UMass Lowell EECE.6540 Heterogeneous Computing

Lab 5: Get Started with oneAPI/DPC++

Due Date: See the Blackboard announcement.

rev:04/14/2020

Objectives

- Learn the basics of oneAPI/DPC++ programming framework
- Learn the basics of compiling and executing a oneAPI sample program on a FPGA board.

Description

In this lab, you will follow the getting-started guide on Intel's website to learn how to compile and run a oneAPI/DPC++ sample project. You are expected to learn from the sample how a oneAPI/DPC++ program is implemented. You can then compare it with the OpenCL programs that you have worked with.

You need to do the following in this lab:

(1) Login to Intel DevCloud.
Run "devcloud_login" to choose "Arria 10 oneAPI" node. You will then be logged into a node that is equipped with Arria 10 and supports oneAPI development

(2) Setup the oneAPI development environment.
Run "tools_setup" to choose "Arria 10 oneAPI" to set up the proper environment variables in your shell.

(3) Follow the instructions at https://software.intel.com/en-us/get-started-with-intel-oneapi-base-linux-run-a-sample-project-using-the-command-line
to create a cpp project. You can then select "Intel oneAPI Base Tookit"-> "oneAPI DPC++ Compiler" -> "CPU, GPU and FPGA" -> "Vector Add" project supported by all types of processors. Save the project to a directory that you specify.

(4) After creating the project folder, you can then compile the source code to emulation option using the following commands

```
make fpga_emu -f Makefile.fpga
```

you can compile the report using the command below, and You can view the report at `vector-add_report.prj/reports/report.html.`

```
make report -f Makefile.fpga
```

To run the emulation code, you can do
`make run_emu -f Makefile.fpga`

(5) To compile for execution on FPGA hardware, do the following
`make hw -f Makefile.fpga`
This will take a couple of hours. After compilation completes, we can run the design on FPGA card
`make run_hw -f Makefile.fpga`

(6) Read the source code of the design. List the major differences between this DPC++ code and its OpenCL version.
(7) Read the reports generated during the compilation phase to understand the analytics data about the function realized on FPGA, such as area and memory usages, loop structure and other information.
(8) Optionally, try to select and create other DPC++ projects and repeat the process.

In this lab, you will practice the commands, and perform the compilation and execution steps in a Linux environment. **You must execute the binary on the FPGA board**, instead of running the program in emulation mode. Please plan your work early so you can have adequate time for FPGA development on Intel DevCloud.

Helpful Notes

Start the lab early.

Due date

See the due date posted on Blackboard.

Deliverables

A Lab report that contains the following sections:

1. Description of the lab in your own words
2. Summary of the outcome (final results, working, partial working, etc.). In this lab you will need to capture the outputs from the execution. In addition, please describe how the a DPC++ program differs from an OpenCL program.
3. Main hurdles and difficulties (expected to include some specifics)
4. Things learned from this lab (valuable takeaways)
5. Suggestions (Optional)

Reference

[1] Lab Assignment materials posted on git repository :
https://github.com/ACANETS/eece-6540-labs