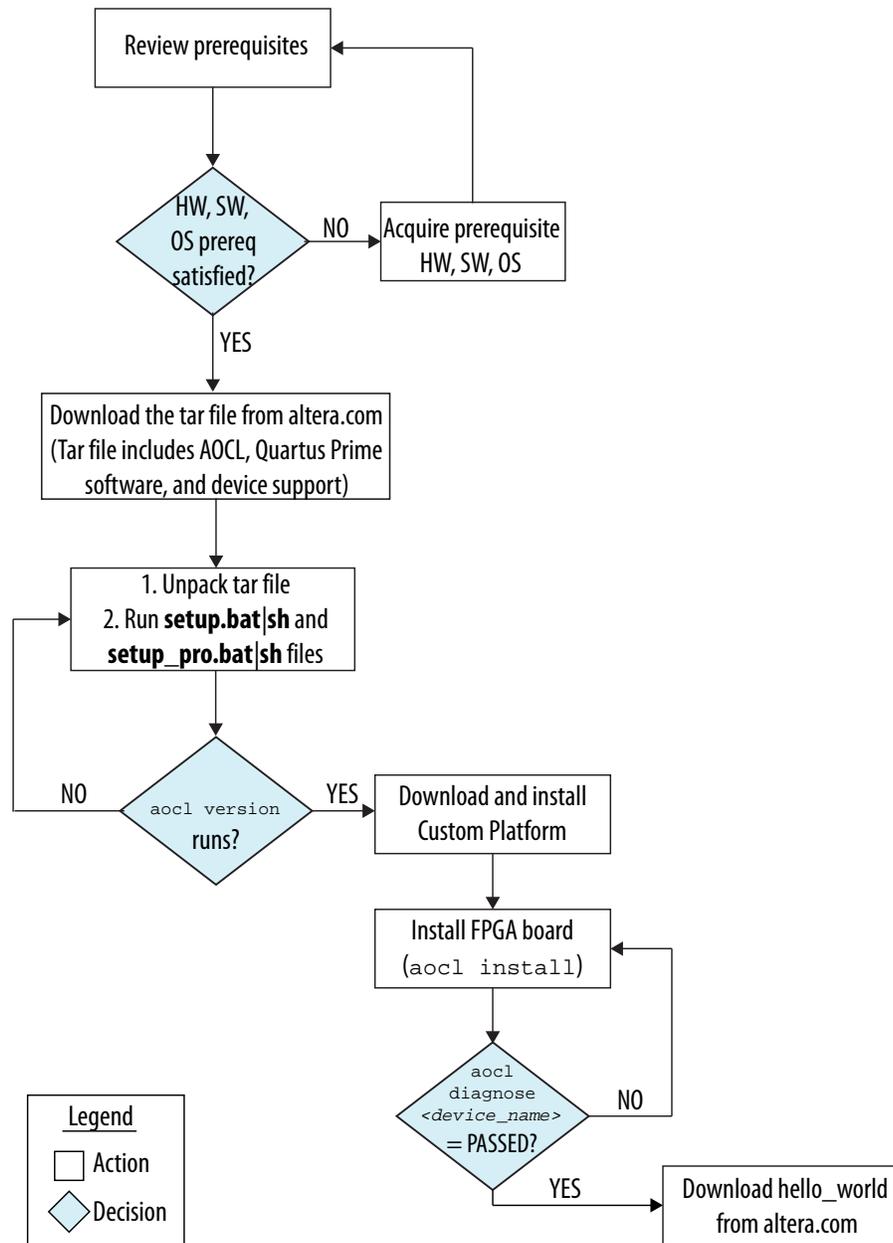
[OpenCL Design Examples page on the Altera website](#)

Overview of the Altera SDK for OpenCL Setup Process

The *Altera SDK for OpenCL Getting Started Guide* outlines the procedures for installing the Altera SDK for OpenCL and programming your FPGA.

The figure below summarizes the steps for setting up the necessary software and installing the FPGA board.

Figure 1-1: AOCL Installation Process Overview

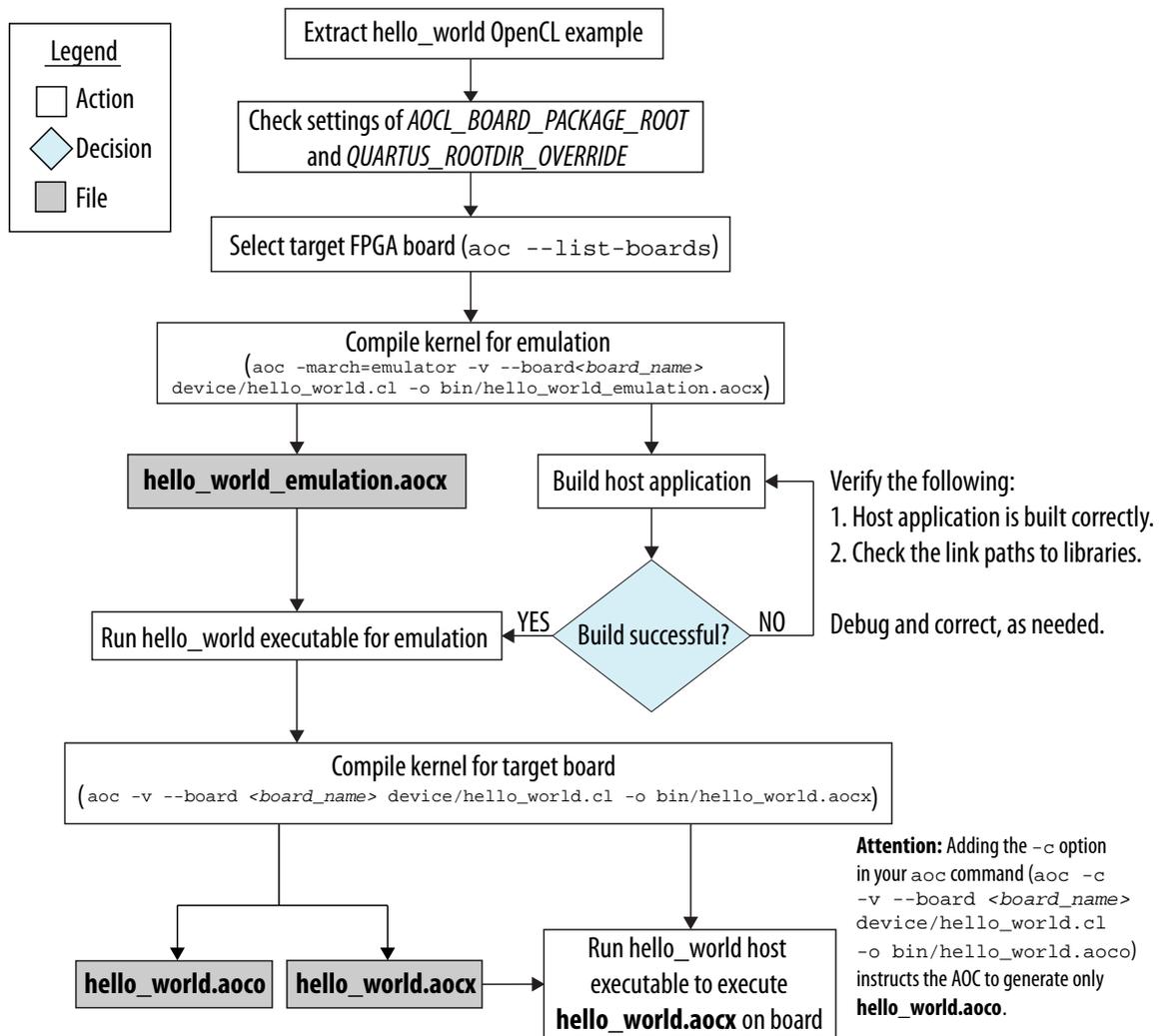


After you complete the initial software and hardware setup successfully, you can create a hardware image from the *hello_world* OpenCL design example.

Important: Before you program your FPGA with the hardware image, ensure that your FPGA contains an image created using a current version of the AOCL. Refer to *Updating the Hardware Image on the FPGA* for more information

The figure below summarizes the steps you perform to program your FPGA.

Figure 1-2: FPGA Programming Overview



Related Information

- [Updating the Hardware Image on the FPGA](#) on page 2-11
- [Updating the Hardware Image on the FPGA](#)

Getting Started with the Altera SDK for OpenCL for Windows

2

2016.05.02

UG-OCL001



Subscribe



Send Feedback

The Altera SDK for OpenCL setup process includes downloading and installing the software, installing the FPGA board, and then executing an OpenCL kernel on the FPGA.

- 1. Downloading the Altera SDK for OpenCL** on page 2-2
Download the installation package that contains the Altera SDK for OpenCL and all related software for Windows from the Download Center on the Altera website.
- 2. Installing the Altera SDK for OpenCL** on page 2-2
Install the Windows version of the Altera SDK for OpenCL in a folder that you own.
- 3. Setting the Altera SDK for OpenCL User Environment Variables** on page 2-3
You have the option to set the Altera SDK for OpenCL Windows user environment variables permanently or transiently.
- 4. Verifying Software Installation** on page 2-5
Invoke the `version` utility command and verify that the correct version of the Altera SDK for OpenCL is installed.
- 5. Licensing the Software** on page 2-5
Obtain the Altera SDK for OpenCL license from the Altera Self Service Licensing Center.
- 6. Installing an FPGA Board** on page 2-6
- 7. Verifying Host Runtime Functionality via Emulation** on page 2-7
Test the functionality of the host runtime by emulating an OpenCL design example using the Altera SDK for OpenCL Emulator.
- 8. Creating the FPGA Hardware Configuration File of an OpenCL Kernel** on page 2-10
Compile the `hello_world` kernel program using the Altera Offline Compiler to create the **hello_world.aocx** AOC Executable File.
- 9. Updating the Hardware Image on the FPGA** on page 2-11
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.
- 10. Executing an OpenCL Kernel on an FPGA** on page 2-13
Build your OpenCL host application in Microsoft Visual Studio version 2010 Professional, and run the application by invoking the **hello_world.exe** executable.
- 11. Uninstalling the Software** on page 2-15
To uninstall the Altera SDK for OpenCL for Windows, run the uninstaller, and restore all modified environment variables to their previous settings.

© 2016 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO
9001:2008
Registered

12. Uninstalling the FPGA Board on page 2-15

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 SDK for OpenCL

Download the installation package that contains the Altera SDK for OpenCL and all related software for Windows from the Download Center on the Altera website.

The Download Center provides an **AOCL-*<version>*-*<build>*-windows.tar** file that includes all of the following software and files:

- AOCL
 - Quartus Prime Standard Edition software for devices other than Arria® 10 devices
 - Quartus Prime Pro Edition software for Arria 10 devices
 - Device support
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 **Akamai DLM3 Download Manager** or **Direct Download** as the download method.
 6. Click the **Windows SDK** tab. Click **More** beside **Download and install instructions** to view the download and installation procedure.
 7. Click the download button to start the download process.
 8. Perform the steps outlined in the download and installation instructions on the download page.

Related Information

[Altera website](#)

Installing the Altera SDK for OpenCL

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

Install the Windows version of the Altera SDK for OpenCL in a folder that you own.

Before you begin

You must have administrator privileges.

To install the AOCL, Quartus Prime software, and device support files simultaneously, perform the following tasks:

1. Unpack the downloaded **AOCL-*<version>*-*<build>*-windows.tar** file into a temporary folder.
2. Run the installers. Altera recommends that you install both editions of the Quartus Prime software.
 - a. Run the **setup.bat** file to install the AOCL with the Quartus Prime Standard Edition software.
 - b. Run the **setup_pro.bat** file to install the AOCL with the Quartus Prime Pro Edition software.
3. **Note:** The installer sets the user environment variable `ALTERAOCLSDKROOT` to point to the path of the software installation.

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

If the returned path does not point to the location of the AOCL installation, edit the `ALTERAOCLSDKROOT` setting.

For instructions on modifying environment variable settings, refer to *Setting the Altera SDK for OpenCL User Environment Variables*.

Related Information

[Setting the Altera SDK for OpenCL User Environment Variables](#) on page 2-3

Setting the Altera SDK for OpenCL User Environment Variables

You have the option to set the Altera SDK for OpenCL Windows user 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.

Table 2-1: Altera SDK for OpenCL Windows User Environment Variable Settings

Environment Variable	Path to Include
<i>PATH</i>	<ol style="list-style-type: none"> 1. %ALTERAOCLSDKROOT%\bin 2. %ALTERAOCLSDKROOT%\windows64\bin 3. %ALTERAOCLSDKROOT%\host\windows64\bin <p>where <i>ALTERAOCLSDKROOT</i> points to the path of the software installation</p>

- To apply permanent environment variable settings, perform the following tasks:
 1. Click **Windows Start menu > Control Panel** (or open the Control Panel application in Windows 8.1).
 2. Click **System and Security > System**.
 3. In the **System** window, click **Advanced system settings**.
 4. Click the **Advanced** tab in the **System Properties** dialog box.
 5. Click **Environment Variables**.
The **Environment Variables** dialog box appears.
 6. To modify an existing environment variable setting, select the variable under **User variables for <user_name>** and then click **Edit**. In the **Edit User Variable** dialog box, type the environment variable setting in the **Variable value** field.
 7. If you add a new environment variable, click **New** under **User variables for <user_name>**. In the **New User Variable** dialog box, type the environment variable name and setting in the **Variable name** and **Variable value** fields, respectively.

For an environment variable with multiple settings, add semicolons to separate the settings.

- To apply transient environment variable settings, open a command window and run the **%ALTERAOCLSDKROOT%\init_opencil.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_opencil.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 Software Installation

Invoke the `version` utility command and verify that the correct version of the Altera SDK for OpenCL is installed.

- 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 SDK 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 Base on the Altera website for more information.

Related Information

- [Altera Software Installation and Licensing](#)
- [Altera Knowledge Base](#)

Licensing the Software

Obtain the Altera SDK for OpenCL license from the Altera Self Service Licensing Center.

The AOCL license allows the software to access relevant Quartus Prime software functionality and device support without an additional Quartus Prime 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 on the Altera website.

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

Related Information

- [Altera Licensing page on the Altera website](#)
- [Using the Self-Service Licensing Center](#)
- [Altera website](#)
- [Licensing Altera Software](#)

Installing an FPGA Board

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.

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. To download an Altera SDK for OpenCL Reference Platform (for example, the Stratix® V Network Reference Platform (s5_net)), refer to the Altera SDK for OpenCL FPGA Platforms page on the Altera website.
3. Install the Custom Platform in a folder that you own (that is, not a system folder).
4. **Note:** If you ran the `%ALTERAOCLSDKROOT%\init_opencil.bat` script to set the AOCL user environment variables, the script has set `AOCL_BOARD_PACKAGE_ROOT` to point to `%ALTERAOCLSDKROOT%\board\s5_ref`, by default.

Set the user 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 s5_net, set `AOCL_BOARD_PACKAGE_ROOT` to point to the `<path_to_s5_net>\s5_net` folder.

5. Set the `QUARTUS_ROOTDIR_OVERRIDE` user environment variable to point to the correct Quartus Prime software installation directory.

If you have an Arria 10 device, set `QUARTUS_ROOTDIR_OVERRIDE` to point to the installation directory of the Quartus Prime Pro Edition software. Otherwise, set `QUARTUS_ROOTDIR_OVERRIDE` to point to the installation directory of the Quartus Prime Standard Edition software.

6. Add the paths to the Custom Platform libraries (for example, the memory-mapped (MMD) library) to the `PATH` environment variable setting.

For example, if you use s5_net, the Windows `PATH` environment variable setting is `%AOCL_BOARD_PACKAGE_ROOT%\windows64\bin`.

For information on setting user environment variables and running the `init_opencil` script, refer to the *Setting the Altera SDK for OpenCL User Environment Variables* section.

7. **Remember:** You need administrative rights to install a board. To run a Windows command prompt as an administrator, click **Start > All Programs > Accessories**. Under **Accessories**, right click **Command Prompt**. In the right-click menu, click **Run as Administrator**.

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.

8. 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 `acl31`.

For more information on querying the `<device_name>` of your accelerator board, refer to the *Querying the Device Name of Your FPGA Board* section.

9. 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](#)
- [Querying the Device Name of Your FPGA Board](#) on page 2-12
- [Altera SDK for OpenCL FPGA Platforms page on the Altera website](#)
- [Setting the Altera SDK for OpenCL User Environment Variables](#) on page 2-3

Verifying Host Runtime Functionality via Emulation

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

Before you begin

1. Install a Custom or Reference Platform because emulation targets a specific FPGA board.
2. Verify that the `AOCL_BOARD_PACKAGE_ROOT` environment variable points to the location of the `board_env.xml` file within the Custom or Reference Platform. Open a Windows command window and then type `echo %AOCL_BOARD_PACKAGE_ROOT%` at the command prompt.
3. Verify that the `QUARTUS_ROOTDIR_OVERRIDE` environment variable points to the correct edition of the Quartus Prime software. Open a Windows command window and then type `echo %QUARTUS_ROOTDIR_OVERRIDE%` at the command prompt.

If the path to the installation folder of the Quartus Prime software is not returned, add it to the `QUARTUS_ROOTDIR_OVERRIDE` setting.

- For non-Arria 10 devices, `QUARTUS_ROOTDIR_OVERRIDE` points to the installation folder of the Quartus Prime Standard Edition software.
 - For Arria 10 devices, `QUARTUS_ROOTDIR_OVERRIDE` points to the installation folder of the Quartus Prime Pro Edition software.
4. Add the path to the `LINK.exe` file in Microsoft Visual Studio to the `PATH` user environment variable setting.
 5. Add the path to the Microsoft compilation time libraries in Microsoft Visual Studio to the `LIB` user environment variable setting.
1. [Downloading an OpenCL Design Example](#) on page 2-8
The OpenCL Design Examples page on the Altera website contains sample applications of varying complexities that you can download and run on your FPGA.

2. **Compiling a Kernel for Emulation** on page 2-8
To compile an OpenCL kernel for emulation, include the `-march=emulator` option in your `aoc` command.
3. **Building the Host Application** on page 2-9
The `<path_to_exm_openccl_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 OpenCL host application in the `main.cpp` file.
4. **Emulating Your OpenCL Kernel** on page 2-9
To emulate your OpenCL kernel, run the emulation `.aocx` file on the platform on which you build your kernel.

Downloading an OpenCL Design Example

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

The following instructions are for downloading the Hello World design.

1. In the OpenCL Design Examples page on the Altera website, under Basic Examples, click **Hello World**.
2. In the Hello World Design Example page, under Downloads, click `<version> x64 Windows package (.zip)` to download the compressed file for your platform.
3. Unzip the `exm_openccl_hello_world_x64_windows_<version>.zip` file and save it in a folder to which you have write access.

Related Information

[OpenCL Design Examples page on the Altera website](#)

Compiling a Kernel for Emulation

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

- To compile the kernel for your target FPGA board, at a command prompt, navigate to the `hello_world` design and then invoke the following command:

```
aoc -march=emulator -v --board <board_name> device/hello_world.cl -o bin/hello_world_emulation.aocx
```

The Altera Offline Compiler compiles the `hello_world.cl` kernel source file and creates the `hello_world_emulation.aocx` emulation-specific AOC Executable File in the `bin` subdirectory.

Building the Host Application

The `<path_to_exm_openc1_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 OpenCL host application in the `main.cpp` file.

To build the host application, perform the following tasks:

1. Open the `<path_to_exm_openc1_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_openc1_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.

Emulating Your OpenCL Kernel

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

To emulate your kernel, perform the following steps:

1. Run the utility command `aocl linkflags` to find out which libraries are necessary for building a host application. The software lists the libraries for both emulation and regular kernel compilation flows.
2. Link your host application to the libraries returned by the `aocl linkflags` utility command.
3. Move the `hello_world_emulation.aocx` file to the current working directory so that the host can locate it easily.
4. To run the host application for emulation, first define the number of emulated devices by invoking the `set CL_CONTEXT_EMULATOR_DEVICE_ALTERA=<number_of_devices>` command and then run the host application.
This command specifies the number of identical emulation devices that the Emulator needs to provide.
5. After you run the host application, unset the `CL_CONTEXT_EMULATOR_DEVICE_ALTERA` variable by invoking the `set CL_CONTEXT_EMULATOR_DEVICE_ALTERA=` command.

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

Related Information

[Running the Host Application](#) on page 2-14

Creating the FPGA Hardware Configuration File of an OpenCL Kernel

Download the Windows version of the `hello_world` OpenCL design example from the OpenCL Design Examples page on the Altera website. Compile the `hello_world` kernel program using the Altera Offline Compiler to create the **hello_world.aocx** AOC Executable File. The **.aocx** file is the FPGA hardware configuration file.

After you successfully install your FPGA board and emulate your kernel, you can create a **.aocx** file that executes on the device. The steps below describe the process of creating a **.aocx** file from the `hello_world` example design. For more information on the OpenCL design examples, refer to the OpenCL Design Examples page on the Altera website.

1. **Note:** The AOC compiles an OpenCL kernel for a target FPGA board. You must install the FPGA board before creating a hardware configuration file. If you have not installed the FPGA board, refer to *Installing an FPGA Board* for more instructions.

Verify that the environment variable `AOCL_BOARD_PACKAGE_ROOT` points to a local subfolder within your Custom Platform that contains the **board_env.xml** file. Open a Windows command window and then type `echo %AOCL_BOARD_PACKAGE_ROOT%` at the command prompt.

If `AOCL_BOARD_PACKAGE_ROOT` is not set, or if the returned path does not point to the Custom Platform for your FPGA board, reset the `AOCL_BOARD_PACKAGE_ROOT` environment variable setting. Refer to *Setting the Altera SDK for OpenCL User Environment Variables* for more instructions.

2. Verify that the `QUARTUS_ROOTDIR_OVERRIDE` environment variable points to the correct edition of the Quartus Prime software. Open a Windows command window and then type `echo %QUARTUS_ROOTDIR_OVERRIDE%` at the command prompt.

If the path to the installation folder of the Quartus Prime software is not returned, add it to the `QUARTUS_ROOTDIR_OVERRIDE` setting.

- For non-Arria 10 devices, `QUARTUS_ROOTDIR_OVERRIDE` points to the installation folder of the Quartus Prime Standard Edition software.
 - For Arria 10 devices, `QUARTUS_ROOTDIR_OVERRIDE` points to the installation folder of the Quartus Prime Pro Edition software.
3. Select your target FPGA board. To list the FPGA boards available in your Custom Platform, invoke the command `aoc --list-boards` at a command prompt.

For more information on the `--list-boards` option of the `aoc` command, refer to the *Listing the Available FPGA Boards in Your Custom Platform (--list-boards)* section of the *Altera SDK for OpenCL Programming Guide*.

4. At a command prompt, navigate to the `hello_world` design folder containing the **hello_world.cl** file that you used for emulation.
5. To compile the kernel for your target FPGA board, invoke the following command:

```
aoc -v --board <board_name> device/hello_world.cl -o bin/hello_world.aocx
```

This command performs the following tasks:

- Generates the Quartus Prime design project files from the OpenCL source code.
- Checks for initial syntax errors.
- Performs basic optimizations.
- Creates a **hello_world** folder containing necessary intermediate files.
- Creates the *Altera Offline Compiler Object File (.aoco)*.
- Creates the **.aocx** file.

Attention: The **.aocx** file might take hours to build, depending on the complexity of the kernel. To view the progress of the compilation on-screen, include the `-v` flag in your `aoc` command. An example output is shown below.

```
aoc: Environment checks are completed successfully.
aoc: You are now compiling the full flow!!
aoc: Selected target board <board_name>
aoc: Running OpenCL parser....
aoc: OpenCL parser completed successfully.
aoc: Compiling....
aoc: Linking with IP library ...
aoc: First stage compilation completed successfully.
aoc: Setting up project for CvP revision flow....
aoc: Hardware generation completed successfully.
```

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

For more information on the `--board <board_name>` option of the `aoc` command, refer to the *Compiling a Kernel for a Specific FPGA Board (--board <board_name>)* section of the *Altera SDK for OpenCL Programming Guide*.

For more information on the `-v` option of the `aoc` command, refer to the *Generating Compilation Progress Report (-v)* section of the *Altera SDK for OpenCL Programming Guide*.

For more information on the `-o <filename>` option of the `aoc` command, refer to the *Specifying the Name of an AOC Output File (-o <filename>)* section of the *Altera SDK for OpenCL Programming Guide*.

Related Information

- [Listing the Available FPGA Boards in Your Custom Platform \(--list-boards\)](#)
- [Compiling a Kernel for a Specific FPGA Board \(--board <board_name>\)](#)
- [Generating Compilation Progress Report \(-v\)](#)
- [Specifying the Name of an AOC Output File \(-o <filename>\)](#)
- [OpenCL Design Examples page on the Altera website](#)
- [Setting the Altera SDK for OpenCL User Environment Variables](#) on page 2-3
- [Installing an FPGA Board](#) on page 2-6

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 Quartus Prime 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 Altera 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 acl31) that corresponds to the FPGA device. When you query a list of accelerator boards, the AOCL 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>

DIAGNOSTIC_PASSED
```

Programming the Flash Memory of an FPGA

Configure the FPGA by loading the hardware image of an Altera SDK for OpenCL example design into 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 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.

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. Verify 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. Open a Windows command window and type `echo %AOCL_BOARD_PACKAGE_ROOT%` at the command prompt. If the returned path does not point to the location of the `board_env.xml` file within your Custom Platform, follow the instructions in *Setting the Altera SDK for OpenCL User Environment Variables* to modify the environment variable setting.
3. Download an example design for your Custom Platform.

Remember: 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 `acl31`) that corresponds to your FPGA device, and `<example_design_filename>.aocx` is the hardware configuration file you create from the `<example_design_filename>.cl` file 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](#)
- [OpenCL Design Examples page on the Altera website](#)
- [Setting the Altera SDK for OpenCL User Environment Variables](#) on page 2-3

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 Altera SDK for OpenCL is compatible with 64-bit host binaries only.

Related Information

[Building the Host Application](#) on page 2-9

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_openccl_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 Altera SDK 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

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

1. From the Windows Start Menu shortcut, navigate to the **Altera <version> Standard Edition** folder.
2. Select **Uninstall Quartus Prime Standard Edition**.
The uninstallation wizard appears.
3. In the uninstallation wizard, perform the following tasks:
 - a. Select **Individual components** and then click **Next**.
 - b. Select **Altera SDK for OpenCL** and then click **Next**.
The uninstaller uninstalls the AOCL.
4. To uninstall the AOCL associated with the Quartus Prime Pro Edition software, select **Uninstall Quartus Prime Pro Edition** from the **Altera <version> Pro Edition** folder to open the uninstallation wizard.
 - a. Repeat [step 3](#).
5. Remove the following paths from the *PATH* environment variable:
 - a. **%ALTERAOCLSDKROOT%\bin**
 - b. **%ALTERAOCLSDKROOT%\host\windows64\bin**
6. Remove the *ALTERAOCLSDKROOT* environment variable.
7. Remove the *QUARTUS_ROOTDIR_OVERRIDE* 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 that belongs to 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, PCIe drivers). The Altera SDK for OpenCL 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.

2016.05.02

UG-OCL001



Subscribe



Send Feedback

The Altera SDK for OpenCL setup process includes downloading and installing the software, installing the FPGA board, and then executing an OpenCL kernel on the FPGA.

1. **Downloading the Altera SDK for OpenCL** on page 3-2
Download the installation package that contains the Altera SDK for OpenCL and all related software for Linux from the Download Center on the Altera website.
2. **Installing the Altera SDK for OpenCL** on page 3-2
Install the Linux version of the Altera SDK for OpenCL in a directory that you own.
3. **Setting the Altera SDK for OpenCL User Environment Variables** on page 3-3
You have the option to set the Altera SDK for OpenCL Linux user environment variables permanently or transiently.
4. **Verifying Software Installation** on page 2-5
Invoke the `version` utility command and verify that the correct version of the Altera SDK for OpenCL is installed.
5. **Licensing the Software** on page 2-5
Obtain the Altera SDK for OpenCL license from the Altera Self Service Licensing Center.
6. **Installing an FPGA Board** on page 3-5
To install your board into a Linux host system, invoke the `install` utility command.
7. **Verifying Host Runtime Functionality via Emulation** on page 3-7
Test the functionality of the host runtime by emulating an OpenCL design example using the Altera SDK for OpenCL Emulator.
8. **Creating the FPGA Hardware Configuration File of an OpenCL Kernel** on page 3-9
Compile the `hello_world` kernel program using the Altera Offline Compiler to create the **hello_world.aocx** AOC Executable File.
9. **Updating the Hardware Image on the FPGA** on page 2-11
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.
10. **Executing an OpenCL Kernel on an FPGA** on page 3-12
You must build your OpenCL host application with the **Makefile** file, and run the application by invoking the **hello_world** executable.
11. **Uninstalling the Software** on page 3-14
To uninstall the Altera SDK for OpenCL for Linux, remove the software package via the GUI uninstaller, then delete the software directory and restore all modified environment variables to their previous settings.

© 2016 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO
9001:2008
Registered

ALTERA
now part of Intel

12. Uninstalling the FPGA Board on page 3-14

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 SDK for OpenCL

Download the installation package that contains the Altera SDK for OpenCL and all related software for Linux from the Download Center on the Altera website.

The Download Center provides an **AOCL-<version>-<build>-linux.tar** file that includes all of the following software and files:

- AOCL
 - Quartus Prime Standard Edition software for devices other than Arria 10 devices
 - Quartus Prime Pro Edition software for Arria 10 devices
 - Device support
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 click **More** beside **Download and install instructions** to view the download and installation procedure.
 7. Click the download button to start the download process.
 8. Perform the steps outlined in the download and installation instructions on the download page.

Related Information

[Altera website](#)

Installing the Altera SDK for OpenCL

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

Install the Linux version of the Altera SDK for OpenCL 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 AOCL, Quartus Prime software, and device support files simultaneously, perform the following tasks:

1. Unpack the downloaded **AOCL-<version>-<build>-linux.tar** file into a temporary directory.
2. Run the installers. Altera recommends that you install both editions of the Quartus Prime software.
 - a. Run the **setup.sh** file to install the AOCL with the Quartus Prime Standard Edition software.
 - b. Run the **setup_pro.sh** file to install the AOCL with the Quartus Prime Pro Edition software.
3. **Note:** The installer sets the environment variable `ALTERAOCLSDKROOT` to point to the path of the software installation.

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

If the returned path does not point to the location of the AOCL installation, edit the `ALTERAOCLSDKROOT` setting.

For instructions on modifying environment variable settings, refer to *Setting the Altera SDK for OpenCL User Environment Variables*.

Related Information

[Setting the Altera SDK for OpenCL User Environment Variables](#) on page 3-3

Setting the Altera SDK for OpenCL User Environment Variables

You have the option to set the Altera SDK for OpenCL Linux user 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.

Table 3-1: Altera SDK for OpenCL Linux User Environment Variable Settings

Environment Variable	Path to Include
<code>PATH</code>	<code>\$ALTERAOCLSDKROOT/bin</code> where <code>ALTERAOCLSDKROOT</code> points to the path of the software installation

Environment Variable	Path to Include
<code>LD_LIBRARY_PATH</code>	<p><code>\$ALTERAOCLSDKROOT/host/linux64/lib</code></p> <p><code>\$AOCL_BOARD_PACKAGE_ROOT/linux64/lib</code></p> <p>where <code>AOCL_BOARD_PACKAGE_ROOT</code> points to the path of the Custom or Reference Platform</p>

- To apply permanent environment variable settings, open a shell and then type the `export <variable_name>=<variable_setting>:$(<variable_name>)` command. For example, the command `export PATH="$ALTERAOCLSDKROOT/bin":$PATH` adds **`$ALTERAOCLSDKROOT/bin`** to the list of `PATH` settings.
- To apply transient environment variable settings, open a command-line terminal and run the `source $ALTERAOCLSDKROOT/init_openc1.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
```

Verifying Software Installation

Invoke the `version` utility command and verify that the correct version of the Altera SDK for OpenCL is installed.

- 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 SDK 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 Base on the Altera website for more information.

Related Information

- [Altera Software Installation and Licensing](#)
- [Altera Knowledge Base](#)

Licensing the Software

Obtain the Altera SDK for OpenCL license from the Altera Self Service Licensing Center.

The AOCL license allows the software to access relevant Quartus Prime software functionality and device support without an additional Quartus Prime 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 on the Altera website.

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

Related Information

- [Altera Licensing page on the Altera website](#)
- [Using the Self-Service Licensing Center](#)
- [Altera website](#)
- [Licensing Altera Software](#)

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 a Linux 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.

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. To download an Altera SDK for OpenCL Reference Platform (for example, the Altera Stratix V Network Reference Platform), refer to the Altera SDK for OpenCL FPGA Platforms page on the Altera website.
3. Install the Custom Platform in a directory that you own (that is, not a system directory).
4. **Note:** If you ran the **\$ALTERAOCLSDKROOT/init_openc1.sh** script to set the AOCL user environment variables, the script has set **AOCL_BOARD_PACKAGE_ROOT** to point to **\$ALTERAOCLSDK-ROOT/board/s5_ref**, by default.

Set the user 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 **s5_net**, set **AOCL_BOARD_PACKAGE_ROOT** to point to the **<path_to_s5_net>/s5_net** directory.

5. Set the **QUARTUS_ROOTDIR_OVERRIDE** user environment variable to point to the correct Quartus Prime software installation directory. Open a shell and then type **echo \$QUARTUS_ROOTDIR_OVERRIDE** at the command prompt.

If you have an Arria 10 device, set **QUARTUS_ROOTDIR_OVERRIDE** to point to the installation directory of the Quartus Prime Pro Edition software. Otherwise, set **QUARTUS_ROOTDIR_OVERRIDE** to point to the installation directory of the Quartus Prime Standard Edition software.

6. Add the paths to the Custom Platform libraries (for example, memory-mapped (MMD) library) to the **LD_LIBRARY_PATH** environment variable setting.
For example, if you use **s5_net**, the Linux **LD_LIBRARY_PATH** setting is **\$AOCL_BOARD_PACKAGE_ROOT/linux64/lib**.

For information on setting Linux user environment variables and running the **init_openc1** script, refer to the *Setting the Altera SDK for OpenCL User Environment Variables* section.

7. **Remember:** You need **sudo** or **root** privileges to install a board.

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.

8. 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 **acl31**.

For more information on querying the **<device_name>** of your accelerator board, refer to the *Querying the Device Name of Your FPGA Board* section.

9. To verify the successful installation of the FPGA board, invoke the command **aocl diagnose <device_name>** to run any board vendor-recommended diagnostic test.

Verifying Host Runtime Functionality via Emulation

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

Before you begin

1. Install a Custom or Reference Platform because emulation targets a specific FPGA board.
2. Verify that the `AOCL_BOARD_PACKAGE_ROOT` environment variable points to the location of the `board_env.xml` file within the non-SoC Custom or Reference Platform. Open a shell and then type `echo $AOCL_BOARD_PACKAGE_ROOT` at the command prompt.
3. Verify that the `QUARTUS_ROOTDIR_OVERRIDE` environment variable points to the correct edition of the Quartus Prime software. Open a shell and then type `echo $QUARTUS_ROOTDIR_OVERRIDE` at the command prompt.

If the path to the installation directory of the Quartus Prime software is not returned, add it to the `QUARTUS_ROOTDIR_OVERRIDE` setting.

- For non-Arria 10 devices, `QUARTUS_ROOTDIR_OVERRIDE` points to the installation folder of the Quartus Prime Standard Edition software.
 - For Arria 10 devices, `QUARTUS_ROOTDIR_OVERRIDE` points to the installation folder of the Quartus Prime Pro Edition software.
4. Verify that the `LD_LIBRARY_PATH` environment variable setting includes the paths identified in *Setting the Altera SDK for OpenCL User Environment Variables*. Open a shell and then type `echo $LD_LIBRARY_PATH` at the command prompt.

If the returned paths do not include `$ALTERAOCLSDKROOT/host/linux64/lib` and `$AOCL_BOARD_PACKAGE_ROOT/linux64/lib`, add them to the `LD_LIBRARY_PATH` setting.

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

1. **Downloading an OpenCL Design Example** on page 3-7
The OpenCL Design Examples page on the Altera website contains sample applications of varying complexities that you can download and run on your FPGA.
2. **Compiling a Kernel for Emulation** on page 3-8
To compile an OpenCL kernel for emulation, include the `-march=emulator` option in your `aoc` command.
3. **Building the Host Application** on page 3-8
Build the host executable with the `<path_to_exm_opencl_hello_world_x64_linux_<version>>/hello_world/Makefile` file.
4. **Emulating Your OpenCL Kernel** on page 3-8
To emulate your OpenCL kernel, run the emulation `.aocx` file on the platform on which you build your kernel.

Downloading an OpenCL Design Example

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

The following instructions are for downloading the Hello World design.

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

Related Information

[OpenCL Design Examples page on the Altera website](#)

Compiling a Kernel for Emulation

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

- To compile the kernel for your target FPGA board, at a command prompt, navigate to the `hello_world` design and then invoke the following command:

```
aoc -march=emulator -v --board <board_name> device/hello_world.cl -o bin/hello_world_emulation.aocx
```

The Altera Offline Compiler compiles the **hello_world.cl** kernel source file and creates the **hello_world_emulation.aocx** emulation-specific AOC Executable File in the **bin** subdirectory.

Building the Host Application

Build the host executable with the `<path_to_exm_openc1_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_openc1_hello_world_x64_linux_<version>>/hello_world/bin` directory.

Emulating Your OpenCL Kernel

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

To emulate your kernel, perform the following steps:

1. Run the utility command `aocl linkflags` to find out which libraries are necessary for building a host application. The software lists the libraries for both emulation and regular kernel compilation flows.
2. Link your host application to the libraries returned by the `aocl linkflags` utility command.
3. Move the **hello_world_emulation.aocx** file to the current working directory so that the host can locate it easily.
4. To run the host application for emulation, invoke the `env CL_CONTEXT_EMULATOR_DEVICE_ALTERA=<number_of_devices> <host_application_filename>` command.

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

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

Related Information

[Running the Host Application](#) on page 3-13

Creating the FPGA Hardware Configuration File of an OpenCL Kernel

Download the Linux version of the `hello_world` OpenCL design example from the OpenCL Design Examples page on the Altera website. Compile the `hello_world` kernel program using the Altera Offline Compiler to create the `hello_world.aocx` AOC Executable File. The `.aocx` file is the FPGA hardware configuration file.

After you successfully install your FPGA board and emulate your kernel, you can create a `.aocx` file that executes on the device. The steps below describe the process of creating a `.aocx` file from the `hello_world` example design. For more information on the OpenCL design examples, refer to the OpenCL Design Examples page on the Altera website.

1. **Note:** The AOC compiles an OpenCL kernel for a target FPGA board. You must install the FPGA board before creating a hardware configuration file. If you have not installed the FPGA board, refer to *Installing an FPGA Board* for more instructions.

Verify that the environment variable `AOCL_BOARD_PACKAGE_ROOT` points to a local subdirectory within your Custom Platform that contains the `board_env.xml` file. Open a shell and then type `echo $AOCL_BOARD_PACKAGE_ROOT` at the command prompt.

If `AOCL_BOARD_PACKAGE_ROOT` is not set, or if the returned path does not point to the Custom Platform for your FPGA board, reset the `AOCL_BOARD_PACKAGE_ROOT` environment variable setting. Refer to *Setting the Altera SDK for OpenCL User Environment Variables* for more instructions.

2. Verify that the `QUARTUS_ROOTDIR_OVERRIDE` environment variable points to the correct edition of the Quartus Prime software. Open a shell and then type `echo $QUARTUS_ROOTDIR_OVERRIDE` at the command prompt.

If the path to the installation directory of the Quartus Prime software is not returned, add it to the `QUARTUS_ROOTDIR_OVERRIDE` setting.

- For non-Arria 10 devices, `QUARTUS_ROOTDIR_OVERRIDE` points to the installation folder of the Quartus Prime Standard Edition software.
- For Arria 10 devices, `QUARTUS_ROOTDIR_OVERRIDE` points to the installation folder of the Quartus Prime Pro Edition software.

3. Select your target FPGA board. To list the FPGA boards available in your Custom Platform, invoke the command `aoc --list-boards` at a command prompt.

For more information on the `--list-boards` option of the `aoc` command, refer to the *Listing the Available FPGA Boards in Your Custom Platform (--list-boards)* section of the *Altera SDK for OpenCL Programming Guide*.

4. At a command prompt, navigate to the `hello_world` design directory containing the `hello_world.cl` file that you used for emulation.
5. To compile the kernel for your target FPGA board, invoke the following command:

```
aoc -v --board <board_name> device/hello_world.cl -o bin/  
hello_world.aocx
```

This command performs the following tasks:

- Generates the Quartus Prime design project files from the OpenCL source code.
- Checks for initial syntax errors.
- Performs basic optimizations.
- Creates a **hello_world** directory containing necessary intermediate files.
- Creates the *Altera Offline Compiler Object File (.aoco)*.
- Creates the **.aocx** file.

Attention: The **.aocx** file might take hours to build, depending on the complexity of the kernel. To view the progress of the compilation on-screen, include the `-v` flag in your `aoc` command. An example output is shown below.

```
aoc: Environment checks are completed successfully.  
You are now compiling the full flow!!  
aoc: Selected target board <board_name>  
aoc: Running OpenCL parser...  
aoc: OpenCL parser completed successfully.  
aoc: Compiling...  
aoc: Linking with IP library ...  
aoc: First stage compilation completed successfully.  
aoc: Setting up project for CVP revision flow...  
aoc: Hardware generation completed successfully.
```

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

For more information on the `--board <board_name>` option of the `aoc` command, refer to the *Compiling a Kernel for a Specific FPGA Board (--board <board_name>)* section of the *Altera SDK for OpenCL Programming Guide*.

For more information on the `-v` option of the `aoc` command, refer to the *Generating Compilation Progress Report (-v)* section of the *Altera SDK for OpenCL Programming Guide*.

For more information on the `-o <filename>` option of the `aoc` command, refer to the *Specifying the Name of an AOC Output File (-o <filename>)* section of the *Altera SDK for OpenCL Programming Guide*.

Related Information

- [Installing an FPGA Board](#) on page 3-5
- [OpenCL Design Examples page on the Altera website](#)
- [Listing the Available FPGA Boards in Your Custom Platform \(--list-boards\)](#)
- [Compiling a Kernel for a Specific FPGA Board \(--board <board_name>\)](#)
- [Generating Compilation Progress Report \(-v\)](#)
- [Specifying the Name of an AOC Output File \(-o <filename>\)](#)

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 Quartus Prime 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 Altera 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 acl31) that corresponds to the FPGA device. When you query a list of accelerator boards, the 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>

DIAGNOSTIC_PASSED
```

Programming the Flash Memory of an FPGA

Configure the FPGA by loading the hardware image of an Altera SDK for OpenCL example design into 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 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. Verify 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. Open a shell and then type `echo $AOCL_BOARD_PACKAGE_ROOT` at the command prompt. If the returned path does not point to the location of the `board_env.xml` file within your Custom Platform, follow the instructions in *Setting the Altera SDK for OpenCL User Environment Variables* to modify the environment variable setting.

3. Download an example design for your Custom Platform.

Remember: 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 `acl31`) that corresponds to your FPGA device, and `<example_design_filename>.aocx` is the hardware configuration file you create from the `<example_design_filename>.cl` file 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](#)
- [Setting the Altera SDK for OpenCL User Environment Variables](#) on page 3-3

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.

Related Information

[Building the Host Application](#) on page 3-8

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_openccl_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 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 Altera SDK for OpenCL for Linux, remove the software package via the GUI uninstaller, then delete the software directory and restore all modified environment variables to their previous settings.

1. Remove the software package by performing one of the following tasks:
 - a. To uninstall the AOCL, run the **aocl-<version>-uninstall.run** program located in the **<install directory>/uninstall** directories for the Quartus Prime Standard Edition software and the Quartus Prime Pro Edition software.
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.
5. Remove the *QUARTUS_ROOTDIR_OVERRIDE* 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 that belongs to 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, PCIe drivers). The Altera SDK for OpenCL 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.

Document Revision History



2016.05.02

UG-OCL001



Subscribe



Send Feedback

Table A-1: Document Revision History of the Altera SDK for OpenCL Getting Started Guide

Date	Version	Changes
May 2016	2016.05.02	<ul style="list-style-type: none">Replaced the lists of supported Windows and Linux versions to a link to the Operating System Support page on the Altera website.Added the <code>%ALTERAOCLSDKROOT%\windows64\bin</code> setting to the list of Windows environment variables.Corrected the Windows instructions for setting the <code>CL_CONTEXT_EMULATOR_DEVICE_ALTERA</code> variable for emulating multiple devices.
November 2015	2015.11.02	<ul style="list-style-type: none">Changed instances of <i>Quartus II</i> to <i>Quartus Prime</i>.Added Windows 8.1 to supported Windows versions.Modified download and installation instructions for the tar file that includes the AOCL, Quartus Prime software, and device support.Deprecated and removed AOCL-only installation instructions because they are invalid for the current version.Added instructions to verify host runtime functionality by emulating the <code>hello_world</code> example design.Modified the figure <i>FPGA Programming Overview</i> to include emulation in the programming flow.Updated uninstallation instructions.
May 2015	15.0.0	<ul style="list-style-type: none">Reorganized instructions into the following sections:<ul style="list-style-type: none">Getting Started with the AOCL on WindowsGetting Started with the AOCL on Linux

© 2016 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO
9001:2008
Registered



Date	Version	Changes
December 2014	14.1.0	<ul style="list-style-type: none"> • Reorganized information flow. • Updated Red Hat Enterprise Linux (RHEL) version support. • Included the <i>Contents of the AOCL</i> section. • Updated licensing instructions for the new Altera Software Development Kit (SDK) for OpenCL (AOCL) single license. • Updated board uninstallation instructions to include the <code>aocl uninstall</code> utility command. • Included information on the init_opencl script for setting environment variables. • Grouped software and board uninstallation instructions under <i>Uninstalling the Software and the FPGA Board</i>.
June 2014	14.0.0	<ul style="list-style-type: none"> • Updated the <i>Prerequisites</i> section. • Updated the figure <i>AOCL Installation Process Overview</i>. • Updated software download instructions. • Updated AOCL installation and uninstallation instructions for Windows. • For Linux systems: <ul style="list-style-type: none"> • Added the sections <i>Installing the AOCL on Linux Using RPM</i> and <i>Uninstalling the AOCL on Linux Using RPM</i>. • Updated the section <i>Installing the AOCL on Linux Using the GUI Installer</i>. • Added the section <i>Licensing the Software</i>. • Updated the section <i>Installing an FPGA Board</i> with updated instructions on querying your devices and running diagnostic tests. • Updated the section <i>Creating the FPGA Hardware Configuration File of an OpenCL Kernel</i>: <ul style="list-style-type: none"> • Updated path that you have to set for <code>AOCL_BOARD_PACKAGE_ROOT</code>. • Updated example AOC output for compiling hello_world.cl with the <code>-v</code> option. • Updated AOC output. • Added the section <i>Identifying the Device Name of Your FPGA Board</i>. • Modified instructions for building and running the host application with updated <code>hello_world</code> directory and file names. • Added the section <i>Uninstalling an FPGA Board</i>.
December 2013	13.1.1	<ul style="list-style-type: none"> • Updated the <i>Prerequisites</i> section to include a reminder to install Linux kernel source, headers, and GCC.

Date	Version	Changes
November 2013	13.1.0	<ul style="list-style-type: none"> • Reorganized information flow. • Updated the <i>Prerequisites</i> section. • Updated board installation instructions. • Updated software download instructions. • Inserted the section <i>Installing the AOCL and the Quartus II Software Together</i>. • Updated software installation and uninstallation instructions. • Inserted the following figures: <ul style="list-style-type: none"> • <i>AOCL Installation Process Overview</i> • <i>FPGA Programming Overview</i> • Removed the <i>Licensing</i> section. • Removed all board-specific installation and configuration instructions. • Changed example OpenCL application used to demonstrate kernel configuration and FPGA programming from <code>moving_average</code> to <code>hello_world</code>. • Inserted the section <i>Updating the Hardware Image on the FPGA</i>, which contained the updated flash programming instructions. • Removed the section <i>Installing the USB-Blaster Driver on Windows</i>. • Updated output from successful execution of <code>hello_world</code> kernel on FPGA for Windows and Linux systems. • Removed the figure <i>Contents of the Moving Average Example</i>. • Removed the figure <i>Opening host.sln in Visual Studio</i>.
June 2013	13.0 SP1.0	<ul style="list-style-type: none"> • Updated requisite Quartus II and AOCL software versions from 13.0 to 13.0 SP1. • Inserted the figure <i>A Correct Windows Device Manager After Complete Board Driver Installation for a BittWare Board</i>. • Updated the vendor and device IDs in the <i>Verifying the Functionality of the BittWare Board</i> section for Windows. • Updated the AOCL installation instructions for Linux systems that do not contain a <code>.cshrc</code> or a <code>.bashrc</code> file in the directory. • Updated path to the AOCL design examples. • Updated the figure <i>Contents of the Moving Average Example</i>. • Updated flash programming instructions.

Date	Version	Changes
May 2013	13.0.1	<ul style="list-style-type: none"> • Renamed the OpenCL_SDK folder or directory to AOCL. • Inserted warning about the AOCL installation dependency on .cshrc and .bashrc files for Linux systems. • Included reminder to BittWare board users about installing the BittWare development software. • Inserted warning about potential Jungo WinDriver installation failure for systems running on Windows 7. Included reinstallation instructions. • Inserted warnings about error messages displayed for <code>aocl</code> commands that have not been implemented for the BittWare FPGA board. • Inserted caution message about setting the environment variable <code>AOCL_BOARD_PACKAGE_ROOT</code>. • Updated board driver installation procedures for Windows and Linux systems. • Modified the path to the default location of the AOCL for Windows and Linux systems. • Modified the path name added to the <code>PATH</code> environment variable when installing the AOCL on Linux systems. The path name should be <code>\$QUARTUS_ROOTDIR/bin</code> instead of <code>\$QUARTUS_ROOTDIR/bin64</code>.
May 2013	13.0.0	<ul style="list-style-type: none"> • Updated installation and compilation procedures. • Incorporated licensing procedure. • Updated flash programming procedure and moved it to Appendix A. • Updated links to software and documentation download pages.
November 2012	12.1.0	Initial release.