ECE 611 ADVANCED MICROPROCESSORS  Spring 2014

Class info: Wednesdays, 7:20pm-10 pm, Robinson Hall B A243
Instructor: Prof. Houman Homayoun
Web: http://cseweb.ucsd.edu/~hhomayoun/
Course website: Materials will post on Blackboard
Email: hhomayou@gmu.edu
Please use “AM: Your header here” for all your e-mails
Office hours: TBA
Prerequisite: ECE 511 or equivalent.

Course Description
The microprocessor industry is undergoing a significant change with the introduction of multicore and many-core processors. As semiconductor technology changes, the tradeoffs underlying microprocessor design constantly evolve which is leading to dramatic changes in the design at the architecture and underlying microarchitecture level. This course qualitatively and quantitatively examines computer design trade-offs. In this course we study the major developments in microprocessor design over the past decade and then explore future directions for computer architecture in light of current process technology trends. The first part of the course examines microarchitecture techniques employed in current superscalar processors. Then we explore the challenges to achieving the full benefits of future process technology scaling, the architecture and microarchitecture solutions currently being adopted, as well as potential solutions that may be adopted in the future. The course is structured around the three primary building blocks of general-purpose computing systems: processors, memories, and networks. The first half of the course focuses on the fundamentals of each building block and will enable students to understand how these three building blocks can be integrated to build a simple multicore system. Topics include processor pipelining; single-core and multi-core cache microarchitecture; and network topology, routing, and flow control. The second part of the course delves into more advanced topics related to processors and memories and will enable students to understand sophisticated, modern multicore systems.

Who should be interested in this course?
The course should be of interest to most ECE and CS graduate students, including hardware oriented students wishing to understand the impact of low level optimizations on system performance/cost; software oriented students interested in making the most effective use of future hardware systems; and communications or systems oriented students wishing to study examples of highly complex systems.

About Computer Architecture
Computer architecture is the science of selecting and interconnecting hardware components to create a computer that meets functional, performance and power goals. In other words, computer architecture is about the numerous ways chip architects translate an ever growing supply of transistors into exciting products that take advantage of process technology improvements. Two major challenges facing computer architects today are dealing with tight power budgets and achieving high performance as off-chip bandwidth diminishes in comparison with available on-
chip compute resources. This course provides a strong foundation for understanding modern computer system architecture and for designing future systems to address these challenges.

**Goals in This Class**

- Understand Advanced techniques deployed in state-of-the-art microprocessors from Intel/AMD/IBM and other major industries.
- Learn how to use tools, techniques, and models to evaluate microprocessor design.

**Course Content (This is tentative and will change as we progress in the class)**

- Introduction; Pipelining, Concept of Superscalar (Week 1)
- Quantitative Analysis of Superscalar Processors (Week 2)
- Quantitative Analysis of Multicores Processors (Week 3)
- Advanced Superscalar Techniques: Advanced Branch Predictors (Week 4)
- Advanced Superscalar Techniques: Advanced Cache Management (Week 5)
- Advanced Superscalar Techniques: Front-end Design (Week 6)
- Advanced Superscalar Techniques: Back-end Design (Week 7)
- Student Presentation (Week 8-10)
- Paper Reading, Discussions, and Student Seminars (Student will give lectures and we all discuss) (Week 6-12)
  - You select among a set of research papers (from the reading list)
- Intel P4 Architecture, Niagara Architecture (Week 11)
- GPU Architecture, AMD and Nvidia GPU comparison (Week 12)
- AMD Fusion (Week 13)

**Literature**

- NO TEXTBOOK REQUIRED
- Reading material will be selected from leading conferences, journals, and magazines including ISCA, MICRO, HPCA, DAC, DATE conferences as well as active research projects. All required material will be made available on the course web page.

**Grading Policy**

- **Assignments: (30%)**
  - Assignment#1 (Week 3) 6%
  - Assignment #2 (Week 4) 6%
  - Assignment #3 (Week 5) 6%
  - Assignment #4 (Week 6) 6%
  - Assignment #5 (Week 7) 6%
- **Project: (20%)**
  - Project proposal and 1-page initial progress report (Week 6): 5%
  - Final project report and results: 20%
- **Presentation (10%)**
Student presentation; to be scheduled for each student individually (10%)
  - Paper Summary (5%) (Week 6-12)
  - Midterm (10%) (Week 6)
  - Final (25%)

**Project**
To be scheduled. Students need to form their own group (up to 3 students per group). The project involve learning and using computer architecture simulation tools for measuring performance, estimating power and temperature, and evaluating reliability of the state of the art multi-core processors such as Intel Nehalem or AMD Bulldozer.

**Reading List**
Depending on the number of students in class, each student needs to present 1~2 papers from the reading list. (the reading list includes key papers in computer architecture, so you will most likely find slides for these papers online. If not, you can contact the authors, and they will forward you the slides, if they have).

**Assignments**
There will be 3 programming assignments that are designed to help you learn the simulation infrastructure that is commonly used in computer architecture community: [www.simplescalar.com](http://www.simplescalar.com) and [http://cseweb.ucsd.edu/~tullsen/smtsim.html](http://cseweb.ucsd.edu/~tullsen/smtsim.html)
Assignments require programming skills in C or C++. Also require that you are familiar with UNIX systems. It is also recommended you know one of scripting language such as python (recommended) or perl.

**Project**
This is probably the most important part of the course. You will be required to conduct “research” in computer architecture
  - Several possible ideas will be given
  - Also you may come up with your own

Since this is a class project negative results are OK. i.e. you do not need to prove something is working! If you proof some idea is not working it could also be good! Groups of 2 or 3 if necessary (depends on class size too). More details coming “soon”

**Tools:**
In this class you will learn how to use these architecture simulator tools:
  - Simplescalar
  - SMTSIM

**Link**
Architecture tools that will be used in this class:
Acknowledgment:
Some of the notes used in this course have incorporated materials/adapted from Computer Architecture courses by Mattan Erez, Andreas Moshovos and David Brooks.