CSE 378 Spring 2010

Computer Organization and Assembly Programming

Course Staff:		email	Office
	Luis Ceze, instructor,	luisceze@cs,	CSE 576
	Aaron Miller, TA,	ajmiller@cs	TBA
	Jacob Nelson, TA ,	nelson@cs	TBA

Meetings:			
	Lectures	MWF, 9:30-10:20,	EEB 037
	Section AA,	Tue, 2:30-5:20,	CSE 003 lab
	Section AB,	Th, 2:30-5:20,	CSE 003 lab

Book: David Patterson and John Hennessy, ***Computer Organization and Design***, 4th ed. There is substantial overlap with the 3rd edition, of course, and you should be ok if you have it, but there were significant updates in the 4th ed. and we recommend you use that.
A Note about international editions: generally, purchasing and using the international version of this book is fine for this class. Homework assignments are given by the US edition, however, and numbering for problems is different between the international and US editions.

Grading: 30% Labs, 20% Homework, 20% Midterm, 25% Final, 5% Class participation.

Due dates: All deadlines are at 5pm on the date stated.

Late policy: Any assignment (except the midterm and final) can be turned in late *at most 2 days* without penalty as long as the *total* number of late days *does not exceed 5*. Exceeding 5 late days (totaled over all assignments) will decrease your grade by 10% per extra late day for the late assignment. Any assignment turned in late more than 2 days can not be accepted, we need this limit to be able to discuss assignment in class in a timely fashion.

Homework: There are about 4 homework assignments in this class. You can expect that some will include programming assignments. The programming assignments will be mostly on writing programs in MIPS assembly and running them using a program called "SPIM", which is a MIPS simulator. You can download your own copy of SPIM and install it on your computers, or use the one pre-installed on the department machines. There also will be book-work assignments. Programming assignments that are not in MIPS assembly can be written in any high-level language.

Labs: This class has a significant lab component, and as such is a good portion of your grade. There are 4 labs. Finishing each lab is likely to require all the time we give for it. START THE LAB SOON AFTER WE HAND IT TO YOU. THEY MIGHT BE TIME CONSUMING. You also will need time outside the scheduled lab meetings to complete these projects – do not expect to get them done in one afternoon per week. These labs will walk you through the design and construction of a pipelined MIPS processor. When you finish these labs, you will have designed a fully working MIPS processor, capable of executing arbitrary C code, and one that runs on real hardware inside of an FPGA. It'll be fun! Trust us. You should

work on the lab in pairs. While you can work alone on it, given the *substantial* time commitment required, we *highly* recommend you work with a partner.

Exams: There will be a midterm and a final. We expect that if you attend class, keep up with reading, and are doing well in the lab, then the exams should go smoothly.

Cheating: Don't go there; it'll end your academic career.

Course Topics

- basic computer organization
  - CPU, memory, I/O
  - representation of data
- performance metrics for computer systems
  - execution time, CPI, MIPS, MFLOPS
- instruction set design
  - registers
  - arithmetic-logical instructions
  - load-store instructions and operand addressing
  - flow of control instructions
- instruction encoding
  - instruction formats
  - RISC vs. CISC
- translation of HLL program into assembly
  - register, user stack, static data area, heap
  - procedure call conventions
- how a program is compiled and executed
- processor implementation
  - datapath/control
  - pipelining
  - ideal pipeline
  - data hazards & forwarding
  - control hazards & branch prediction
- instruction-level parallelism
- memory hierarchy
  - caches, cache organizations
  - performance metrics for caches, taxonomy of cache misses
  - parameters for cache design
  - write strategies
  - virtual memory, page tables, TLBs
- exceptions/interrupts (interaction with operating system)