Overview

Graphics processing units (GPUs) with support for general-purpose computing are becoming mainstream designs in current commercial processors, often integrated in a single die forming heterogeneous CPU-GPU devices. 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. We will analyze the architecture of a GPU device, comparing it with a traditional CPU architecture, and focusing on the architectural features that a programmer should be aware of for performance scalability.

The last part of the course will cover state-of-the-art research topics related with GPUs, including architecture, simulation, and compilation. We will hold project presentations and discussions, and we will discuss possible ways to engage research and development activities 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
This course is not an official course at Northeastern, and thus the grades will not compute in the students' official GPA. However, each student will receive a final grade based on:

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

Homework assignments
There will be weekly homework assignments. The assignments should be submitted on Blackboard before class starts on the specified due dates (see course calendar below). Each assignment will be formed of a set of questions to practice the material presented in class, and one or more open-ended questions.

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, simulation, or compilation. We will discuss possible projects in class as we cover each topic. We will probably also have invited speakers from our GPU research group who will suggest open problems related with their own research.

Working in groups
All work assigned in this course 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 a good understanding of the material and the motivation to move through the units, my suggestion is that you complete at least the first problems (not open-ended problems) in the homework assignments.

The best grades
For those students with the top 3 to 5 grades in the course, I will offer to set up weekly meetings to continue with a guided research activity, either extending the course project or starting with a new topic. In the case of undergraduate students, I will offer 4-credit directed studies during the Fall 2014 semester.

The very best grade
The student with the best final grade (or with an outstanding final project) will obtain an awesome final course prize. Surprise!
**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>Thu 5/8</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Mon 5/12</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>Thu 5/15</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>Mon 5/19</td>
<td>#1</td>
</tr>
<tr>
<td>5</td>
<td>Thu 5/22</td>
<td>-</td>
</tr>
<tr>
<td>No class</td>
<td>Mon 5/26</td>
<td>#2</td>
</tr>
<tr>
<td>6</td>
<td>Thu 5/29</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>Mon 6/2</td>
<td>#3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Date</th>
<th>HW due</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Thu 6/5</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>Mon 6/9</td>
<td>#4</td>
</tr>
<tr>
<td>10</td>
<td>Thu 6/12</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>Mon 6/16</td>
<td>#5</td>
</tr>
<tr>
<td>12</td>
<td>Thu 6/19</td>
<td>-</td>
</tr>
<tr>
<td>13</td>
<td>Mon 6/23</td>
<td>#6</td>
</tr>
<tr>
<td>14</td>
<td>Thu 6/26</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, 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 -lOpenCL -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.

- All projects will be presented during the last day of class. Each presentation will be between 10 and 20 minutes, depending on the number of projects. This means the right amount of slides you need to prepare is probably between 5 and 15. Please rehearse your presentations to make sure you stay within the time restrictions. At the end of the project presentations session, we will hold an award ceremony, where the student with the best performance in the course will receive 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.