EC513: Computer Architecture – Fall, 2009

Basics

Instructor: Martin Herbordt, PHO 333
Office Hours: MW 3-5 and by appointment
Phone: x3-9850
Email: herbordt@bu.edu
Webpage: http://courseinfo.bu.edu/courses/09fallengec513_a1

Mission Statement: "What you need to know about computation logic to build ASICs, FPGAs, and Embedded Systems.''

Course Description: Principles of computer architecture and design, including computer arithmetic and ALU design, instruction sets, CPU design, memory hierarchies, and I/O systems. Case studies.

Prerequisites: For undergraduates, EC413 (Computer Organization). For graduate students, high-level language programming and basic knowledge of computer organization and assembly language. For both, knowledge of operating systems will be useful.

Texts: Hennessy & Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann, 2007. Other readings TBD.

Course Mechanics

- **Classes:** The overall class style will be lecture/exam. Since the class is likely to be small, I'm counting on there being substantial interaction!
- **Grading:** Exams: 70%
  Homework Assignments: 15%
  Final Project: 15%
- **Exams:** There will be two mid-term exams and a final**.** Exams are open textbook and open notes.
- **Homework:** There will be eight to nine homework assignments. You are encouraged to work together to learn the material and to discuss approaches to solving homework problems. However, *you must come up with and write -up the solutions on your own.*
- **Programming Assignments:** There may be a few programming assignments (up to 3) related to probing and analyzing computers.
- **Final Project:** There will be a project. The purpose is to add depth in one particular area of what is a broad course. Possibilities are doing a performance evaluation study, writing a report on a design issue, creating a design, etc. This will vary substantially depending on the class size and on your interests.

Administration

- **Office Hours:** Office hours are tentatively MW 3-5. The best time to catch me otherwise is right after class; the worst time is right before class! On rare occasion I may be unable to keep office hours, so please contact me before traveling a long way to meet me.
- **Email:** You are required to periodically check your email since this is the way many assignments will be distributed. Questions via email are always good. If the solution has general interest, I will broadcast it to the class (leaving the questioner anonymous); if the solution is very involved, we may

need to go over it in person. Also, please check your email for unexpected occurrences like errors in assignments, cancellations, etc.

- **Course Web Site:** The CourseInfo site is given above. I use it to post class notes, lab and homework assignments, and other course information.
- **Incompletes:** Incompletes will be granted only in accordance with university policy, which (broadly) requires a major non-academic crisis near the end of the semester.
- **Course Notes:** I hand out lecture slides before class. I strongly recommend annotating them during class.
- **Academic Honesty:** Please read the university academic honesty policy. If something is not clear, then ask. In particular, plagiarism is regarded as a serious offence and students engaging in this activity will be reported.
- **HW Solutions:** I post solution sketches to HW assignments, although with great reluctance. I believe that solutions are great for quickly finding minor problems, but otherwise a very bad way to learn material. If you are having difficulty, the best way is to see me outside of class.

Keys to Success in this (and most other) Course(s)

- Attendance. You must come to class. Much of the material (and perspective) in this course will be found nowhere else.
- Do the readings. Work out the examples as you read. If you are not positive that you understand something completely, try inventing and solving your own problems.
- Take notes. In particular, annotate the lecture notes during class.
- Participate. Ask questions; talk with your fellow students. Be active.
- Keep up. Before each class (at least), read over the notes from the previous class.
- Allocate enough time! Much of the material is time-consuming to master. There is a big difference between "kind of" understanding a subject and "really" understanding it. In computer architecture there are many topics that interact in subtle ways; mastery of any one requires a good understanding of all of them.
- How do you know that you know the material? A good metric is whether you would feel comfortable standing in front of a class explaining it. Another is whether you think that you could explain it to a job interviewer!

Course Objectives

Review

- MIPS computer organization

Learn about

- Computer Organization – what the parts of the computer are and how they work
- Computer Architecture – the design trade-offs within and among the components, and how to evaluate those trade-offs.
- Components to a degree that you are able to evaluate them with respect to Hardware/Software design decisions in embedded systems

Gain experience with

- Evaluating CPUs and other computer system components
- Evaluating Computer Systems

Week	Class	Date	Topic	HW	Readings	Handouts
1	1	9/3	Administration Computer Architecture Overview			
2	2	9/8	CA 1: Positional Numbering Systems Adders	hw1 out		
	3	9/10	CA 2: Multiplication 1 (FPL)			
3	4	9/15	CA 3: Multiplication 2	hw1 due		
	5	9/17	CA 4: Division (basic only) Floating Point	hw2 out		
4	6	9/22	CA 5: Floating Point	hw3 out		
	7	9/24	Performance Evaluation	hw2 due		
5	8	9/29	Review MIPS -- assembly language and toy CPUs	hw3 due hw4 out		
	9	10/1	Instruction set design, part 1			
6	10	10/6	Instruction set design, part 2	hw4 due hw5 out		
	11	10/8	Pipelining 1,2: definitions, performance, ideal pipeline Pipelining 3: Data hazards with no stalls	hw5 due		
7		10/13	***COLUMBUS DAY***			
	12	10/15	**EXAM 1**			
8	13	10/20	Pipelining 3: Data hazards with stalls, Pipelining 4: Begin control hazards	hw6 out		
	14	10/22	Pipelining 4: Finish control hazards Pipelining 5: Exceptions and exception handling			
9	15	10/27	Pipelining 5: Finish Exceptions Pipelining 6: R4000: 64-bit, multiple units	hw6 due		
	16	10/29	Pipelining 6: longer pipeline Pipelining 7: Dataflow machine, Ideal Scheduling			
10	17	11/3	Pipelining 7: Dynamics scheduling with a scoreboard Pipelining 8: Renaming-based techniques			
	18	11/5	Review Pipelining 8: Tomasulo's algorithm	hw7 out		
11	19	11/10	Pipelining 8: Speculative execution, ROB Pipelining 9: Superscalar			
	20	11/12	Pipelining 10: Branch Prediction	hw7 due hw8 out		
12	21	11/17	Cache 1: Review			
	22	11/19	Cache 2: Finish Review Overview, cache size/program chars, block size	hw8 due		
13	23	11/24	**EXAM 2**			
		11/26	***THANKSGIVING***			
14	24	12/1	Cache 3: finish associativity			
	25	12/3	Cache 4: code optimization multilevel caches, reducing hit time	hw9 out		
15	26	12/8	Cache 5: reducing hit time, summary Virtual memory			
	27	12/10	I/O	hw9 due		
16						
		12/16	**FINAL EXAM -- 9-11**			