Digital Electronics & Computer Engineering (E85)

Harris **Spring 2017**

Syllabus

Teaching Staff

Professor:	David Money Harris	Parsons 2374	x73623	David_Harris@hmc.edu
Lab Assistants:	Kitty Belling			kbelling@g.hmc.edu
	Vai Viswanathan			vviswanathan@g.hmc.edu

David Money Harris

Kitty Belling

Vai Viswanathan

Schedule

Lecture:	MW 8:10 or 9:35 am Shan 3481
Office Hours:	TBD
Lab Hours:	Saturday 2-4, Sunday 12-2 in Parsons B183
TBΠ Tutor Hours:	Monday 7-9 pm, Platt Austin Chun & Robert Cyprus

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 and in small groups.

Text

Harris & Harris, *Digital Design and Computer Architecture, ARM Ed.*, Morgan Kaufmann 2016.

Electronic Communication

Class web page:	http://pages.hmc.edu/harris/class/e85
Class email list:	eng-85-l@g.hmc.edu

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:

- Build digital systems at all levels of abstraction from transistors through circuits, logic, microarchitecture, architecture, and C culminating with implementing and programming a microprocessor soft core on a field programmable gate array.
- Manage complexity using the digital abstraction, data types, static and dynamic disciplines, and hierarchical design.
- Design and implement combinational and sequential digital circuits using schematics and hardware description languages.
- Program a commercial microcontroller in C and assembly language and use it in a physical system.
- Begin the practice of implementing and debugging digital systems with appropriate lab techniques including breadboarding, interpreting datasheets, and using field-programmable gate arrays and microcontroller boards, simulators, debuggers, and test-and-measurement equipment.

Grading E85

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

E85A

Labs:	30%
Problem Sets:	20%
In-Class Activities:	5%
Midterm:	45%

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. 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 short in-class activities 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

Lecture	Date	Topics	Readings	Assignment
0	1/18	Introduction: digital abstraction, number systems, logic gates, HDL	1.1-1.5, A1-A4, 4.1-4.2.2	
1	1/23	Static discipline, CMOS transistors	1.6-1.9, A5-A7	
10	1/25	Combinational logic design	2.1-2.8	PS 1 due
11	1/30	Timing, sequential circuits	2.9-2.10, 3.1-3.2	Lab 1 due Digital Circuits
100	2/1	Finite state machines	3.3-3.4	PS 2 due
101	2/6	Dynamic discipline, metastability	3.5-3.7	Lab 2 due Comb Logic
110	2/8	Hardware description languages: Verilog	4.1-4.3	PS 3 due
111	2/13	Verilog, Part II	4.4-4.10	Lab 3 due Structural FSM
1000	2/15	Arithmetic circuits	5.1-5.2	PS 4 due
1001	2/20	Fixed and floating point number systems	5.3	Lab 4 due Behavioral FSM
1010	2/22	Sequential building blocks, Memory arrays, logic arrays	5.4-5.7	PS 5 due
1011	2/27	Datasheets	A.1	Lab 5 due Building blocks
	3/1	Midterm		
1100	3/6	C Programming	C.1-C.7	
1100	3/8	C Programming	C.8-C.11	
	3/13	SPRING BREAK: No class		
	3/15	SPRING BREAK: No class		
1101	3/20	Microcontrollers: Memory-mapped I/O	9.1-9.3.3	Lab 6 due C
1110	3/22	Parallel & serial interfacing	9.3-9.4	PS6 due
1111	3/27	ARM assembly language	6.1-6.3.6	Lab 7 due C I/O
10000	3/29	Function calls, machine language	6.3.7-6.9	PS 7 due
10001	4/3	Single-cycle processor datapath	7.1-7.3.1	Lab 8 due C Peripherals
10010	4/5	Single-cycle processor control	7.3	PS 8 due
10011	4/10	Multicycle processor	7.4	Lab 9 due Assembly
10100	4/12	Pipelining	7.5.1-2	PS 9 due
10101	4/17	Pipeline hazards and stalls	7.5.3-4	Lab 10 due Multicycle Proc
10110	4/19	Advanced architecture: a sampler	7.7	PS 10 due
10111	4/24	Case study: ARM processors	6.7, 8.7, 8.5	Lab 11 due: Multicycle Proc
11000	4/26	Class summary and review		