Course Syllabus

GPU Programming and Architecture
Spring 2014

Instructor: Dr. Rafael Ubal (ubal@ece.neu.edu)
Teaching Assistant: Fanny Nina Paravecino (ninaparavecino.f@husky.neu.edu)
Schedule: Wednesday 6pm-7.30pm
Office hours: By appointment.
Office location: 140 The Fenway (TF), see directions below
Phone number: 617-373-3895

Overview

Graphics processing units (GPUs) with support for general-purpose computing, as well as integrated CPU-GPU heterogeneous devices, are becoming mainstream designs in current commercial processors. This course covers the architectural and programming framework of a GPU, using the AMD Southern Islands GPU family as a subject of study. We will start presenting the OpenCL programming model as a high-level language for general-purpose GPU computing. Going down into the hardware, we will present the Southern Islands instruction set architecture (ISA) executed by the machine. Finally, we will analyze the architecture of a GPU device, focusing on the design features that make it different from a CPU. The last part of the course will be devoted to state-of-the-art research topics in computer architecture, with a discussion of possible ways in which students can get involved in the research activities conducted in the NUCAR (Northeastern University Computer Architecture Research) group.

Textbook

Office Location

1) Find the office building at 140 The Fenway (TF), and enter the main door located at the parking lot.

2) Take the main elevator to the 3rd floor.

3) Once on the 3rd floor, call me at 617-373-3895. My office is in a locked research laboratory. I will meet you on the hallway right by the elevator and let you in.
Grading
A non-official grade will be given at the end of the course based on:

- Homework assignments – 50%
- Final project – 50%

Homework assignments.
There will be a total of 7 homework assignments. Each homework assignment will have two sets of questions: mandatory and optional questions.

- Mandatory questions cover material that should be understood and internalized in order to keep up with the following lectures. These questions will give you 100% of the HW credit.
- Optional questions are open-ended problems. Students can decide how much time and effort they want to invest in these questions. Optional questions are intended to be completely flexible and even replaceable. If you come up with a better problem, go ahead: define it and solve it! This set of questions will give you extra credit in the final grade.

You can submit your HW assignment by email as a PDF document to the course teaching assistant (see his contact information above). Submitted homework is only accepted if submitted before class starts on the due date.

Final project.
Students can develop a final project related with the topics presented in class, including, but not limited to, OpenCL programming, GPU hardware design, or simulation. A possible inspiration for the final project can be the open-ended optional questions of the homework assignments. Also, we will discuss possible projects in class while we cover a given topic. We will also have some invited speakers from our GPU research group who will suggest open problems from their own research.

Working in groups.
All work assigned in class, including the final project, is voluntary, and can be done individually or in groups of two. You can decide how deeply you want to get involved with the course. To keep your interest and understanding on the new material, my suggestion is that you complete at least the mandatory problems in the homework assignments.

The best grades.
Those students having the best grades in the course will be offered to participate in a directed study either in the Fall 2013 or the Sprint 2014 semesters. A directed study is a 4-credit course consisting in conducting or collaborating with an active project in our research group at Northeastern. The number of students selected for a directed study will depend on my availability during the corresponding semester.

The very best grade.
The student (or group of two students) with the best grade, or with an outstanding final project, will receive a surprise. An awesome one.
Topics

The following list is a preliminary schedule of the topics to be covered in the course. The topics and their order are subject to change.

Unit 1 – Introduction to OpenCL
- Why GPU computing?
- The OpenCL programming model.
- Host program and device kernel.
- OpenCL objects.
- Basic program: vector addition.

Unit 2 – Algorithms in OpenCL
- Square matrix transpose.
- Square matrix multiplication.
- Work-groups.
- OpenCL synchronization model.
- OpenCL memory model.
- Matrix multiplication with local memory.
- Parallel reduction algorithms.
- Sorting algorithms.

Unit 3 – Architecture of a GPU
- The AMD Southern Islands instruction set architecture.
- SIMD (Single-Instruction Multiple-Data) execution model.
- Scalar and vector instructions.
- Thread divergence.
- Nested control flow.
- The Multi2Sim simulation framework.
- Disassembler, emulator, timing simulator, and pipeline visualization.

Unit 4 – Miscellaneous Research Topics and Opportunities
- Memory hierarchies and coherence protocols on APUs.
- Interconnection networks on GPUs.
- Rendering graphics using OpenGL.
- The GPU graphics pipelines.
- Simulation of new GPU architectures.
- OpenCL/CUDA to LLVM compiler front-ends.
- LLVM to NVIDIA/Intel/AMD compiler back-ends and optimizers.
- Discussion of course projects and future collaborations.
Lectures

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Date</th>
<th>HW due</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Wed. 1/8</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Wed. 1/15</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>Wed. 1/22</td>
<td>#1</td>
</tr>
<tr>
<td>4</td>
<td>Wed. 1/29</td>
<td>#2</td>
</tr>
<tr>
<td>5</td>
<td>Wed. 2/5</td>
<td>#3</td>
</tr>
<tr>
<td>6</td>
<td>Wed. 2/12</td>
<td>#4</td>
</tr>
<tr>
<td>7</td>
<td>Wed. 2/19</td>
<td>#5</td>
</tr>
<tr>
<td>8</td>
<td>Wed. 2/26</td>
<td>#6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>HW due</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>Wed. 3/5</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>No class</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Wed. 3/12</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>Wed. 3/19</td>
<td>#7</td>
</tr>
<tr>
<td>11</td>
<td>Wed. 3/26</td>
<td>#8</td>
</tr>
<tr>
<td>12</td>
<td>Wed. 4/2</td>
<td>#9</td>
</tr>
<tr>
<td>13</td>
<td>Wed. 4/9</td>
<td>#10</td>
</tr>
<tr>
<td>14</td>
<td>Wed. 4/16</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Presentations</td>
<td></td>
</tr>
</tbody>
</table>

User Accounts

Each student has an account on a Linux machine with an AMD Fusion processor and the AMD OpenCL framework installed on it. The machine name is fusion1.ece.neu.edu.

Account information.
The user name is the same as your Husky account, but replacing dots “.” with dashes “-.” For example, user smith.j@husky.neu.edu can log in as smith-j. The password is opencl2014, but should be changed on the first login using command passwd. Accounts keeping the original password will be erased automatically after the second week of class.

Remote connection.
You can access the fusion1 machine through an secure SSH connection. If you use Linux or MacOS, you can log in by opening a terminal and running command ssh smith-j@fusion1.ece.neu.edu. If you use Windows, you can download an SSH client, such as PuTTY (www.putty.org).

Compiling OpenCL source code.
When using the GNU C compiler gcc to build an OpenCL program, some additional flags need to be added to the gcc command. Flag -I specifies a directory where the compiler should look for non-standard header files; flag -L specifies a directory to look for non-standard libraries; and flag -l links the final program with a particular additional library. When compiling an OpenCL program stored in demo.c, you can use command line gcc demo.c -o demo -IOpenCL -I/opt/AMDAPP/include -L/opt/AMDAPP/lib/x86. Notice that there should be no space between the flags and the paths that follow.
Final Project

Each student should choose the topic for the final project individually or in groups of two. Possible topics are suggested in the optional, open-ended questions of each homework assignment, but students are encouraged to propose any interesting alternative topic. The project will be graded based on:

- A short PDF report, describing the project motivation, goal, tools used, implementation, and conclusions. The report should not exceed 2 pages, using an 11pt Times font. If the project includes an implementation, the software should be attached as a ZIP (or tar) file. The PDF and ZIP files should be sent to me by email by the due date specified below.

- All projects will be presented during an additional lecture. Each presentation should be around 5 minutes long (it should not exceed 10 minutes in any case). This means the right amount of slides you need to prepare is probably 5. Please rehearse your presentations to make sure you keep it within the time restrictions. At the end of the presentation lecture, the student with the best performance in the course will be awarded with the promised surprise!

Everyone is encouraged to attend the project presentation lecture (not only people presenting a project). This is a promising event where you can learn what other motivated students decided to work on.