CSE 471: Computer Design and Organization
Spring 2007
Lectures: TTh 10:30-11:50 CSE 503
Section: Th 2:30-3:30 EEB 042

Instructors

Jean-Loup Baer AC 474, 685-1376, baer@cs.washington.edu
Office Hours: Tuesdays, 1:30 - 2:20 and Wednesdays, 10:30 - 11:20 and by appointment

Susan Eggers, AC 586, 543-2118, eggers@cs.washington.edu
Office Hours: TBA

TA

Jacob Nelson, AC 220, nelson@cs.washington.edu
Office hours: Mondays, 11:00 – 11:50 and Tuesdays, 2:30 - 3:20

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 dynamic branch prediction, multiple-instruction issue, dynamic (out-of-order) instruction scheduling, advanced caching techniques, (chip) multiprocessors, multithreading, and dataflow machines. We will start by briefly reviewing topics you covered to some extent in 378: performance metrics, pipelining and caching.

You'll augment your knowledge of the architectural concepts and designs by doing experimental studies that examine and compare the performance of several alternative implementations for a particular feature. Here you will learn (1) how to design architectural experiments, (2) how to choose metrics that best illustrate a feature's performance, (3) how to analyze performance data and (4) how to write up your experiment and results – all skills you will need if you plan to do computer evaluation either in development or research, and in any applied subfield of computer science, not just computer architecture.

You must have already taken 370 and 378 in order to take this class.

Reading

Most reading assignments will be taken from *Computer Architecture: A Quantitative Approach* by John L. Hennessy & David A. Patterson, Morgan Kaufmann, (either the 3rd Edition 2003 or the 4th Edition 2007 is fine). To get the most out of the lectures**, read the material *before* topics are discussed in class**. There will also be some supplementary reading that you will be able to access from the course web pages.

Some chapters of the draft of a book in progress by one of the instructors (JLB) will be available on-line. Corrections, criticisms and ideas for exercises are welcomed and should be e-mailed to baer@cs.washington.edu.

Schedule

There is a daily 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. The schedule is *highly tentative* at this point and will be updated continuously. So you should check it frequently, so that you can anticipate what material we will be covering.

Class Discussion

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.

Exams

There will be two midterms, one in the middle and one at the end. The one at the end will cover material from the second part of the course only.

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 be done using the SimpleScalar uniprocessor simulator and (hopefully) a multiprocessor simulator. In the discussion sections Jacob will explain how to use the simulators.

You'll be working in teams of at two students for each project. Try to work with a different partner for each assignment.

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 with 4GB of memory and running Linux 2.6.12. You may use the workstations in AC 002, AC 006 or AC 022 to log into attu. Alternatively you may use other computers to log in.

Grading

Grades will be computed using the following **approximate** weighting: midterm1 = 25%, midterm2 = 25% and projects = 50% for all. 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, exams 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. We 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 come to our office hours.) Therefore you should register on the class mailing list **immediately**. To add yourself to the class email list, follow the instructions in the Web page.