Digital Electronics & Computer Engineering (E85)

Harris **Fall 2007**

Syllabus

Teaching Staff

Professor:	David Money Harris	Parsons 2374	x73623	David_Harris@hmc.edu
Lab Assistants:	Corina Tom			Corina_Tom@hmc.edu
	Tom Donze			Thomas_Donze@hmc.edu
	Anu Kohli			Anu_Kohli@hmc.edu

David Money Harris

Corina Tom

Tom Donze

Anu Kohli

Schedule

Lecture:	MW 11-12:15
Office Hours:	TBD
Lab Hours:	Saturday 2-4, Sunday 12-2 in Parsons B183
TBΠ Tutor Hours:	TBD

Feel free to stop by even if I do not have official office hours. One of the main reasons that I teach at Harvey Mudd is that I value working with students 1-on-1 or in small groups.

Text

Harris & Harris, *Digital Design and Computer Architecture*, Morgan Kaufmann 2007.

Electronic Communication

Class web page:	http://www3.hmc.edu/~harris/class/e85
Class email list:	eng-85-l

Be sure to check that you are on the class email list. You should have received email before the beginning of classes. If you did not receive mail, add yourself to the list or risk missing important late-breaking announcements. To subscribe, send email to listkeeper@hmc.edu with one line in the body:

subscribe eng-85-l

You also will need a Harvey Mudd College computer to complete your labs. If you are not a HMC student, email me your full name and school affiliation and I will request an account for you.

Course Objectives

Digital systems have revolutionized our world. From television to cell phones to GPS to warfare to medicine to automobiles, computers and digital processing have reshaped the way we live and work. Computers are also a vital part of daily practice in every field of science and engineering.

Previous generations of engineers learned the "nuts and bolts" of the profession by doing things like disassembling and rebuilding engines. As technology has advanced, cars have become too complicated for the layperson to work on. Ironically, the same advances have made computers much easier to build. While most fields of engineering require extensive mathematics and complicated analysis of even rather simple components, digital systems merely require counting from 0 to 1. Their challenge, instead, is in combining many simple building blocks into a complex whole. Field programmable gate arrays (FPGAs), containing the equivalent of thousands or millions of logic gates, make it possible to build these complex systems in the lab without the tedium of manually connecting components. In this class, you will build your own microprocessor and test it on a FPGA. In the process, you will master the art and science of digital design. You will learn to speak to and control processors in their native tongue, assembly language. And you will put all the pieces together to demystify how a computer works.

As you probably know, very few complex systems work the first time you put them together. Engineers must become good at systematically and efficiently debugging their creations. One of the course objectives that can be frustrating but vitally important is to learn to teach oneself professional-strength computer-aided design tools and to use these tools to debug systems.

By the end of this course, a successful student will be able to:

- design and debug combinational and sequential digital circuits using schematics and Verilog
- program in MIPS assembly language
- build a microprocessor

Grading

Labs:	30%
Problem Sets:	20%
In-Class Activities:	5%
Midterm:	15%
Final:	30%

The only way to really master the material in this class is to design a microprocessor. The labs in this class build upon each other until you design your own 32-bit MIPS microprocessor in Labs 9-11. You **must** complete these labs and demonstrate a working microprocessor to pass this class.

Solutions to the labs and problem sets from previous semesters are undoubtedly floating around campus and on the web. You may **not** refer to solutions while doing the assignments; they must be your own work. Many of the labs build on previous labs. If you are sick or do not turn in a lab, you may refer to the solutions handed out to complete the lab when it is needed for a subsequent lab. However, you may not simply copy another student's files.

Labs and homework are due by the end of class and will not be graded if submitted late. However, even if you do not complete your microprocessor on time, you must still submit it before the final exam to pass the class. Your lowest lab and problem set score will be dropped before the average is calculated. You are welcome to discuss labs and problem sets with other students or with the instructor or lab assistants or tutors **after** you have made an effort by yourself. However, you must turn in your own work, not work identical to that of another student. Be sure to credit at the top of your assignment anyone with whom you discussed ideas. **It is an honor code violation to simply copy someone else's work.**

On a regular basis, there will be a short in-class activity related to a recent lecture. You are strongly encouraged to come to regularly attend class, review your notes before class, and ask questions during class. If you stay on top of the material, you should have no difficulty doing well with these activities. The two lowest scores will be dropped.

Tentative Schedule

The attached schedule is a tentative plan that may adjust during the semester. The deadlines, however, are fixed unless otherwise notified; *do not assume* that they will change just because the lecture schedule changes.

Lecture	Date	Topics	Readings	Assignment
0	9/5	Introduction: digital abstraction, number systems, logic gates	1.1-1.5, A1-A4	
1	9/10	Static discipline, CMOS transistors	1.6-1.9, A5-A7	
10	9/12	Combinational logic design	2.1-2.7	PS 1 due
11	9/17	Timing, sequential circuits	2.8-2.10, 3.1-3.2	Lab 1 due
100	9/19	Finite state machines	3.3-3.4	PS 2 due
101	9/24	Dynamic discipline, metastability	3.5	Lab 2 due
110	9/26	Hardware description languages: Verilog	4.1-4.3,	PS 3 due
111	10/1	Verilog, Part II	4.4-4.9	Lab 3 due
1000	10/3	Arithmetic circuits	5.1-5.2	PS 4 due
1001	10/8	Fixed and floating point number systems	5.3	Lab 4 due
1010	10/10	Sequential building blocks, Memory arrays, Logic arrays	5.4-5.7	PS 5 due
1011	10/15	Transmission lines	Appendix A.8	Lab 5 due
	10/17	Midterm		
	10/22	-- Fall Break: no class --		
1100	10/24	MIPS instruction set and registers	6.1-6.3	
1101	10/29	Branches, Procedure calls	6.4-6.5	
1110	10/31	Linking and launching applications	6.6-6.7	PS6 due
1111	11/5	Single-cycle processor datapath	7.1-7.3.1	Lab 6 due
10000	11/7	Single-cycle processor control	7.3.2-7.3.4	PS 7 due
10001	11/12	Multicycle processor	7.4	Lab 7 due
10010	11/14	Exceptions	7.7	PS 8 due
10011	11/19	Pipelining	7.5.1-.75.2	Lab 8 due
10100	11/21	Pipeline hazards and stalls	7.5.3-7.5.5	PS 9 due
10101	11/26	Caches	8.1-8.3	Lab 9 due
10110	11/28	Virtual memory	8.4	
10111	12/3	Memory-mapped I/O	8.5	Lab 10 due
11000	12/5	Advanced architecture: a sampler	7.8	PS 10 due
11001	12/10	Case study: IA-32 Processors	5.8, 7.9, 8.6	Lab 11 due
11010	12/12	Class summary and review		