CSE 548 Syllabus

Winter 2004

Instructor
Mark Oskin, oskin@cs.washington.edu
Office: 564 *The Beautiful* Paul Allen Center
Office Hours: Open-door / by-appointment

Teaching Assistance
Douglas Low, douglas@cs.washington.edu
Office Hours: by-appointment

Lecture: Monday/Wednesday 3:00 - 4:20 EE1 031

Text: Computer Architecture: A Quantitative Approach, 3rd Edition
by Hennessy and Patterson

This book will be heavily augmented by supplemental reading material. The book is a good reference and some reading will come from it.

Class format: This class is taught as an interactive discussion. Prior to class each day there will be a reading assignment posted. We (you and Douglas and I) read this material *prior* to class and then come to lecture and talk about it. It will take a week or two to adjust to this format, but this is the third time I've taught this class this way, and believe me, its just good times! Architecture is, if it is anything, a debatable field. Every hack has its benefits and its downside. There hardly ever is a really correct answer, so it makes for a lively discussion. We will try and keep these discussions at a high level, but at times we may need to dwell into details to thoroughly explore some things.

Reading: You will be expected to read a lot. In addition to some of the huge tome (H&P) you can expect to read 1-3 research papers a week. Along with the reading will be a brief assignment. This assignment is to jump start the in-class discussion.

Project: In this class you are to design and engage in a small research project in computer architecture. You will have extensive freedom to choose the project topic, but there will be expectations on the scope.

Quizzes: This is a "quals" course; thus, we will have a couple of tests. In the past, I have given essay tests. I may continue with this practice, or move to a take home exam, or something else.

Grading: 40% project, 10% project presentation, 10% in-class participation, 10% homework, 15% quiz 1, 15% quiz 2.

General Late Policy: Anything, unless otherwise specified, may be turned in 1 day late so long

as you email myself and Douglas an excuse, in the form of a Haiku, **prior** to the original due date. Haiku's must either be true excuses, or at the very least funny. Convincingly true, but amusing excuses are preferred.

Special Late Policy: Once a quarter, anything written, can be turned in up to a week late. Prior to the original deadline, however, you must send a *true and funny* (even ironically funny will do) excuse, formatted as a Villanelle (a French form of poetry that made its way into English in the 1800's).

Email 1: If you ask me a question that has implications for the rest of the class, or is generally interesting, I may post the response to the class list server.

Email 2: You should join the class list server, good stuff will appear there now and then.

Email 3: I do not keep regular hours. If you email me past 8pm, I hope to be asleep. I have been known not to sleep at all so you may get responses at strange hours of the day. Generally though, if you email me and cc Douglas, you will get some kind of response in 24 hours, hopefully far less.

Email 4: Please turn in *everything* electronically by email by sending it to Douglas and cc'ing me. Please do not send it to just one of us. Generally speaking, Douglas tracks everything, but I like to read them too.

Research deadlines/Sickness/Other Serious Matters: Contact me. We will try and work around them.

Academic Accommodations: (the following is a form statement DSS asks us to include, but the short of it is contact us, we'll work with you): To request academic accommodations due to a disability, please contact Student Services, 448 Schmitz, (206) 543-8924 (V/TTY). If you have a letter from Disabled Student Services indicating that you have a disability that requests academic accommodations, please present the letter to me so we can discuss the accommodations you might need in this class.

Topics: A *tentative* list of subjects:

- Evolution / Revolution and Reincarnation (of instruction sets)
- Instruction Level Parallelism: scoreboarding, Tomasulo's algorithm, VLIW/EPIC
- Memory Hierarchy (caches, fetching)
- Characteristics of applications (locality, predictability, parallelism)
- Technology and scaling implications on architectures
- Multitheading, Multiprocessing
- Dataflow
- Embedded computing
- Case studies (P4, Alpha, Power4, USIII)
- Nanotechnology, quantum computing