ECE 8823: GPU Architectures

Sudhakar Yalamanchili
School of Electrical and Computer Engineering
Georgia Institute of Technology

Course Objectives

• Basic CUDA Programming
  ❖ Programming Model: Basic structures and idioms
  ❖ Simple microarchitecture-targeted optimizations

• Core concepts of general purpose graphics processing unit (GPGPU) architecture
  ❖ Execution Model: How massively parallel program execution is orchestrated using the Single Instruction Multiple Thread (SIMT) model
  ❖ Basic microarchitecture: Organization principles

• Advanced microarchitecture concepts
  o Optimization of control flow, memory access behavior, and thread scheduling
  o Datapath organization, power/energy optimizations
What You Will Do

- Understand how GPU architectures work
  - Fundamental architectural concepts and challenges

- Acquire knowledge to optimize systems
  - As a designer
  - As a user
  - As a researcher

- Assignments (tentative)
  - CUDA Programming (2-3)
  - Build an instruction set emulator for a GPU processor
  - Implement solutions for control and memory divergence
  - Project: cycle-level microarchitecture simulator or equivalent
    - Multiple components

What You Will Learn

- How the bulk synchronous parallel (BSP) model is represented in CUDA
  - Basic organizational principles of a CUDA program
  - To write some simple CUDA programs

- The Single Instruction Multiple Thread (SIMT) execution model
  - And how the hardware executes them

- Determinants of SIMT program performance
  - And how it can be improved
Course Information

- Web page:
  http://users.ece.gatech.edu/~sudha/academic/class/ece8823/Fall2015
  - Will be constantly updated, so check it regularly
- Prerequisite: ECE 4100/6100 or CS 4290/6290
- Course Materials:
  - Journal & Conference papers, patents, class notes
- Online content for the text
  http://booksite.elsevier.com/9780124159921/

Grading Policy

- Homework Assignments: 40%
  - Individual work, no collaboration other than specified!
  - No late assignments will be accepted
  - You are expected to make a passing grade on the assignments to pass the course!
- Exams
  - Midterm: 15% (October 7th, 2015)
  - Assignments: 40%
  - Project: 30%
  - Final: 15%: December 9th, 2015, 2:50-5:40
- Teaching Assistant: Si Li (contact info TBD)
Academic Honesty

• You are allowed and encouraged to discuss assignments with other students in the class

• Any copying of code is unacceptable
  ◆ Includes reading someone else’s code and then going off to write your own
  ◆ Includes verbal specific directions on data structures and code

• Giving/receiving help on an exam is unacceptable

Preliminary Course Layout*

• CUDA Programming (CUDA Programming Guide) 3-4 Weeks
  ◆ Introduction to CUDA C (Chapter 3)
  ◆ CUDA Data Parallel Execution Model (Chapter 4)
  ◆ Memory Hierarchy (Chapter 5, Chapter 8.3)
  ◆ Performance Issues (Chapter 6)

• GPU Microarchitecture (papers, patents) 2-3 Weeks
• Control Divergence (papers) 2-3 Weeks
• Memory Divergence (papers) 1-2 Weeks
• Scheduling (papers) 2-3 Weeks
• Power (papers) 1-2 Weeks
• Advanced Topics (CDP, OpenCL, OpenACC) 1 Week

*Note that this is subject to change
Getting A Good Grade

- Attend class and ask questions in and out of class!
- Follow reading assignments
- Start early on assignments – they will take time!
- Listen in class – pay attention to suggestions
- Make use of the TA, Piazza, my office hours, and email
- Do your own work!
Getting a Poor Grade

- Miss class
- Go weeks without reading or following up on lecture material
- Late start on assignments
- Ignore readings
- Rely on others to tutor you
- Poor coding habits
- Ignore gaps in your understanding
- Pay poor attention to assignment contents

Lets Get Started........