

CSE 471: Computer Design and Organization
Spring 2015
Lectures: TTh 10:30-11:50 CSE 203
Section: Th 1:30-2:20 CSE 203

Instructor

Susan Eggers, CSE 458, 543-2118, eggers@cs.washington.edu
Office Hours: Tuesdays, 1:30 - 2:20, or by appointment

TAs

Edward (Teddy) Brow, CSE 220, tgibrow@cs
Office Hours: M 3:30 – 4:30 pm
Mark Wyse, CSE 220, wysem@cs
Office Hours: F 10:00 – 11:00 am

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 machine examples. Since parallel programming is currently the biggest unsolved issue in computer architecture, we'll focus primarily on parallel designs and issues that relate to them. We'll also cover designs that are based on two other important concepts in computer architecture, locality and speculation. In particular, we'll cover the rationale for and the designs of strategies for dynamic branch prediction (speculation), multiple-instruction issue (instruction-level parallelism), dynamic instruction scheduling (instruction-level parallelism), advanced cache configurations (locality), chip multiprocessors (thread-level parallelism), multithreaded processors (thread-level parallelism), vector and graphics processors (data parallelism), and dataflow machines (all sorts of parallelism). Some of these topics depend on material you covered in CSE 351/352; for these, we'll quickly review that material, and then go on from there.

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'll learn: (1) how to pose an architectural research problem, (2) how to determine the severity of the problem in practice, (3) how to solve the problem, (4) how to design architectural experiments that measure the severity of and solution to the problem, (5) how to choose metrics that best illustrate a design's performance, (6) how to analyze performance data, and (7) how to write up your experiment and results – all skills you'll need if you plan to do computer evaluation either in development or research, or actually in any applied subfield of computer science, not just computer architecture.

We will also have several guest lecturers during the quarter, who will be talking about topics in their areas of expertise. Some guest lecturers will discuss the topics above; others will tackle architectures of the future. Attendance at the guest lectures is mandatory.

You must have already taken 311/351/352 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, 5th edition, 2011. To get the most out of the lectures, **read the material before topics are discussed in class**. I'll be presenting the material, but

often from a different slant than the text. There will also be some supplementary reading that you will be able to access from the course web pages.

Schedule

There is a daily schedule in the course web area. The schedule will tell you what topics we'll cover and when, what reading should be done before a particular lecture, and when homework is due and exams held. This document is a work in progress. I'll be updating it continuously, as I plan each set of lectures. You should check it frequently, so that you can anticipate what material we'll be covering.

Class Discussion

Coming to and participating in class is part of this course. 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. Although I'll be lecturing, this class is part seminar. And the more seminar-like it is in reality, the more fun we'll have!

Exams

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

Homework

There will be at least four homework assignments, each progressively more challenging. The first one is an empirical study that will give you experience in designing architectural experiments and evaluating architecture features, and will hone your intuitions about the performance ramifications of changing certain aspects of a hardware implementation – in this case, branch prediction hardware. You'll also make your first stab at writing up your results in research-paper style. The second through fourth will involve designing and empirically evaluating a multiprocessor cache coherency scheme. In the discussion sections Teddy and Mark will explain more about the different assignments and how to use the software infrastructure that will allow you to implement and evaluate your designs.

You should work in teams of two students for each assignment. Try to work with a different partner for the first assignment and the second through fourth assignments.

In addition, each of you must write a one-pager critiquing one of the guest lectures, summarizing what was important about the topic, and adding your two cents.

All 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 (E5430) servers with 4GB of memory and running Linux 3.6.13. You may use the workstations in CSE 002, CSE 006 or CSE 022 to log into *attu*. Alternatively, you may use your own computers to remotely log in.

Grading

Grades will be computed using the following **approximate** weighting: the first midterm = 15%, the second midterm = 25%, the team homework = 55% for all and the guest lecture write-up 5%. This may change somewhat, depending on the size and difficulty of the team projects.

Collaboration

Discussing the course content with fellow students is an effective way to learn the material, and is encouraged – this is why you'll be working in teams on the homeworks. However, exams must represent your own mastery of the material.

Communicating

We'll communicate a lot through e-mail. Teddy, Mark and I will be emailing out assignments and clarifications of the assignments, if needed. And you should use e-mail to the whole class for asking and answering each other's questions. But if you have questions that need a detailed or long explanation, it would be much easier to come to our office hours, Teddy and Mark for the assignments, Susan for the lectures. So register on the class email list **immediately**. To add yourself to the class email list, follow the instructions on the 471 web pages.