CSE 586: Computer Architecture
Spring 2003
Tuesdays 6:30-9:20
EE1-003

Instructor

Susan Eggers, 315 Sieg, 543-2118, eggers@cs.washington.edu
Office Hours: by appointment

TAs

Evan Wellbourne, evan@cs.washington.edu
Office Hours: 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 modern machine examples. We will cover the rationale for and the designs of strategies for instruction sets, dynamic branch prediction, exception handling, multiple-instruction issue, dynamic (out-of-order) instruction scheduling, multithreaded processors, and shared memory multiprocessors. The machine examples will be taken from the latest processors from Compaq, IBM, Intel, MIPS or Sun.

Assignments

You will augment your knowledge of the architectural concepts and schemes by doing a variety of homework projects that range from:

- problems from the book that will allow you to work through the nitty gritty of how microarchitecture components work and perform,
- programming assignments whose aim is also to test your understanding of how microarchitecture components work,
- critiques of research papers so that you can come to your own conclusion about whether a particular microarchitecture design is a good idea or not,
- experimental studies that examine and compare the performance of several alternative implementations for a particular feature. For this latter 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 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.
  For some assignments you will work in teams of two.

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). Your part is due at the beginning of class; no late assignments will be accepted.

Reading

Most reading assignments will taken from *Computer Architecture: A Quantitative Approach* by John L. Hennessy & David A. Patterson, Morgan Kaufmann, 2003 (the third edition). 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.

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 might be a midterm and there will be a final. If we have a midterm, the final will only cover material from the second part of the course, and will be more like a long midterm than a final. Stay tuned on this.

Machines

We'll be using the instructional machines ceylon, fiji, sumatra, and tahiti, which run Linux 2.2.14. You can access them from the instructional labs in 232 and 310. The course simulation software (Blis) and C++ are already installed on these machines.

Grading

Grades will be computed using the following approximate weighting: final = 30% and projects = 70% for all. The main message here is that the homework will take much more time than studying for and taking the final, and this should be reflected in your final grade.

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 you or your team.

Communicating

We will communicate a lot through e-mail. Evan and I will be mailing out general information, assignments and clarifications of the assignments, if needed on cseP548@cs. There will also be a special email, cseP548-discuss@cs that you should use for asking and answering each others' questions about the homework. (But if you have questions that need a detailed or long explanation, it would be much easier to ask these in class so we can talk.) Both will be archived. You should register on both class mailing lists **immediately**.