CSE 548: Computer Architecture
Winter 2006
Lectures: MW 12:00-1:20 CSE 403

Instructor

Susan Eggers, CSE 586, 543-2118, eggers@cs.washington.edu
Office Hours: Mondays, 1:30-2:20 or by appointment

TA

Andrew Putnam, CSE 378, 685-4439, aputnam@cs.washington.edu
Office Hours: Wednesdays, 3:30-4:30 or by appointment

Course Material

The purpose of this course is to give you a broad understanding of the concepts behind several advanced microarchitectural features in today's microprocessors and to illustrate those concepts with appropriate (usually modern) machine examples. We will cover the rationale for and the designs of strategies for instruction sets, dynamic branch prediction, multiple-instruction issue, dynamic (out-of-order) instruction scheduling, multithreaded processors, shared memory multiprocessors, and dataflow machines. Some of these topics require some understanding from what is normally thought of as undergraduate material; for these, we'll briefly review that material, and then go on from there.

You will augment your knowledge of the architectural concepts and schemes by doing a quarter-long experimental study that examines and evaluates the performance of a particular architectural design. What you do is up to you, but we'll discuss some options the first day of class. The particular topic you pick is not as important as that it be a vehicle for teaching you (1) how to design architectural experiments, (2) how to choose metrics that best illustrate a design's performance, (3) how to analyze performance data and (4) how to write up your experiment and results – all skills computer architects, and, actually, researchers and developers in any applied subfield of computer science, use on a regular basis.

Lectures will be posted in our web area before class. You would benefit from printing them out *in color* and bringing them to class.

Reading

Most reading assignments will be taken from *Computer Architecture: A Quantitative Approach* by John L. Hennessy & David A. Patterson, Morgan Kaufmann, 2003. To get the most out of the lectures, read the material *before* topics are discussed in class. My lectures won't necessarily follow the same order of subtopics as the text and might take a different slant; I think you'll find that reading the nuts and bolts approach of the authors before class to be helpful.

There will also be some supplementary reading that you will be able to access from the course web pages.

Schedule

There is a weekly schedule in the course web area. The schedule will tell you what topics we will cover and when, what reading should be done before you come to a particular lecture, and when projects are due and exams will be held. I'll be updating this schedule continuously, as I plan each lecture. So you should check it frequently, so that you can anticipate what material we will be covering.

Class Discussion

As in any graduate class, we will all live or die because of the quality of our discussions of the material. So, before each class, think about what you've read for the upcoming lecture and about the material in the previous lecture and come prepared to pose and answer questions, present your views of the architectural schemes we discuss and offer alternatives. And since we meet squarely in the lunch period, feel free to bring your lunch. I certainly will.

Exams

There will be two midterm-sized tests, one in the middle and one at the end.

Projects

As discussed above, the projects will be experimental studies that will give you experience in evaluating architecture features and hone your intuitions about the performance ramifications of changing certain aspects of their implementation. To share the overhead of experimentation and have a built-in buddy with whom to discuss the issues, you should work in teams of two students.

Grading

Grades will be computed using the following **approximate** weighting: midterms = 20% each and the project = 60%. This may change, depending on the size of the projects.

Collaboration

Discussing the course content with fellow students is an effective way to learn the material, and is encouraged. However, the exam must represent your own mastery of the material, and projects must represent the contribution of your team.

Communicating

We will communicate a lot through e-mail. You can use e-mail for asking and answering each others' questions. (But if you have questions that need a detailed or long explanation, it would be much easier to come to our office hours.) Therefore you should register on the class mailing list **immediately**. To add yourself to the class email list, you can visit https://mailman.cs.washington.edu/mailman/listinfo/cse548.