

FPGA Programming Made Easy - insideHPC

MichaelS

Sponsored Post

Field-programmable gate arrays (FPGAs) are becoming an increasingly popular tool for applications where high performance, low latency and power efficiency are required. Since an FPGA can be re-configured for a specific application, the opportunities for this type of accelerator are quite varied. However, in the past, it was necessary to understand a complex programming language such as Verilog or VHDL, that was designed for a specific FPGA. An FPGA allows for the use of a custom hardware accelerator that can be used for one application and then reconfigured for a completely separate application. Hardware engineers may have understood these high level definition languages, but other software developers may have found these languages hard to use and quite different from other forms of programming.

OpenCL (Open Computing Language) is an API for the programming of a diverse set of processors. OpenCL is a royalty free and open standard that is used for a wide range of accelerators, including FPGAs. Using a familiar language such as OpenCL, developers can become more productive, sooner when deciding to use an FPGA for a specific purpose. OpenCL is portable and is designed to be used with almost any type of accelerator.

As with any devices, there are always optimizations that can be made in software to get better performance. As with most accelerators, one of the most effective methods is to express pipelining, which can be done with OpenCL. Similar to the methods that developers are familiar with, the use of libraries that are optimized for FPGAs task should be used where these libraries are available.

When developing an application for an FPGA that uses OpenCL, there are a few different ways to create a high performing application. These

would be:

- Emulation – develop the application and then run on a Windows or Linux system. The performance will not be high, but will allow for learning about and creating an application that behaves as required.
- Offline compile – create the application using OpenCL and generate an FPGA programming file. Then move the file to an actual FPGA for further testing.
- Cloud – find a cloud provider that has FPGAs as part of their offering. By using the supplied Intel tools in the cloud, further development can be done, with out having to invest in new hardware.

[Simplify your FPGA development using OpenCL](#)[Click To Tweet](#)OpenCL also includes an application program interface (API) for the host (CPU) to communicate with the device (an FPGA in our case), traditionally over PCIe, or for one kernel to communicate with another without host interaction. In addition to this, Intel provides an I/O Channel API to stream data into a kernel directly from a streaming I/O interface such as 10Gb Ethernet.

For most programmers, there is a need to abstract the hardware details and flow away from the development process. With an SDK, reports as to the performance of the application can be generated in seconds, as well as the pipeline dependencies. This allows for faster development and debugging of a new application. By giving software programmers an easy to use and understand interface to the complexities of an FPGA, new and optimized applications can be developed faster.

[*Download Intel® SDK for OpenCL™ Applications for free.*](#)

Have you tried the Intel Intel Media SDK?

Yes

No