Digital Electronics & Computer Engineering (E85)

Harris **Fall 2010**

Syllabus

Teaching Staff

Professor:	David Money Harris	Parsons 2374	x73623	David_Harris@hmc.edu
	Patrick Little	Parsons 2375C	x71712	Patrick_Little@hmc.edu
Lab Assistants:	Kevin Chan			Kevin_G_Chan@hmc.edu
	Alex McAuley			Alexander_G_McAuley@hmc.edu

Schedule

Lecture:	MW 1:15 – 2:30
Office Hours:	M 4-5, T 2-3, Th 10-11
Lab Hours:	Saturday 2-4, Sunday 7-9 in Parsons B183
ΤΒΠ Tutor Hours:	Mon, Tue evening

Feel free to stop by even if we do not have official office hours. One of the main reasons that we teach at Harvey Mudd is that we 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 and use them to control the world. 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 C and 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.

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.

Honor Code Policy

1. All students enrolled in this course are bound by the HMC Honor Code. More information on the HMC Honor Code can be found in the HMC Student Handbook.
2. It is your responsibility to determine whether your actions adhere to the HMC Honor Code. If this document does not clarify the legitimacy of a particular action, you should contact the course instructor and request clarification.
3. Work you submit for individual assignments should be your own, and you should complete all assignments based on your own understanding of the underlying material. If you work with, or receive help from, another individual on an assignment, provide a written acknowledgement in complete sentences that includes the person's name and the nature of the help.
4. This document is not meant to be an exhaustive list of every possible Honor Code violation. Infractions not explicitly mentioned here may still violate the Honor Code.
5. Boundaries of Collaboration

   Verbal collaboration with other students on individual assignments is encouraged AFTER you have given serious thought to each component yourself. However, all submitted written work should be written by yourself individually, and not a collaborative effort or copied from a common source (e.g., a chalkboard). It is NOT acceptable to work on labs in lockstep with another classmate.
6. Use of Published Solutions

   You may check your answers against the solutions in the back of the textbook after completing problems, but may not reference step-by-step solution instructions in separately published solution manuals.
7. Use of Computer Software

   The use of graphing calculators and computer software to aid in course work is acceptable, as long as it does not substitute for an understanding of the course material.
8. Use of Web Resources

   The use of Internet resources to aid in course work is acceptable, as long it does not substitute for an understanding of the course material. Plagiarism and direct copying from online (or any other) sources is strictly prohibited. You may NOT refer to solutions to textbook problems floating around on the Web.
9. Use of Your Own Work from Previous Semesters

   If you have previously attempted this course, you may resubmit your work from previous semesters as this semester's coursework, as long as you understand the underlying material.
10. Use of Other Course Resources from Previous Semesters

    You may reference the tests of this course from previous semesters as study aids. You may not reference assignments (labs, problem sets, activities) of this course from previous semesters.
11. Retention of Course Resources

    Exams from this course may be committed to dorm repositories or otherwise used to help future students.

Schedule

Lecture	Date	Topics	Readings	Assignment
0	9/1	Introduction: digital abstraction, number systems, logic gates	1.1-1.5, A1-A4	
1	9/6	Static discipline, CMOS transistors	1.6-1.9, A5-A7	
10	9/8	Combinational logic design	2.1-2.7	PS 1 due
11	9/13	Timing, sequential circuits	2.8-2.10, 3.1-3.2	Lab 1 due
100	9/15	Finite state machines	3.3-3.4	PS 2 due
101	9/20	Dynamic discipline, metastability	3.5	Lab 2 due
110	9/22	Hardware description languages: Verilog	4.1-4.3	PS 3 due
111	9/27	Verilog, Part II	4.4-4.9	Lab 3 due
1000	9/29	Arithmetic circuits	5.1-5.2	PS 4 due
1001	10/4	Fixed and floating point number systems	5.3	Lab 4 due
1010	10/6	Sequential building blocks, Memory arrays, Logic arrays	5.4-5.7	PS 5 due
1011	10/11	Transmission lines	Appendix A.8	Lab 5 due
	10/13	Midterm		
	10/18	-- Fall Break: no class -- END of E85A		
1100	10/20	Midterm Discussion, Intro to C		
1101	10/25	C Programming: control structures and data types		
1110	10/27	C Programming: functions, arrays, pointers		PS6 due
1111	11/1	MIPS instruction set and registers	6.1-6.3	Lab 6 due
10000	11/3	Branches, procedure calls, addressing	6.4-6.7	PS 7 due
10001	11/8	Single-cycle processor datapath	7.1-7.3.1	Lab 7 due
10010	11/10	Single-cycle processor control	7.3.2-7.3.4	
10011	11/15	Multicycle processor	7.4	Lab 8 due
10100	11/17	Exceptions	7.7	PS 8 due
10101	11/22	Slack		Lab 9 due
10110	11/24	Caches	8.1-8.3	PS 9 due
10111	11/29	Memory-mapped I/O	8.5	Lab 10 due
11000	12/1	Advanced architecture: a sampler	7.8	PS 10 due
11001	12/6	Case study: IA-32 Processors	5.8, 7.9, 8.6	Lab 11 due
11010	12/8	Class summary and review		

Notes:
Problem sets 8 and 9 are pushed back 1 week from the original syllabus
Pipelining is deleted from the course (but will be briefly discussed on 12/1)
One additional lecture is available for C programming and another for spillover in Chapters 6-7

Lab 6: Intro to C Programming
Lab 7: C Programming: Temperature Control
Lab 8: Assembly Language Programming: Fibonacci Numbers & Floating Point Addition
Lab 9: MIPS Single-Cycle Processor
Lab 10: MIPS Multicycle Processor: Test bench and controller
Lab 11: MIPS Multicycle Processor: Datapath & Memory