

## Application acceleration with OpenCL

OpenCL is an open standard which supports the acceleration of applications using non-traditional computing devices such as graphics cards, and more recently programmable logic. Application developers can split the compute heavy parts of their application, and arrange for the computation to be done on highly parallel resources without tying themselves down to a specific vendor. For example, an application can be built which can run on CPUs using threads, on GPUs using 100's or 1000's of shaders, or FPGAs using parallel custom logic.

In practice, functionality is platform independent, but in order to tune for performance, developers must tune their application description in order to take advantage of the specifics of the platform. This offers distinct challenges to OpenCL vendors - being able to provide enough information to allow this tuning to take place is actually quite difficult, and FPGAs are doubly awkward, since the underlying processing architecture is variable (unlike a GPU), though this flexibility is where FPGAs have a latent advantage, if it can be harnessed.

We would like to understand better the tradeoffs of application development using CPUs, GPUs and FPGAs by performing a mapping of an application for each of these targets. The field session team will be working closely with the developers of the Xilinx OpenCL compiler, especially for the FPGA target. It is however important that the application is mapped and tuned with equal effort for CPUs and GPUs. We have hardware available on-site for both GPU and FPGA acceleration.

A successful project will have the following attributes

- An application optimized for CPU, GPU and FPGAs
- The relative performance gains can clearly be seen visually (for example, increased frame rate or size in a video application)
- Guidelines for how the application is tuned for FPGAs
- A good description of WHY acceleration is achieved in each case

Students will work on-site alongside our engineers, using the same tools and methodologies. In addition, it is expected that limitations are found with the current tooling, and students are expected to be able to file bug reports and track their resolution. Finally, though the tool hides a lot of the detail of hardware design from the end user, it again is somewhat inevitable that some hardware debug will be required - therefore some experience with FPGA simulation and debug would be very useful (though there is a lot of expertise here). OpenCL is very easy to pick up, with good documentation and design examples freely available.

Finally, this is not an artificial project. These sorts of demonstrators and the insights gained into performance tuning for FPGAs are crucial for the success of Xilinx's OpenCL solution.