OpenCL* on Intel® Programmable Acceleration Card with Intel® Arria® 10 GX FPGA Quick Start User Guide

Updated for Intel® Acceleration Stack for Intel® Xeon® CPU with FPGAs: **1.2.1**

Online Version

Send Feedback

UG-20275

ID: 683831

Version: 2020.06.05

Contents

1. About this Document

This document describes the OpenCL* implementation for the Intel® Programmable Acceleration Card with Intel Arria® 10 GX FPGA.

1.1. Conventions

Table 1. Document Conventions

Convention	Description
#	Precedes a command that indicates the command is to be entered as root.
$	Indicates a command is to be entered as a user.
`This font`	Filenames, commands, and keywords are printed in this font. Long command lines are printed in this font. Although long command lines may wrap to the next line, the return is not part of the command; do not press enter.
<variable_name>	Indicates the placeholder text that appears between the angle brackets must be replaced with an appropriate value. Do not enter the angle brackets.

1.2. Acceleration Glossary

Table 2. Acceleration Stack for Intel Xeon® CPU with FPGAs Glossary

Term	Abbreviation	Description
Intel Acceleration Stack for Intel Xeon® CPU with FPGAs	Acceleration Stack	A collection of software, firmware, and tools that provides performance-optimized connectivity between an Intel FPGA and an Intel Xeon processor.
Intel Programmable Acceleration Card with Intel Arria 10 GX FPGA	Intel PAC with Intel Arria 10 GX FPGA	PCIe* FPGA accelerator card. Contains an FPGA Interface Manager (FIM) that pairs with an Intel Xeon processor over the PCIe bus.

1.3. Acronyms

Table 3. Acronyms

Acronyms	Expansion	Description
AFU	Accelerator Functional Unit	Hardware Accelerator implemented in FPGA logic which offloads a computational operation for an application from the CPU to improve performance.
AF	Accelerator Function	Compiled Hardware Accelerator image implemented in FPGA logic that accelerates an application.
API	Application Programming Interface	A set of subroutine definitions, protocols, and tools for building software applications.
FIM	FPGA Interface Manager	The FPGA hardware containing the FPGA Interface Unit (FIU) and external interfaces for memory, networking, etc. The Accelerator Function (AF) interfaces with the FIM at run time.
OPAE	Open Programmable Acceleration Engine	The OPAE is a software framework for managing and accessing AFs.
RoT	Root of Trust	A source that can be trusted, such as the BMC in the Intel PAC.
BSP	Board Support Package	A typical Intel PAC BSP consists of software layers and a hardware project created using the Intel Quartus® Prime Pro Edition software that Intel FPGA SDK for OpenCL compiler stitches accelerator code into and compiles. The BSP resides in the AFU.

2. Setting Up the Host Machine

2.1. Introduction

This user guide describes how to get started with the OpenCL on the Intel PAC with Intel Arria 10 GX FPGA for 1.2.1 Release. The instructions use the precompiled OpenCL kernels included in this 1.2.1 Release. This user guide also includes a brief introduction to compiling OpenCL kernels.

This user guide describes how to get started with the OpenCL on the Intel FPGA Programmable Acceleration Card D5005. The instructions use the precompiled OpenCL kernels included in this version 2.0.1 Release. This user guide also includes a brief introduction to compiling OpenCL kernels.

OpenCL designs comprise two components, the kernel and the host. The kernel includes the accelerator code. The host runs on the host machine. The accelerator card plugs into the host machine.

Note: You must have root permission on the host machine to setup OpenCL.

Prerequisites: Before running OpenCL, you must follow the instructions from the *Getting Started* section of the *Intel Acceleration Stack Quick Start Guide for Intel Programmable Acceleration Card with Arria 10 GX FPGAIntel Acceleration Stack Quick Start Guide for Intel FPGA Programmable Acceleration Card D5005*, referred to as *Quick Start Guide* throughout this document.

Attention:

- If you need the OpenCL compiler and tools to build and run OpenCL AFUs, download and install the Intel Acceleration Stack for Development. Installing the development software ensures that the OpenCL SDK is available under`/home/<username>/inteldevstack/` or a Custom Directory, `/<custom Directory>`. This user guide refers to this path as `/<dev Install Path>`.
- If you only require the Intel FPGA SDK for the OpenCL deployment functionality, download and install the Intel Acceleration Stack for Runtime. Installing the runtime environment ensures that the OpenCL RTE is installed under `/home/<username>/intelrtestack/` or a Custom Directory, `/<custom Directory>`. This user guide refers to this path as `/<RTE Install Path>`.
- Do not install the RTE and the DEV on the same host system. The DEV already contains the RTE.

Related Information

- Intel FPGA SDK for Open Computing Language (OpenCL) web-page
- Intel FPGA SDK for OpenCL Pro Edition Getting Started Guide

- Intel Acceleration Stack Quick Start Guide for Intel Programmable Acceleration Card with Intel Arria 10 GX FPGA
- Getting Started
- Installing the Intel Acceleration Stack Development Package on the Host Machine
- Installing the Intel Acceleration Stack Runtime Package on the Host Machine
- OpenCL on the Intel PAC with Intel Arria 10 GX FPGA Quick Start User Guide Archives on page 25
- Intel Acceleration Stack Quick Start Guide for Intel FPGA Programmable Acceleration Card D5005
- Installing the Runtime Package on the Host Machine
- Installing the Development Package on the Host Machine

2.1.1. Release Content

Follow the installation instructions from the *Quick Start Guide* to set up the Intel PAC with Intel Arria 10 GX FPGA. The release available under $OPAE_PLATFORM_ROOT includes the files for the Intel PAC with Intel Arria 10 GX FPGA 1.2.1 Release. The release includes the following files for OpenCL located in the $OPAE_PLATFORM_ROOT/opencl folder:

- 1.2.1 OpenCL Board Support Package (BSP):
  - `opencl_bsp`
- OpenCL example designs tested with:
  - `exm_opencl_hello_world_x64_linux.tgz`
  - `exm_opencl_vector_add_x64_linux.tgz`
- Pre-compiled kernels <aocx>:
  - `hello_world.aocx`
  - `vector_add.aocx`

When you install the Intel Acceleration Stack package and run the initialization script, the environment variables are set. The $OPAE_PLATFORM_ROOT points to the extracted installation package. The release includes the following files for OpenCL located in the $OPAE_PLATFORM_ROOT/opencl folder:

- 2.0.1 OpenCL Board Support Package (BSP):
  - `opencl_bsp.tar.gz`
- OpenCL example designs tested with:
  - `exm_opencl_hello_world_x64_linux.tgz`
  - `exm_opencl_vector_add_x64_linux.tgz`
- Pre-compiled kernels <aocx>:
  - `hello_world.aocx`
  - `vector_add.aocx`

Related Information

Understanding the Extracted Intel PAC with Intel Arria 10 GX FPGA Release Package

2.2. Initializing the Environment for OpenCL with Intel Acceleration Stack

The `init_env.sh` script performs all the initialization and setup for the Acceleration Stack for OpenCL. The script is available in either `/<RTE install path>/` or `/<DEV install path>/`.

The script completes the following tasks:

- Exports the following environment variables:

Environment Variables	Description
OPAE_PLATFORM_ROOT	Points to the extracted Intel Acceleration Stack release.
AOCL_BOARD_PACKAGE_ROOT	Points to the unpacked OpenCL BSP.
INTELFPGAOCLSKDROOT	The Intel FPGA SDK for OpenCL installation directory.
ALTERAOCLSDKROOT	Builds and runs the OpenCL samples in the installation directory. *Note:* If you are using the Intel Acceleration Stack for Runtime Release 2.0.1 , please add the following line at the end of `init_env.sh` script: `export ALTERAOCLSDKROOT= $INTELFPGAOCLSDKROOT`
QUARTUS_HOME	Exported only if you are using the Intel Acceleration Stack for Development. Points to Intel Quartus Prime installation used for compiles. *Note:* The Acceleration Stack for Development includes Intel Quartus Prime software. Use this version for all your OpenCL development and compiles as FIM on the board is developed using this particular Intel Quartus Prime version. The `init_env.sh` points to this version by default.

- Runs the OpenCL initialization script to enable the runtime environment or the development environment (if installed) by running `init_opencl.sh`

  Note: If this is your first time running `init_env.sh`, you must restart and rerun the script for permanent permissions and system parameter settings to take effect.

  Note: Each time you restart the host or start a new shell, rerun the `init_env.sh` script. Most settings are temporary.

- Sets various permissions and system parameters by running `setup_permissions.sh`

- Adds the Intel SDK for OpenCL (`aocl`) utility located at `$INTELFPGAOCLSDKROOT/bin` to your PATH

  Note: Ensure that you install the FPGA driver as per the instructions in the Intel Acceleration Stack Quick Start Guide for Intel FPGA Programmable Acceleration Card D5005 Intel Acceleration Stack Quick Start Guide for Intel PAC with Intel Arria 10 GX FPGA and the `init_env.sh` script sources the `setup_permission.sh` script. You must execute `setup_permisssion.sh` script after every reboot. Intel recommends you to include it as part of the init_env.sh script.

Complete the following steps to run the OpenCL design:

1. Initialize the environment to use OpenCL and Intel Acceleration Stack:

```
source <RTE install path>/init_env.sh or <DEV install path>/init_env.sh
```

2. Install OpenCL drivers by running the following command:

```
aocl install $AOCL_BOARD_PACKAGE_ROOT
```

Note: You may have to run the above `source init_env.sh` command as root. This helps setting up the right environment to run `aocl install` as root after. This is the one time step. After the `.fcd` file is set, you don't need to run this command every time you try to compile or use the Intel PAC card.

Sample Output:

```
# aocl install
Do you want to setup the FCD at directory /opt/Intel/OpenCL/Boards [y/n] y
aocl install: Adding the board package /tools/inteldevstack/
a10_gx_pac_ias_1_2_1_pv/opencl/opencl_bsp to the list of installed packages
aocl install: Setting up the FPGA Client Driver (FCD) to the system.
Install the FCD file to /opt/Intel/OpenCL/Boards
Installing the board package driver to the system.
aocl install: Running install from /tools/inteldevstack/
a10_gx_pac_ias_1_2_1_pv/opencl/opencl_bsp/linux64/libexec
```

3. Check if you have two ICD files, `Altera.icd`, and `Intel_FPGA_SSG_Emulator.icd` loaded in your `/etc/OpenCL/vendors` directory when you run the stack installation. The ICD driver links the host against the OpenCL device libraries. If not installed at the required location, run the following commands to manually copy them:

```
sudo cp $INTELFPGAOCLSDKROOT/Altera.icd /etc/OpenCL/vendors/
sudo cp $INTELFPGAOCLSDKROOT/Intel_FPGA_SSG_Emulator.icd /etc/OpenCL/
vendors/
```

4. Enter the following command to see the boards connected to the host machine:

```
aocl diagnose
```

Note: Use `aocl diagnose <device-names>` to run diagnose for specified devices and use `aocl diagnose all` to run diagnose for all devices.

Sample Output:

```
# aocl diagnose
--------------------------------------------------------------------
ICD System Diagnostics
--------------------------------------------------------------------

Using the following location for ICD installation:
    /etc/OpenCL/vendors

Found 1 icd entry at that location:
    /etc/OpenCL/vendors/Altera.icd

the following OpenCL libraries are referenced in the icd files:
    libalteracl.so

checking LD_LIBRARY_PATH for registered libraries:
    libalteracl.so was registered on the system at /
<installation_directory>/tools/intelFPGA_pro/quartus_19.2.0b57/hld/host/
linux64/lib

Using the following location for fcd installations:
    /opt/Intel/OpenCL/Boards
```

```
Found 1 fcd entry at that location:
    /opt/Intel/OpenCL/Boards/dcp_bsp.fcd

the following OpenCL libraries are referenced in the fcd files:
    /<installation_directory>/inteldevstack/a10_gx_pac_ias_1_2_1_pv/opencl/
opencl_bsp/linux64/lib/libintel_opae_mmd.so

checking LD_LIBRARY_PATH for registered libraries:
    /<installation_directory>/inteldevstack/a10_gx_pac_ias_1_2_1_pv/opencl/
opencl_bsp/linux64/lib/libintel_opae_mmd.so was registered on the system.

Number of Platforms = 1
    1. Intel(R) FPGA SDK for OpenCL(TM)                             |
Intel(R) Corporation           | OpenCL 1.0 Intel(R) FPGA SDK for
OpenCL(TM), Version 19.4
--------------------------------------------------------------------
ICD diagnostics PASSED
--------------------------------------------------------------------
--------------------------------------------------------------------
BSP Diagnostics
--------------------------------------------------------------------
--------------------------------------------------------------------
Device Name:
acl0

BSP Install Location:
/<installation_directory>/inteldevstack/a10_gx_pac_ias_1_2_1_pv/opencl/
opencl_bsp

Vendor: Intel Corp

Physical Dev Name   Status            Information

pac_ef00000         Passed            Intel PAC Platform (pac_ef00000)
                                      PCIe 05:00.0
                                      FPGA temperature = 49 degrees C.

DIAGNOSTIC_PASSED
--------------------------------------------------------------------

Call "aocl diagnose <device-names>" to run diagnose for specified devices
Call "aocl diagnose all" to run diagnose for all devices
```

5. Program the required OpenCL configuration file from the host machine by typing the following command:

```
aocl program <device name> <filename>
```

Sample Output:

```
$ aocl program acl0 vector_add.aocx
aocl program: Running program from /home/DCPsupport/intelrtestack/
a10_gx_pac_ias_1_2_1_pv/opencl/opencl_bsp/linux64/libexec
Program succeed.
```

You can see the device name in the output for `aocl diagnose` and you can use one of the pre-compiled bitstream for programming. For example,

```
aocl program acl0 $OPAE_PLATFORM_ROOT/opencl/hello_world.aocx
```

2.3. Initializing the Intel Acceleration Stack for OpenCL in a Virtualized Environment

Follow the installation instructions from the *Quick Start Guide* to set up the Intel PAC with Intel Arria 10 GX FPGAIntel FPGA PAC D5005.

Once the stack setup is complete, follow the steps below to setup OpenCL on Intel FPGA PAC D5005.

You can run the OPAE software in a non-virtualized environment with the Single Root I/O Virtualization (SR-IOV) disabled or in a virtualized environment with the SR-IOV enabled.

To run the OpenCL reference design in a virtualized environment that includes SR-IOV, follow steps 1 to 3 from section Initializing the Environment for OpenCL with Intel Acceleration Stack on page 7 and then complete the following additional steps:

1. Program the required OpenCL configuration from the host machine by typing the following command:

   ```
   aocl program <device name> <filename>
   ```

   Sample Output:

   ```
   $ aocl program acl0 signed_vector_add.aocx
   aocl program: Running program from /home/DCPsupport/intelrtestack/
   a10_gx_pac_ias_1_2_1_pv/opencl/opencl_bsp/linux64/libexec
   Program succeed.
   ```

   You can see the device name in the output for `aocl diagnose` and you can use one of the pre-compiled bitstream for programming. For example,

   ```
   aocl program acl0 $OPAE_PLATFORM_ROOT/opencl/hello_world.aocx
   ```

   Note: The 1.2.1 Release does not allow partial reconfiguration in virtualized environment.

   Note: The 2.0.1 Release does not support partial reconfiguration in virtualized environment.

2. Enable virtualization using the instructions from section Updating Settings Required for VFs and section Configuring the VF Port on the Host of the *Quick Start Guide*.
3. Set the `CL_CONTEXT_COMPILER_MODE_INTELFPGA` environment variable in the virtual machine to disable FPGA configuration or reconfiguration during OpenCL host runtime:

   ```
   $ export CL_CONTEXT_COMPILER_MODE_INTELFPGA=3
   ```

4. Run the required application from the virtual machine.
5. Disable virtualization using the instruction from section Disconnecting the VF from the VM and Reconnecting to the PF of the *Quick Start Guide*.

Related Information

Intel Acceleration Stack Quick Start Guide for Intel Programmable Acceleration Card with Intel Arria 10 GX FPGA

3. Running Diagnostics

Before running diagnostics, load an OpenCL kernel to the board. The following instructions use the OpenCL kernel or you may also use your own.

1. Load OpenCL kernel:

```
$ aocl program acl0 vector_add.aocx
aocl program: Running program from /home/DCPsupport/intelrtestack/
a10_gx_pac_ias_1_2_1_pv/opencl/opencl_bsp/linux64/libexec
Program succeed.
```

   Sample program output:

```
aocl program: Running program from $OPAE_PLATFORM_ROOT/opencl/opencl_bsp \
/linux64/libexec
Program succeed.
```

2. Run the simple diagnostic utility:

```
$ aocl diagnose
```

   Sample diagnostic output:

```
--------------------------------------------------------------------
ICD System Diagnostics
--------------------------------------------------------------------

Using the following location for ICD installation:
    /etc/OpenCL/vendors

Found 1 icd entry at that location:
    /etc/OpenCL/vendors/Altera.icd

the following OpenCL libraries are referenced in the icd files:
    libalteracl.so

checking LD_LIBRARY_PATH for registered libraries:
    libalteracl.so was registered on the system at /
<installation_directory>/tools/intelFPGA_pro/quartus_19.2.0b57/hld/host/
linux64/lib

Using the following location for fcd installations:
    /opt/Intel/OpenCL/Boards

Found 1 fcd entry at that location:
    /opt/Intel/OpenCL/Boards/dcp_bsp.fcd

the following OpenCL libraries are referenced in the fcd files:
    /<installation_directory>/inteldevstack/a10_gx_pac_ias_1_2_1_pv/opencl/
opencl_bsp/linux64/lib/libintel_opae_mmd.so

checking LD_LIBRARY_PATH for registered libraries:
    /<installation_directory>/inteldevstack/a10_gx_pac_ias_1_2_1_pv/opencl/
opencl_bsp/linux64/lib/libintel_opae_mmd.so was registered on the system.

Number of Platforms = 1
    1. Intel(R) FPGA SDK for OpenCL(TM)                             |
Intel(R) Corporation           | OpenCL 1.0 Intel(R) FPGA SDK for
```

```
OpenCL(TM), Version 19.4
--------------------------------------------------------------------
ICD diagnostics PASSED
--------------------------------------------------------------------
--------------------------------------------------------------------
BSP Diagnostics
--------------------------------------------------------------------
--------------------------------------------------------------------
Device Name:
acl0

BSP Install Location:
/<installation_directory>/inteldevstack/a10_gx_pac_ias_1_2_1_pv/opencl/
opencl_bsp

Vendor: Intel Corp

Physical Dev Name   Status            Information

pac_ef00000         Passed            Intel PAC Platform (pac_ef00000)
                                      PCIe 05:00.0
                                      FPGA temperature = 49 degrees C.

DIAGNOSTIC_PASSED
--------------------------------------------------------------------
```

3. Run the advanced diagnostic:

```
$ aocl diagnose acl0
```

Sample advanced diagnostic output:

```
--------------------------------------------------------------------
ICD System Diagnostics
--------------------------------------------------------------------
Using the following location for ICD installation:
    /etc/OpenCL/vendors

Found 1 icd entry at that location:
    /etc/OpenCL/vendors/Altera.icd

the following OpenCL libraries are referenced in the icd files:
    libalteracl.so

checking LD_LIBRARY_PATH for registered libraries:
    libalteracl.so was registered on the system at /storage/shared/
home_directories/homichel/tools/intelFPGA_pro/quartus_19.2.0b57/hld/host/
linux64/lib

Using the following location for fcd installations:
    /opt/Intel/OpenCL/Boards

Found 1 fcd entry at that location:
    /opt/Intel/OpenCL/Boards/dcp_bsp.fcd

the following OpenCL libraries are referenced in the fcd files:
    /storage/shared/home_directories/homichel/inteldevstack/
a10_gx_pac_ias_1_2_1_pv/opencl/opencl_bsp/linux64/lib/libintel_opae_mmd.so

checking LD_LIBRARY_PATH for registered libraries:
    /storage/shared/home_directories/homichel/inteldevstack/
a10_gx_pac_ias_1_2_1_pv/opencl/opencl_bsp/linux64/lib/libintel_opae_mmd.so
was registered on the system.

Number of Platforms = 1
    1. Intel(R) FPGA SDK for OpenCL(TM)                             |
Intel(R) Corporation           | OpenCL 1.0 Intel(R) FPGA SDK for
OpenCL(TM), Version 19.4
--------------------------------------------------------------------
ICD diagnostics PASSED
```

```
--------------------------------------------------------------------
--------------------------------------------------------------------
BSP Diagnostics
--------------------------------------------------------------------
Using platform: Intel(R) FPGA SDK for OpenCL(TM)
Using Device with name: pac_a10 : Intel PAC Platform (pac_ef00000)
Using Device from vendor: Intel Corp
clGetDeviceInfo CL_DEVICE_GLOBAL_MEM_SIZE = 8589933568
clGetDeviceInfo CL_DEVICE_MAX_MEM_ALLOC_SIZE = 8589933568
Allocated 8589933568 bytes
Actual maximum buffer size = 8589933568 bytes
Writing 8191 MB to global memory ...
Allocated 1073741824 Bytes host buffer for large transfers
Write speed: 5843.54 MB/s [5689.33 -> 6183.29]
Reading and verifying 8191 MB from global memory ...
Read speed: 6663.87 MB/s [6523.73 -> 6729.59]
Successfully wrote and readback 8191 MB buffer

Transferring 262144 KBs in 512 512 KB blocks ... 3525.69 MB/s
Transferring 262144 KBs in 256 1024 KB blocks ... 3734.74 MB/s
Transferring 262144 KBs in 128 2048 KB blocks ... 3131.35 MB/s
Transferring 262144 KBs in 64 4096 KB blocks ... 4269.86 MB/s
Transferring 262144 KBs in 32 8192 KB blocks ... 4801.54 MB/s
Transferring 262144 KBs in 16 16384 KB blocks ... 5177.86 MB/s
Transferring 262144 KBs in 8 32768 KB blocks ... 5413.49 MB/s
Transferring 262144 KBs in 4 65536 KB blocks ... 5627.71 MB/s
Transferring 262144 KBs in 2 131072 KB blocks ... 6849.71 MB/s
Transferring 262144 KBs in 1 262144 KB blocks ... 6895.99 MB/s

As a reference:
PCIe Gen1 peak speed: 250MB/s/lane
PCIe Gen2 peak speed: 500MB/s/lane
PCIe Gen3 peak speed: 985MB/s/lane

Writing 262144 KBs with block size (in bytes) below:

Block_Size Avg    Max    Min    End-End (MB/s)
  524288 3073.65 3525.69 1750.25 2440.57
 1048576 2564.51 3734.74 1979.41 2208.89
 2097152 2437.80 2701.91 2240.24 2265.86
 4194304 3391.19 4220.46 2581.52 3236.38
 8388608 4225.16 4700.36 3138.08 4052.50
16777216 4419.41 4961.91 3436.53 4343.19
33554432 4775.78 5413.49 3895.03 4728.55
67108864 4889.22 5497.75 4160.56 4860.37
134217728 6233.47 6242.94 6224.03 6218.41
268435456 6218.76 6218.76 6218.76 6218.76

Reading 262144 KBs with block size (in bytes) below:

Block_Size Avg    Max    Min    End-End (MB/s)
  524288 3187.99 3457.18 2448.31 2631.75
 1048576 3077.55 3332.59 2491.45 2480.11
 2097152 2835.68 3131.35 1976.36 2540.81
 4194304 3675.78 4269.86 2552.75 3447.85
 8388608 4203.11 4801.54 2851.31 4056.30
16777216 4488.70 5177.86 3125.90 4398.89
33554432 4658.78 5391.93 3216.03 4608.17
67108864 4896.32 5627.71 3927.64 4870.05
134217728 6735.61 6849.71 6625.26 6726.29
268435456 6895.99 6895.99 6895.99 6895.99

Write top speed = 6242.94 MB/s
Read top speed = 6895.99 MB/s
Throughput = 6569.46 MB/s

DIAGNOSTIC_PASSED
```

4. OpenCL Support for Multi-Card Systems

Before running an OpenCL application, program the Intel PAC with an Accelerator Function (AF) that includes the BSP logic. Use the `aocl` program command to load an `aocx` file to the Intel PAC. It is only necessary to program the AF one time per Intel PAC. After the initial programming, you can use the OpenCL API to load different applications to the Intel PAC using the `aocx` program command.

Note: For a system with one Intel PAC, Intel recommends that you allocate the number of hugepages to 20. If your system has multiple Intel PACs, you must allocate 20 hugepages per card. For example, a system with four Intel PAC requires of total 80 hugepages.

To set the hugepages to 80, enter the following command:

```
$ sudo sh -c "echo 80 > /sys/kernel/mm/hugepages/hugepages-2048kB \
/nr_hugepages"
```

Run the `aocl diagnose` command to determine how many FPGAs the system includes. For example, running the `aocl diagnose` command on a system with two Intel PAC might show output similar to the following:

1. `$ aocl diagnose`

```
--------------------------------------------------------------------
ICD System Diagnostics
--------------------------------------------------------------------

Using the following location for ICD installation:
/etc/OpenCL/vendors

Found 2 icd entry at that location:
/etc/OpenCL/vendors/Altera.icd
/etc/OpenCL/vendors/Intel_FPGA_SSG_Emulator.icd

the following OpenCL libraries are referenced in the icd files:
libalteracl.so
libintelocl.so

checking LD_LIBRARY_PATH for registered libraries:
libalteracl.so was registered on the system at /tools/quartus/hld/host/
linux64/lib
libintelocl.so was registered on the system at /tools/quartus/hld/linux64/
lib

Using the following location for fcd installations:
/opt/Intel/OpenCL/Boards

Found 1 fcd entry at that location:
/opt/Intel/OpenCL/Boards/dcp_bsp.fcd

the following OpenCL libraries are referenced in the fcd files:
/tools/inteldevstack/a10_gx_pac_ias_1_2_1_pv/opencl/opencl_bsp/linux64/lib/
libintel_opae_mmd.so
```

```
checking LD_LIBRARY_PATH for registered libraries:
/tools/inteldevstack/a10_gx_pac_ias_1_2_1_pv/opencl/opencl_bsp/linux64/lib/
libintel_opae_mmd.so was registered on the system.

Number of Platforms = 2
1. Intel(R) FPGA SDK for OpenCL(TM)                             | Intel(R)
Corporation           | OpenCL 1.0 Intel(R) FPGA SDK for OpenCL(TM),
Version 19.4
2. Intel(R) FPGA Emulation Platform for OpenCL(TM) (preview)    | Intel(R)
Corporation           | OpenCL 1.0 Intel(R) FPGA SDK for OpenCL(TM),
Version 19.4
--------------------------------------------------------------------
ICD diagnostics PASSED
--------------------------------------------------------------------
--------------------------------------------------------------------
BSP Diagnostics
--------------------------------------------------------------------
--------------------------------------------------------------------
Device Name:
acl0

BSP Install Location:
/home/DCPsupport/intelrtestack/a10_gx_pac_ias_1_2_1_pv/opencl/opencl_bsp

Vendor: Intel Corp

Physical Dev Name   Status            Information

pac_ec00000         Uninitialized     OpenCL BSP not loaded. Must load BSP
using command:
                                      'aocl program <device_name>
<aocx_file>'
                                      before running OpenCL programs using
this device

DIAGNOSTIC_PASSED
--------------------------------------------------------------------
--------------------------------------------------------------------
Device Name:
acl1

BSP Install Location:
/tools/inteldevstack/a10_gx_pac_ias_1_2_1_pv/opencl/opencl_bsp

Vendor: Intel Corp

Physical Dev Name   Status            Information

pac_ec00000         Uninitialized     OpenCL BSP not loaded. Must load BSP
using command:
                                      'aocl program <device_name>
<aocx_file>'
                                      before running OpenCL programs using
this device

DIAGNOSTIC_PASSED
--------------------------------------------------------------------
```

2. The following command programs the first card listed in Step 1:

   ```
   $ aocl program acl0 $OPAE_PLATFORM_ROOT/opencl/
   hello_world.aocx
   ```

   ```
   aocl program: Running program from $OPAE_PLATFORM_ROOT/opencl \
   /opencl_bsp

   Program succeed.
   ```

3. The following command programs the second card listed in Step 1:

```
$ aocl program acl1 $OPAE_PLATFORM_ROOT/opencl/
hello_world.aocx
```

```
aocl program: Running program from $OPAE_PLATFORM_ROOT/opencl \
/opencl_bsp

Program succeed.
```

4. After programming the FPGAs, the `aocl diagnose` command provides information about them:

```
$ aocl diagnose
```

```
--------------------------------------------------------------------
ICD System Diagnostics
--------------------------------------------------------------------

Using the following location for ICD installation:
/etc/OpenCL/vendors

Found 2 icd entry at that location:
/etc/OpenCL/vendors/Altera.icd
/etc/OpenCL/vendors/Intel_FPGA_SSG_Emulator.icd

the following OpenCL libraries are referenced in the icd files:
libalteracl.so
libintelocl.so

checking LD_LIBRARY_PATH for registered libraries:
libalteracl.so was registered on the system at /tools/quartus/hld/host/
linux64/lib
libintelocl.so was registered on the system at /tools/quartus/hld/linux64/
lib

Using the following location for fcd installations:
/opt/Intel/OpenCL/Boards

Found 1 fcd entry at that location:
/opt/Intel/OpenCL/Boards/dcp_bsp.fcd

the following OpenCL libraries are referenced in the fcd files:
/tools/inteldevstack/a10_gx_pac_ias_1_2_1_pv/opencl/opencl_bsp/linux64/lib/
libintel_opae_mmd.so

checking LD_LIBRARY_PATH for registered libraries:
/tools/inteldevstack/a10_gx_pac_ias_1_2_1_pv/opencl/opencl_bsp/linux64/lib/
libintel_opae_mmd.so was registered on the system.

Number of Platforms = 1
1. Intel(R) FPGA SDK for OpenCL(TM)                             | Intel(R)
Corporation           | OpenCL 1.0 Intel(R) FPGA SDK for OpenCL(TM),
Version 19.4
--------------------------------------------------------------------
ICD diagnostics PASSED
--------------------------------------------------------------------
--------------------------------------------------------------------
BSP Diagnostics
--------------------------------------------------------------------
--------------------------------------------------------------------
Device Name:
acl0

BSP Install Location:
/home/DCPsupport/intelrtestack/a10_gx_pac_ias_1_2_1_pv/opencl/opencl_bsp

Vendor: Intel Corp

Physical Dev Name   Status            Information
```

```
pac_ec00000         Passed            Intel PAC Platform (pac_ec00000)
                                      PCIe 134:00.0
                                      FPGA temperature = 42 degrees C.

DIAGNOSTIC_PASSED
--------------------------------------------------------------------
--------------------------------------------------------------------
Device Name:
acl1

BSP Install Location:
/tools/inteldevstack/a10_gx_pac_ias_1_2_1_pv/opencl/opencl_bsp

Vendor: Intel Corp

Physical Dev Name   Status            Information

pac_ec00000         Passed            Intel PAC Platform (pac_ec00000)
                                      PCIe 134:00.0
                                      FPGA temperature = 42 degrees C.

DIAGNOSTIC_PASSED
--------------------------------------------------------------------
```

Note: You can run the advanced diagnostic on any specific device in your multi-card system using the following command:

```
$ aocl diagnose <device name>
```

5. Running Samples

This section describes how to compile and run the host code for the provided samples using the precompiled OpenCL kernels.

5.1. Running Hello World

1. Extract hello_world example:

```
cd $OPAE_PLATFORM_ROOT/opencl

mkdir exm_opencl_hello_world_x64_linux

cd exm_opencl_hello_world_x64_linux

tar xf ../exm_opencl_hello_world_x64_linux.tgz
```

2. Build example:

```
cd hello_world

make
```

3. Copy aocx to example bin folder:

```
cp $OPAE_PLATFORM_ROOT/opencl/hello_world.aocx ./bin/
```

4. Program the aocx file:

```
aocl program acl0 ./bin/hello_world.aocx
```

Note: This step is not necessary for non SR-IOV enabled system as OpenCL by default performs partial reconfiguration to program the new kernel on FPGA. The aocx file in the release OpenCL folder is unsigned. If you are using Intel PAC features and programming a signed aocx file to the device, ensure that you copy the same file in the bin before running.

5. Run example:

```
./bin/host
```

Example sample output:

```
Querying platform for info:
==========================
CL_PLATFORM_NAME                         = Intel(R) FPGA SDK for OpenCL(TM)
CL_PLATFORM_VENDOR                       = Intel(R) Corporation
CL_PLATFORM_VERSION                      = OpenCL 1.0 Intel(R) FPGA SDK for
OpenCL(TM), Version 19.4

Querying device for info:
========================
CL_DEVICE_NAME                           = pac_a10 : Intel PAC Platform
(pac_ef00000)
CL_DEVICE_VENDOR                         = Intel Corp
```

```
CL_DEVICE_VENDOR_ID                      = 4466
CL_DEVICE_VERSION                        = OpenCL 1.0 Intel(R) FPGA SDK for
OpenCL(TM), Version 19.4
CL_DRIVER_VERSION                        = 19.4
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                = 8589933568
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       = 2147483392
CL_DEVICE_MAX_WORK_ITEM_DIMENSIONS       = 3
CL_DEVICE_MEM_BASE_ADDR_ALIGN            = 8192
CL_DEVICE_MIN_DATA_TYPE_ALIGN_SIZE       = 1024
CL_DEVICE_PREFERRED_VECTOR_WIDTH_CHAR    = 4
CL_DEVICE_PREFERRED_VECTOR_WIDTH_SHORT   = 2
CL_DEVICE_PREFERRED_VECTOR_WIDTH_INT     = 1
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?              = true
Command queue profiling enabled?         = true
Using AOCX: hello_world.aocx

Kernel initialization is complete.
Launching the kernel...

Thread #2: Hello from Altera's OpenCL Compiler!

Kernel execution is complete.
```

```
Querying platform for info:
==========================
CL_PLATFORM_NAME                         = Intel(R) FPGA SDK for OpenCL(TM)
CL_PLATFORM_VENDOR                       = Intel(R) Corporation
CL_PLATFORM_VERSION                      = OpenCL 1.0 Intel(R) FPGA SDK for
OpenCL(TM), Version 19.2

Querying device for info:
========================
CL_DEVICE_NAME                           = pac_s10_dc : Intel PAC Platform
(pac_ec00001)
CL_DEVICE_VENDOR                         = Intel Corp
CL_DEVICE_VENDOR_ID                      = 4466
CL_DEVICE_VERSION                        = OpenCL 1.0 Intel(R) FPGA SDK for
OpenCL(TM), Version 19.2
CL_DRIVER_VERSION                        = 19.2
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                = 34359737344
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       = 8589934336
CL_DEVICE_MAX_WORK_ITEM_DIMENSIONS       = 3
CL_DEVICE_MEM_BASE_ADDR_ALIGN            = 8192
CL_DEVICE_MIN_DATA_TYPE_ALIGN_SIZE       = 1024
CL_DEVICE_PREFERRED_VECTOR_WIDTH_CHAR    = 4
CL_DEVICE_PREFERRED_VECTOR_WIDTH_SHORT   = 2
CL_DEVICE_PREFERRED_VECTOR_WIDTH_INT     = 1
```

```
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
Using AOCX: hello_world.aocx

Kernel initialization is complete.
Launching the kernel...

Thread #2: Hello from Altera's OpenCL Compiler!

Kernel execution is complete.
```

5.2. Running Vector Add

1. Extract example:

   ```
   cd $OPAE_PLATFORM_ROOT/opencl

   mkdir exm_opencl_vector_add_x64_linux

   cd exm_opencl_vector_add_x64_linux

   tar xzvf ../exm_opencl_vector_add_x64_linux.tgz
   ```

2. Build example:

   ```
   cd vector_add

   make
   ```

3. Copy precompiled OpenCL kernel to bin folder:

   ```
   cp $OPAE_PLATFORM_ROOT/opencl/vector_add.aocx ./bin
   ```

4. Program the aocx file:

   ```
   aocl program acl0 ./bin/vector_add.aocx
   ```

   Note: This step is not necessary for non SR-IOV enabled system as OpenCL by default performs partial reconfiguration to program the new kernel on FPGA. The `aocx` file in the release OpenCL folder is unsigned. If you are using Intel PAC features and programming a signed `aocx` file to the device, ensure that you copy the same file in the `bin` before running. Also, make sure that you rename the file to `vector_add.aocx` because the hose code looks for the specific name of the kernel.

   Example sample output:

   ```
   Running program from /home/<username>/intelrtestack/d5005_ias_2_0_1_b237/
   opencl/opencl_bsp/linux64/libexec
   Program succeed.
   ```

5. Run example:

   ```
   ./bin/host
   ```

   Example sample output:

   ```
   Initializing OpenCL
   Platform: Intel(R) FPGA SDK for OpenCL(TM)
   Using 1 device(s)
         pac_a10 : Intel PAC Platform (pac_ec00000)
   ```

```
Using AOCX: vector_add.aocx
Launching for device 0 (1000000 elements)

    Time: 227.021 ms
    Kernel time (device 0): 218.247 ms


    Verification: PASS
```

```
Initializing OpenCL
Platform: Intel(R) FPGA SDK for OpenCL(TM)
Using 2 device(s)
  pac_s10_dc : Intel PAC Platform (pac_ee00001)
  pac_s10_dc : Intel PAC Platform (pac_ee00000)
Using AOCX: vector_add.aocx
Launching for device 0 (500000 elements)
Launching for device 1 (500000 elements)

Time: 6.814 ms
Kernel time (device 0): 1.817 ms
Kernel time (device 1): 2.094 ms

Verification: PASS
```

6. Compiling OpenCL Kernels

Before compiling an OpenCL kernel, you must install the Intel Acceleration Stack for Development.

1. Set the user environment variable using one of the following commands:

```
source <DEV Install Path>/init_env.sh
```

2. Ensure that the environment is setup with correct BSP using the following command:

```
aoc -list-boards
```

Example sample output:

```
Board list:
pac_a10
Board Package: /home/username/inteldevstack/opencl_bsp
```

```
Board list:
pac_s10_dc
Board Package: /tools/<username>/dcp_2_0_1_PV_RC2/inteldevstack/
d5005_ias_2_0_1_b237/opencl/opencl_bspBoard list:
```

3. Compile an OpenCL Kernel to an aocx using commands similar to the following:

```
cd $OPAE_PLATFROM_ROOT/opencl/exm_opencl_vector_add_x64_linux/vector_add

aoc device/vector_add.cl -o bin/vector_add.aocx -board pac_a10
```

```
cd $OPAE_PLATFROM_ROOT/opencl/vector_add

aoc device/vector_add.cl -o bin/vector_add.aocx -board pac_s10_dc
```

4. The security features of the Intel PAC with Intel Arria 10 GX FPGA requires the signing of metadata in a bitstream even if you don't intend to use the security features of the card. If you try to program the aocx file as is generated after the compilation in the above step, the PACSign tool gives the following error message:

```
$ aocl program acl0 hello_world.aocx
aocl program: Running program from /home/DCPsupport/inteldevstack_1_2_1_pv/
a10_gx_pac_ias_1_2_1_pv/opencl/opencl_bsp/linux64/libexec
libopae-c reconf.c:427:fpgaReconfigureSlot() **ERROR** : Failed to
reconfigure bitstream: Input/output error
libopae-c reconf.c:450:fpgaReconfigureSlot() **ERROR** : PR incompatible
bitstream error detected
Error writing bitstream to FPGA: reconfiguration error
Error programming device
aocl program: Program failed
```

5. If you are in the development process or have a new Intel PAC with Intel Arria 10 GX FPGA card with no root entry hash programmed to it, follow the instructions from this section Example: Creating an Unsigned .aocx File Using OpenSSL Manager to create an unsigned aocx file with only signing metadata.
6. If you have Intel PAC with Intel Arria 10 GX FPGA with the root entry hash programmed, follow the steps from section Creating the OpenCL Bitstream to create signed aocx file with your preferred method.
7. After you follow the steps above, depending on the condition of your Intel PAC with Intel Arria 10 GX FPGA, you should be able to successfully program the signed or unsigned aocx file by using the following command:

   ```
   aocl program acl0 <unsigned_file.aocx>
   ```

Related Information

- Setting the Intel FPGA SDK for OpenCL User Environment Variables
- Intel PAC with Intel Arria 10 GX FPGA Security for OpenCL Applications on page 24

6.1. Checking Timing Results

Intel recommends that you check for timing failures after compilation of the kernel file.

Check the compilation directory for the presence of the following report files:

```
afu_default.failing_clocks.rpt
```

```
afu_default.failing_paths.rpt
```

For example, after compiling vector_add.cl, locate the $OPAE_PLATFORM_ROOT/opencl/vector_add/bin/vector_add directory. If there is a timing violation, this directory contains the failing report files. The failing report files indicate that the timing is not clean and the functional correctness cannot be guaranteed.

If OpenCL kernel compilation results in timing violations, Intel recommends to retry compilation with a different seed (aoc *<kernel.cl>* -seed=<*integer_value*>).

For example,

```
aoc device/vector_add.cl -seed=2 -o bin/vector_add.aocx -board pac_s10_dc
aoc device/vector_add.cl -seed=3 -o bin/vector_add.aocx -board pac_s10_dc
aoc device/vector_add.cl -seed=63 -o bin/vector_add.aocx -board pac_s10_dc
```

7. Intel PAC with Intel Arria 10 GX FPGA Security for OpenCL Applications

The Intel PAC with Intel Arria 10 GX FPGA allows you to enable security features such as Root of Trust (RoT) and AFU signing for the designs to be loaded onto it. The Intel PAC with Intel Arria 10 GX FPGA containing a hardware image (FIM) version 1.2.1 or greater requires an AFU to have the prepended signature blocks, even if an AFU root entry hash has not been programmed. To determine the FIM version installed on your board, follow the instructions from the section Identifying the Flash Image and BMC Firmware. Intel's PAC sign allows you to prepend the required blocks with an empty signature chain. Please refer to the Intel FPGA PAC Security Guide to see how to create the keys. After you create the keys, follow the instructions from section Signing OpenCL Images to create signed or unsigned OpenCL images. If you have any version of FIM prior to 1.2.1, you need to perform a one-time secure update to enable the security features.

Related Information

Security User Guide: Intel Programmable Acceleration Card with Intel Arria 10 GX FPGA

8. OpenCL on the Intel PAC with Intel Arria 10 GX FPGA Quick Start User Guide Archives

Intel Acceleration Stack Version	User Guide (PDF)
1.2	OpenCL on the Intel PAC with Intel Arria 10 GX FPGA Quick Start User Guide
1.1	OpenCL on the Intel PAC with Intel Arria 10 GX FPGA Quick Start User Guide
1.0	OpenCL on the Intel PAC with Intel Arria 10 GX FPGA Quick Start User Guide

9. Document Revision History for OpenCL Quick Start User Guide Intel Programmable Acceleration Card with Intel Arria 10 GX FPGA

Document Version	Intel Acceleration Stack Version	Changes
2020.07.21	1.2.1 (supported with Intel Quartus Prime Pro Edition Edition 19.2)	Added a link to the KDB in section *Intel FPGA PAC D5005 Security for OpenCL Applications* to clarify how to automate the signing process for *aocx* file.
2020.06.05	1.2.1 (supported with Intel Quartus Prime Pro Edition Edition 19.2)	Added steps to program the signed or unsigned `aocx` file in section *Compiling OpenCL Kernels*.
2020.0306	1.2.1 (supported with Intel Quartus Prime Pro Edition Edition 19.2)	• Added new section *Intel PAC with Intel Arria 10 GX FPGA Security for OpenCL Applications*. • Added two new terms BMC and RoT to *Table: Acronyms*. • Rearranged and moved the content of the *Introduction* chapter to *Setting up the Host Machine* chapter. • Changed the OpenCL RTE version to 19.4. • Added section *Initializing the Intel Acceleration Stack for OpenCL in a Virtualized Environment*. • Removed the following sections: — *Running an OpenCL Design Example* — *Disabling Non-Uniform Memory Access (NUMA) and DMA Worker Threads to Optimize PCIe Bandwidth*
2020.01.02	2.0.1 (supported with Intel Quartus Prime Pro Edition Edition 19.2)	• Updated a note in section *Initializing the Environment for OpenCL with Intel Acceleration Stack* to clarify the use of `aocl install` command. • Fixed typos.
2019.11.18	2.0.1 (supported with Intel Quartus Prime Pro Edition Edition 19.2)	• Added new section *Intel FPGA PAC D5005 Security for OpenCL Applications*. • Added two new terms BMC and RoT to *Table: Acronyms*. • Added missing steps in section *Initializing the Intel Acceleration Stack for OpenCL in a Virtualized Environment*. • Modified instructions in section *OpenCL Support for Multi-Card Systems*. • Added `aocl program` command in the following sections: — *Running Hello World* — *Running Vector Add*

continued...

Document Version	Intel Acceleration Stack Version	Changes
		• Updated sample output in the following sections: — *Initializing the Intel Acceleration Stack for OpenCL in a Virtualized Environment* — *Running Diagnostic* — *OpenCL Support for Multi-Card Systems* — *Running Samples* — *Compiling OpenCL Kernels* • Modified command to set the user environment variable in section *Compiling OpenCL Kernels*. • Removed section *Disabling Non-Uniform Memory Access (NUMA) and DMA Worker Threads to Optimize PCIe Bandwidth*.
2019.10.02	2.0 (supported with Intel Quartus Prime Pro Edition Edition 18.1.2)	• Added missing exports in section *Initializing the Intel Acceleration Stack for OpenCL*. • Modified command to set the hugepages in section *OpenCL Support for Multi-Card Systems*. • Added `aocl program` step in section *Running Hello World*. • Made the following changes in section *Checking Timing Results*: — Corrected the report file names. — Modified the command to do compilation with a different seed.
2019.08.05	2.0 (supported with Intel Quartus Prime Pro Edition Edition 18.1.2)	Initial release.
2019.06.28	1.2 (supported with Intel Quartus Prime Pro Edition 17.1.1)	• Added a step to program the aocx file in section *Running Vector Add*. • Modified command to set the user environment variable in section *Compiling OpenCL Kernels*.
2018.12.04	1.2 (supported with Intel Quartus Prime Pro Edition 17.1.1)	Simplified the installation process by including more commands in the `init_env.sh` script and using <*RTE Install Path*> and <*DEV Install Path*> for the installation paths when appropriate.
2018.11.02	1.1 (supported with Intel Quartus Prime Pro Edition 17.1.1)	Added *Configuring the OpenCL Driver* topic to the *Setting up the Host Machine* chapter.
2018.08.06	1.1 (supported with Intel Quartus Prime Pro Edition 17.1.1)	Initial release.