CSE P548: Computer Architecture
Spring 2005
Lectures: W 6:30-9:20 AC 305

Instructor

Susan Eggers, AC 586, 543-2118, eggers@cs.washington.edu
Office Hours: by phone, Tuesdays through Thursdays

TA

Douglas Low, AC 610, 616-8124, douglas@cs.washington.edu
Office Hours: email to set up an 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, if there is time, 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 schemes by doing experimental studies that examine and compare the performance of several alternative implementations for a particular feature. Here you will learn how to design architectural experiments, how to choose metrics that best illustrate a feature's performance, how to analyze performance data and 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 by 3pm the day of 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

Since each class is a whopping three hours long, they will all live or die because of the quality of our discussions. So think about what you've read for the upcoming lecture and about the material in the previous lecture before each class and come prepared to ask and answer questions, present your opinions of the architecture schemes we discuss and offer alternatives.

Exams

There will be a final.

Projects

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. Experiments will usually be done using the SimpleScalar simulator. Douglas will explain how to use the simulator.

You can work in teams of two students for each project. You should be with a different partner for each assignment.

All homework will be assigned early enough in the week that you will have time to read it over and clarify any issues before the weekend (which is when I assume most of you will be doing the homework).

All project reports are due at the beginning of class; no late assignments will be accepted.

Machines

We'll be using attu, a cluster of four Dual Pentium IV 2.8GHz Xeon servers that are comprised of 4GB memory and run Linux 2.4.26. You may use the workstations in AC 002, AC 006 or AC 022 to log onto attu.cs.washington.edu using the installed secure shell client. You can also log on remotely from your own computer. The secure shell client is part of the UWICK package which you can download from:
http://www.washington.edu/computing/software/uwick/starter.

Grading

Grades will be computed using the following **approximate** weighting: final = 30% and projects = 70%. 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. Douglas and I will be mailing out assignments and clarifications of the assignments, if needed. And you should 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 called during our office hours.) Therefore you should register on the class mailing list **immediately**. To add yourself to the class email list, you can visit http://mailman.cs.washington.edu/mailman/listinfo/csep548. Alternatively you can email csep548-request@cs.washington.edu with the word "help" in the subject to return a message

listing all of the email command options. The list archives can be accessed by clicking on the very first URL on the list "home page":
http://mailman.cs.washington.edu/mailman/listinfo/csep548.
The direct link is http://mailman.cs.washington.edu/mailman/private/csep548.