



Intel® FPGA SDK for OpenCL™ Pro Edition

Getting Started Guide

Updated for Intel® Quartus® Prime Design Suite: **22.4**



Online Version



Send Feedback

ID: **683188**

UG-OCL001

Version: **2022.12.19**

Contents

1. Intel® FPGA SDK for OpenCL™ Pro Edition Getting Started Guide.....	4
1.1. Prerequisites for the Intel FPGA SDK for OpenCL Pro Edition.....	5
1.1.1. GCC Requirement.....	7
1.2. Intel FPGA SDK for OpenCL Pro Edition and OpenCL BSP Backwards Compatibility.....	10
1.3. Contents of the Intel FPGA SDK for OpenCL Pro Edition.....	11
1.4. Overview of the Intel FPGA SDK for OpenCL Pro Edition Setup Process.....	12
2. Getting Started with the Intel FPGA SDK for OpenCL Pro Edition for Windows.....	15
2.1. Downloading the Intel FPGA SDK for OpenCL Pro Edition	15
2.2. Installing the Intel FPGA SDK for OpenCL Pro Edition	16
2.3. Setting the Intel FPGA SDK for OpenCL Pro Edition User Environment Variables.....	17
2.4. Verifying Software Installation.....	18
2.5. Installing an FPGA Board.....	19
2.5.1. Enabling Backwards Compatibility for an FPGA Board.....	20
2.6. Verifying Host Runtime Functionality via Emulation.....	21
2.6.1. Accessing an OpenCL Design Example.....	22
2.6.2. Compiling a Kernel for Emulation.....	22
2.6.3. Building the Host Application.....	22
2.6.4. Emulating Your OpenCL Kernel.....	24
2.7. Creating the FPGA Hardware Configuration File of an OpenCL Kernel.....	24
2.8. Updating the Hardware Image on the FPGA.....	27
2.8.1. Querying the Device Name of Your FPGA Board.....	27
2.8.2. Programming the Flash Memory of an FPGA.....	28
2.9. Executing an OpenCL Kernel on an FPGA.....	29
2.9.1. Running the Host Application.....	29
2.9.2. Output from Successful Kernel Execution.....	29
2.10. Uninstalling an FPGA Board	30
2.11. Uninstalling the Software.....	30
3. Getting Started with the Intel FPGA SDK for OpenCL Pro Edition for Linux.....	32
3.1. Downloading the Intel FPGA SDK for OpenCL Pro Edition.....	32
3.2. Installing the Intel FPGA SDK for OpenCL Pro Edition.....	33
3.3. Setting the Intel FPGA SDK for OpenCL Pro Edition User Environment Variables.....	34
3.4. Verifying Software Installation.....	35
3.5. Installing an FPGA Board.....	35
3.5.1. Enabling Backwards Compatibility for an FPGA Board.....	37
3.6. Verifying Host Runtime Functionality via Emulation.....	38
3.6.1. Accessing an OpenCL Design Example.....	38
3.6.2. Compiling a Kernel for Emulation.....	38
3.6.3. Building the Host Application.....	39
3.6.4. Emulating Your OpenCL Kernel.....	39
3.7. Creating the FPGA Hardware Configuration File of an OpenCL Kernel.....	40
3.8. Updating the Hardware Image on the FPGA.....	42
3.8.1. Querying the Device Name of Your FPGA Board.....	43
3.8.2. Programming the Flash Memory of an FPGA.....	43
3.9. Executing an OpenCL Kernel on an FPGA.....	44
3.9.1. Running the Host Application.....	44
3.9.2. Output from Successful Kernel Execution.....	45

3.10. Uninstalling an FPGA Board.....	46
3.11. Uninstalling the Software.....	46
A. Intel FPGA SDK for OpenCL Pro Edition Getting Started Guide Archives.....	47
B. Document Revision History of the Intel FPGA SDK for OpenCL Pro Edition Getting Started Guide.....	48

1. Intel® FPGA SDK for OpenCL™ Pro Edition Getting Started Guide

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

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

The Intel FPGA SDK for OpenCL Pro Edition provides a compiler and tools for you to build and run OpenCL applications that target Intel FPGA products. The Intel FPGA SDK for OpenCL Pro Edition supports the embedded profile of the OpenCL Specification version 1.0.

Tip: If you are looking for instructions to get started with oneAPI toolkits, use the following links:

- [Intel oneAPI Base Toolkit](#): Includes the Intel oneAPI DPC++/C++ Compiler and provides tools and libraries for developing high-performance, data-centric applications across diverse architectures.
- [Intel FPGA Add-on for oneAPI Base Toolkit](#): Main web page for Intel FPGA development flow for oneAPI.
- [Intel DevCloud](#): A cloud-based sandbox to actively prototype and experiment with workloads on Intel hardware.
- [Intel oneAPI Toolkits Installation Guides](#): Provides instructions for installing the toolkits.
- [Get Started with Intel oneAPI Base Toolkit on the DevCloud](#): Provides instructions to get started with the Intel DevCloud development sandbox.
- [Get Started with the Intel oneAPI Base Toolkit for Linux*](#): Provides Linux-specific getting started instructions.
- [Get Started with the Intel oneAPI Base Toolkit for Windows*](#): Provides Windows-specific getting started instructions.

⁽¹⁾ The Intel FPGA SDK for OpenCL is based on a published Khronos Specification, and has passed the Khronos Conformance Testing Process. Current conformance status can be found at www.khronos.org/conformance.

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

- Attention:**
- If you only require the Intel FPGA SDK for OpenCL's kernel deployment functionality, download and install the Intel FPGA Runtime Environment (RTE) for OpenCL. Refer to the *Intel FPGA RTE for OpenCL Pro Edition Getting Started Guide* for more information.

Do not install the SDK and the RTE on the same host system. The SDK already contains the RTE.

- If you want to use the Intel FPGA SDK for OpenCL Pro Edition with the Intel Arria® 10 GX FPGA Development Kit, refer to the Application Note *Configuring the Intel Arria 10 GX FPGA Development Kit for the Intel FPGA SDK for OpenCL* for more information.
- If you want to use the Intel FPGA SDK for OpenCL Pro Edition with the Intel Stratix® 10 GX Development Kit, refer to *Intel Stratix 10 Development Kit Initialization guide* (bringup guide included in the <path_to_BSP>/s10_ref/bringup directory) and *Intel Stratix 10 GX FPGA Development Kit User Guide* for more information.

Related Information

- [OpenCL Reference Pages](#)
- [OpenCL Specification version 1.0](#)
- [Intel FPGA RTE for OpenCL Pro Edition Getting Started Guide](#)
- [Configuring the Intel Arria 10 GX FPGA Development Kit for the Intel FPGA SDK for OpenCL](#)
- [Intel Stratix 10 GX FPGA Development Kit User Guide](#)

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

To install the Intel FPGA SDK for OpenCL Pro Edition and create an OpenCL application for an Intel FPGA Preferred Board for OpenCL, your system must meet certain hardware, target platform, and software requirements.

Hardware Requirements

Accelerator boards requirements:

- Acquire a Reference Platform from Intel, or a Custom Platform from an Intel preferred board vendor.

For more information, refer to the [Intel FPGA SDK for OpenCL FPGA Platforms](#) page on the Intel FPGA website.

Development system requirements:

- You must have administrator, root, or sudo 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.
- For RAM requirements on the deployment system, follow these instructions:
 1. Visit the [FPGA Software Download Center](#).
 2. Select the desired software product from the collection.
 3. Within the selected software product landing page, see **System Requirements** section. For additional memory recommendations, refer to the latest [Intel Quartus® Prime Pro Edition](#) release notes.

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 run on one of the following supported operating systems:

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

Software Prerequisites

Develop your host application using one of the following Intel FPGA SDK for OpenCL- and Intel Quartus Prime software-compatible C compiler or software development environment:

- For Windows systems, use Microsoft Visual Studio Professional and Microsoft Visual C++ versions 2015 or later.

Note: Intel FPGA SDK for OpenCL plug-ins support Windows* 10 and Windows* Server 2016 systems, with installed Microsoft Visual Studio* 2017 or 2019 version. For correct working of Visual Studio plug-ins, you should have installed extensions for C/C++ Development, for example, Desktop development with C++ and Microsoft .Net Framework 4.5 or higher. For more information about the IDE workflow on Microsoft Visual Studio, refer to [FPGA Workflows in Microsoft Visual Studio](#).
- For Linux systems, use Eclipse 2019-12 or later and GCC 7.2.0. For more information about GCC, refer to [GCC Requirement](#) on page 7.

Note: For correct working of Eclipse extensions, you should have installed C/C++ Development Tools and Java SE 11 or higher. For more information about the IDE workflow on Eclipse, refer to [FPGA Workflows in Eclipse](#).

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

For Intel FPGA SDK for OpenCL packages that include Intel Code Builder, Intel Code Builder requires Java SE version 1.8.71 or later to run.

Related Information

Intel FPGA SDK for OpenCL FPGA Platforms

1.1.1. GCC Requirement

For Linux host systems, the Intel FPGA Emulation Platform for OpenCL software requires that you have at least GCC 7.2.0 on your system. GCC 7.2.0 provides the `libstdc++.so.6` shared library. Newer versions may work as well. Specifically, `libstdc++.so.6.0.24` library or later is required that defines `GLIBCXX_3.4.24` and `CXXABI_1.3.11` symbol versions.

If you do not have a sufficiently new GCC and `libstdc++.so.6` library installed, you may encounter issues when trying to run OpenCL host programs that target the Intel FPGA Emulation Platform for OpenCL software. If the correct version of `libstdc++.so` library is not found at run time, the call to `clGetPlatformIDs` function fails to load the emulation platform and returns `CL_PLATFORM_NOT_FOUND_KHR` (error code -1001). Depending on the version of `libstdc++.so` library found, the call to `clGetPlatformIDs` function may succeed, but a later call to the `clCreateContext` function might fail with `CL_DEVICE_NOT_AVAILABLE` (error code -2).

To verify the GCC version you are using, run the following command:

```
$ gcc -dumpversion
```

Ensure that the command reports version 7.2.0 or greater. If you do not have a sufficiently recent GCC, use the following instructions to install a newer GCC on your system.

Installing GCC 7

The process for installing the required GCC differs depending on the host operating system as described in the following sections.

Ubuntu*

If your Ubuntu version does not include GCC 7.x or later by default, you can obtain GCC 7.x by using the following commands:

```
$ sudo add-apt-repository ppa:ubuntu-toolchain-r/test
$ sudo apt-get update
$ sudo apt-get install gcc-7 g++-7
$ sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-7 60 \
--slave /usr/bin/g++ g++ /usr/bin/g++-7
$ sudo update-alternatives --config gcc
```

Note: Ensure that you select `gcc-7` for the `sudo update-alternatives` command. Use the `update-alternatives` utility to swap between GCC versions as required.

CentOS and Red Hat Enterprise Linux* Version 7

Note: Unfortunately, installing the [Developer Toolset*](#) (for example, `devtoolset-7`), does not provide the desired result because it does not include a newer `libstdc++.so.6` library. The Developer Toolset* provides a custom version of the GCC that uses the older `libstdc++.so.6` library originally installed on the system.

For CentOS and Red Hat Enterprise Linux version 7, Intel recommends that you compile GCC from source and install it using the following instructions:

1. Download the tarball of the GCC version you want to install.

```
wget http://www.netgull.com/gcc/releases/gcc-7.2.0/gcc-7.2.0.tar.gz
```

2. Unpack the tar archive and change the current working directory.

```
tar xzf gcc-7.2.0.tar.gz
cd gcc-7.2.0
```

3. Install the bzip2 package.

```
yum -y install bzip2
```

4. Run the download_prerequisites script to download some prerequisites required by the GCC.

Note: You must run the script from the top-level of the GCC source tree.

```
./contrib/download_prerequisites
```

5. Once the prerequisites are downloaded, execute the following command to start configuring the GCC build environment:

```
./configure --disable-multilib --enable-languages=c,c++
```

6. Run the following command to compile the source code. Compilation may take a few hours to complete.

```
make -j 4
make install
```

7. Refer to [Verifying Your GCC Installation](#) on page 8 section for instructions to verify if your GCC installation was successful.

SUSE Enterprise Linux* 12

For SUSE, Intel recommends that you compile GCC from source and install it by following the instructions provided for CentOS and Red Hat Enterprise Linux.

Verifying Your GCC Installation

To verify your GCC installation, ensure the following:

- Shell can find the correct GCC binaries.
- The newer GCC is using correct shared libraries (in particular, libstdc++.so.6).

This depends on how you installed the GCC and set the \$PATH and \$LD_LIBRARY_PATH environment variables. You might have installed the GCC binaries and libraries to the default locations for your operating system.

Verifying the Correct GCC Binaries are Used

Perform these steps:

1. Verify if your system is using the newly-installed GCC binaries.

```
$ which gcc
```

2. Ensure the output points to the correct GCC.

```
$ gcc -dumpversion
```

3. Ensure GCC reports a version of 7.2.0 or higher.

If the above commands do not report the correct GCC version, modify your `$PATH` environment variable to point to the location of the newly installed GCC binaries.

For Ubuntu, the default install location for GCC binaries is `/usr/bin`; and libraries is `/usr/lib64`. However, binaries are versioned (for example, `gcc-7`). Use the `update-alternatives` utility to ensure `gcc` is a symbolic link to the desired GCC version. On Ubuntu systems, you need not update `$PATH` and `$LD_LIBRARY_PATH` environment variables as the shell and dynamic linker should automatically search in the correct places for the binaries and libraries, respectively.

Tip:

You can place the following command in your shell startup script or an environment setup script so that the development environment is automatically set up when you enter it.

```
$ export PATH=</path/to/new/gcc>/bin:$PATH
```

To verify that your `$PATH` environment variable has been modified appropriately, use the following command:

```
$ echo $PATH
```

Check the output to ensure that the path to your newly installed GCC is part of your `$PATH`. Repeat these steps to ensure your system is using the correct GCC binaries.

Verifying the Use of Correct Shared Libraries

To verify your system is using the new GCC libraries, check that a test compile links against the correct `libstdc++.so.6` library version by using the following steps:

```
$ echo "int main() { return 0; }" > test.cpp
$ g++ -Wl,--no-as-needed test.cpp -o test
$ ldd test
    linux-vdso.so.1 => (0x00007ffff6df35000)
    libstdc++.so.6 => /usr/lib64/libstdc++.so.6 (0x0000003451600000)
    libm.so.6 => /lib64/libm.so.6 (0x0000003446a00000)
    libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x000000344d600000)
    libc.so.6 => /lib64/libc.so.6 (0x0000003446600000)
    /lib64/ld-linux-x86-64.so.2 (0x000055ff56398000)
$ strings /usr/lib64/libstdc++.so.6 | grep GLIBCXX_3.4.24
GLIBCXX_3.4.24
```

Use the output from the `ldd` command to determine the path to the `libstdc++` `.so.6` library that you need to use with the `strings` command (see highlighted strings; your output may differ). Ensure that the `strings` command returns `GLIBCXX_3.4.24`.

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/908051137143006116>