CSE P548: Computer Architecture
Spring 2005

Lectures: W 6:30-9:20 AC 305

Instructor/TA

- Susan Eggers, AC 586, 543-2118, eggers@cs.washington.edu
- Office Hours: by phone, Tuesdays through Thursdays
- Douglas Low, AC 610, 616-8124, douglas@cs.washington.edu
- Office Hours: email to setup 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 sched-uling, 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/Projects

- There will be a final.
- 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.

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.

Schedule

Week	Topic	Reading	Homework Assignments
1 3/30	Architecture overview	**Let your eyes float over chapter 1. We won't cover this in class, but it is good for your general background in computer architecture.**	Take the undergraduate exam; due April 6.
	Instruction set design	**Speedread chapter 2. This is a good summary of background instruction set design material. Gaze at Appendix D. Gaze is a more cursory reading than speedread.**	
2 4/6	Instruction-level parallelism	Read section 3.1.	
	Basics of pipelining	Speedread sections A.1 - A.3. This is a good summary of the basics of pipelining and its implementation, which you will need to know to understand the more advanced material in this course. I'll be covering it, but quickly, at the beginning of lecture.	
	Dynamic branch prediction	Read sections 3.4 and 3.5.	
	Predicated execution	pp. 340-344, 356, 358.	
	Exceptions & pipelining	Read A-38 to A-45 and A-54 to A-56. The beginning of section A.5 is optional.	
3 4/13	Superscalars	Read pp. 215-220.	
	Overview of dynamic scheduling	Read pp. 181-184, 220-224.	
	Tomasulo's algorithm	Read pp. 184-196.	

Schedule (2)

4 4/20	R10000-style dynamic scheduling (a physical register pool)	The Smith/Sohi article on superscalars. The R10000 article. Read from Register mapping, p. 32 through Register files, p. 35.
	Pentium-style dynamic scheduling (reorder buffers)	Read pp. sections 3.7, 3.10 and 3.13. Two articles on the Pentium Pro (ps, pdf) and (ps, pdf).
5 4/27	Software techniques to exploit ILP	Pp. 304-314 cover 2 techniques that we have already discussed. If you want our authors' take on them, this is the place to read. Pp. 329-340 cover compiler techiques that we will discuss briefly.
	VLIW machines	Read pp. 315-319. Read section 4.7 and the IA-64 papers. In the HP/Intel architecture paper, omit the memory model, software pipelining, & floating-point again, IA-432 compatibility & machine resources per port. In the MFR paper, read only the first 4 pages. There is also a critique by a rival.
	Hardware support for compiler speculation	Read pp. 345-351.
	Wrap-up	Read section 4.10.
6 5/4	Basics of caches	Read pp. 390-410, 423-430.
	Advanced caching techniques	Read pp. 410-413, section 5.4, pp. 430-435, sections 5.6, 5.7.
	Main memory	Read sections 5.8, 5.9.

Schedule (3)

- 5/11	**Tera-style multithreading**	Read the Tera paper (PS, PDF). Tera's runtime system (not required - this is just in case the GSRCT students are interested).
	Simultaneous multithreading	Read section 6.9 and the SMT paper.
8 5/18	**Overview of multiprocessing**	Read section 6.1.
	Cache coherence, snooping and directory protocols	Read sections 6.3 - 6.6.
	Synchronization	Read section 6.7.
9 5/25	**Catch up with whatever has been left undone.** **Discussion of the final.** **Course evaluations.**	
10 6/1	**Current architecture research at UW CSE: guest speakers.**	**Andrew Petersen on Quantum Computing.** WaveScalar FPGAs **Margaret Martonosi on ZebraNet**
6/8	**Scheduled final at 8:30pm to 10:20pm. Clearly, we'll choose another time. I'll be at the architecture conference Sunday - Wednesday, June 5 - June 8. Douglas will proctor the exam.**	