UMass Lowell EECE.6540 Heterogeneous Computing

Lab 1: OpenCL Programming

Due Date: See the due date posted on Blackboard page.

rev:05/09/20

Objectives

- Learn the basics of OpenCL programming environment and tools
- Learn the basics of OpenCL kernel design
- Learn the design flow of OpenCL on a FPGA platform

Description

In this lab, you will design an OpenCL program that computes the following matrix operations on input matrices A, B and C and store the result in output matrix D:

D = **A** x **B** + **C**

A is a 200 by 400 matrix (200 rows and 400 columns), B is a 400 by 600 matrix, and both C and D are 200 by 600 matrices. The elements of A, B, C and D are all floating-point numbers. Every element in A has value of 1.0, every element in B has value of 2.0, and every element in C has value of 3.0.

You need to design your OpenCL program so that:

(1) It declares the input matrices **A**, **B** and **C** as arrays, compute **D** on the OpenCL device, and prints out the elements of resulting matrix **D**.
(2) It calculates the results using only **one** OpenCL kernel. That is, you may not use two separate kernels (e.g., one for "multiply" and the other one for "add"), even though it is possible to use two;
(3) It releases host side and device side resources before completion;
(4) It has error checking at the important steps.
(5) Optional step: make the size of matrices as command line arguments m, n, p: A as m x n matrix, B as n x p, C and D as m x p matrices.

In addition, this lab is to familiarize you with the OpenCL development environment, tools and particularly the design flow on an FPGA based platform. You will practice the commands and perform the compilation and execution steps in a Linux environment. While the lab can be completed on a MacOS platform where OpenCL is natively supported, we expect you to eventually use the Linux server on Intel DevCloud (follow instructions in Blackboard Labs section and the lecture in Week 4).

Helpful Notes

You may already notice the similarity of this lab to the Matrix Multiplication example. You can build your design on top of the example we discussed in the lecture. You are encouraged to think from the data parallelism perspective how the Multiple-Add in this lab differs from the Matrix-Multiplication example.

Suggested steps:

1. Sketch on paper how the multiply-add operations are performed on matrices A, B and C.
2. Figure out the essential arithmetic operations that lead to the result of **one** element in D.
3. Make a copy of the MatrixMulti to see how you can reuse some of the code.
4. Design the new OpenCL kernel function.
5. Compile, execute and debug your code.
   You can use the emulation mode on the Intel FPGA SDK to void the long FPGA compilation time during the debug stage. Type “make emu” will generate binary for emulation. For details on using emulation, please refer to Section 1.9 in the Intel FPGA SDK for OpenCL Programming Guide.
6. Repeat 4 and 5, or 2 if necessary, until the program outputs correct results.
7. Hint: You can use a smaller matrix such as 2x4, 4x6 to test your design, then scale it up.

Due Date

See 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.)
3. Main hurdles and difficulties (expected to include some specifics)
4. Things learned from this lab (valuable takeaways)
5. Suggestions (Optional)
6. Link to your final source code on github

Reference

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