E85: Digital Electronics & Computer Engineering

Spring 2020

Revised Syllabus

Teaching Team

Professors:

David Harris	Parsons 2374	David_Harris@hmc.edu
Josh Brake	Parsons 2364	jbrake@hmc.edu

Lab Assistants:

Peter Johnson	pjohnson@g.hmc.edu
Pinky King	pking@g.hmc.edu
Jane Watts	jwatts@g.hmc.edu

Schedule

All times are Pacific time zone.

Lecture:
MW 8:10 am - 9:25 am Shan B460 https://zoom.us/j/602578040
Lectures will also be recorded and posted online for you to access after class.

Office Hours:
Brake: Mon. 9:30 am - 10:30 am, Tue. 1 pm - 2 pm, or by appointment https://zoom.us/j/968153826
Harris: Wednesday at 1 or by appointment https://zoom.us/j/628716473

Grutor Lab Hours:
Peter: Friday, 10 am
Pinky: Friday, 12 pm
Jane: Saturday, 2 - 4 pm PT

TBΠ Tutor Hours: Accessible via email:
Caleb Norfleet (cnorfleet@g.hmc.edu): Fast responses on Monday, 4:30 - 5:30 pm PT
Veronica Cortes (vcortes@g.hmc.edu)
Ethan Falicov (efalicov@g.hmc.edu): Fast responses on Monday, 7 - 8 pm PT

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 and in small groups.

Text

You will get the most out of class if you do the reading before lecture. See the schedule for recommended reading. Copies of the textbook are available in the lab and in the Engineering lounge.

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-85-1-2020-sp@g.hmc.edu, eng-85a-1-2020-sp@g.hmc.edu

You also will need a Harvey Mudd College computer to complete your labs. If you are not an 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 yourself professional-strength computer-aided design tools and to use these tools to debug systems.

By the end of this course, you should 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:	20%*
Problem Sets:	30%*
Midterm:	17%
Final:	33%

E85A

Labs:	30%
Problem Sets:	20%
Midterm:	50%

* Note: The weighting of the labs and problem sets has been modified to account for the removal of the embedded systems labs.

Lab 11 is the capstone of the labs, in which you design and simulate a microprocessor, drawing on most of the skills you have acquired over the semester.

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 do not turn in a lab, you may refer to the solutions handed out to work through the lab you missed to learn the skills needed for a subsequent lab. However, you may not simply copy another student's files.

Labs and homework are due online via Sakai by 8:10 am PT on their due date. If you are having trouble submitting an assignment on time due to extenuating circumstances, you may request an extension from Prof. Brake via email. We understand that completing your coursework remotely will likely create challenging circumstances from time to time and so we will liberally grant extensions given (1) you provide us with a reason for needing an extension and (2) you are making a good faith effort at completing the assignment. Please also include a new date by which you would like to submit the assignment. This extension policy replaces the original one-week extension policy. Your lowest problem set and lab grade over the course of the semester will still each be dropped. No late work will be accepted after May 1st.

Pass/No Credit is available on written request no later than April 29 if the student has health, family, network, or other difficulties that prevent them from giving the class their normal level of attention. As a reminder, a C- or better equivalent grade would become Pass, and a D+ or lower would count as No Credit.

Labs 10 and 11 are now optional. If you do choose to submit them, they will only be counted if they improve your already existing lab grade.

You are strongly encouraged 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. For labs, asking classmates or tutors for help when you are stuck on a specific issue is encouraged (especially on difficulties with the tools and equipment), but sitting at adjacent computers and working through the lab together in lock-step is specifically prohibited. Pair/group programming is also prohibited. 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.**

With the move to online coursework and with the above collaboration guidelines in place, we strongly encourage you to set up virtual study groups with your classmates. If you do not have an already established study group, you may want to email the class (eng-85-1-2020-sp@g.hmc.edu) to see if there are others in the class that may wish to study with you.

Readings for each lecture are listed on the schedule below. Many students say they have found the readings valuable and enjoyable. You'll get the most out of the class if you read the sections in advance of the lecture and come with questions, and then reread as necessary when you work your problem sets and labs.

Tentative Schedule

Lecture	Date	Topics	Readings	Assignment
0	1/22	Introduction: digital abstraction, numbers	1.1-1.5	
1	1/27	Logic gates, Static discipline, transistors	1.6-1.9, A1-A7	
10	1/29	Combinational logic design	2.1-2.8	PS 1 due
11	2/3	Timing, sequential circuits	2.9-2.10, 3.1-3.2	Lab 1 due Digital Circuits
100	2/5	Finite state machines	3.3-3.4	PS 2 due
101	2/10	Dynamic discipline, metastability	3.5-3.7	Lab 2 due Comb Logic
110	2/12	Hardware description languages: Verilog	4.1-4.3	PS 3 due
111	2/17	Verilog, Part II	4.4-4.10	Lab 3 due Structural FSM
1000	2/19	Arithmetic circuits	5.1-5.2	PS 4 due
1001	2/24	Fixed and floating-point number systems	5.3	Lab 4 due Behavioral FSM
1010	2/26	Sequential building blocks, arrays	5.4-5.7	PS 5 due
1011	3/2	Catchup / Midterm Review		Lab 5 due Building blocks
	3/4	Midterm		
1100	3/9	C Programming	C.1-C.7	
1101	3/11	C Programming	C.8-C.11	
	3/16	Spring Break! NO CLASS		
	3/18	Spring Break! NO CLASS		
1110	3/23	Extended Spring Break		
1111	3/25	Extended Spring Break		
10000	3/30	Assembly language	6.1-6.3.5	
10001	4/1	Function calls	6.3.6	PS 6 due
10010	4/6	Machine language	6.4-6.9	Lab 6 due C Programming
10011	4/8	Single-cycle processor datapath	7.1-7.3.1	PS 7 due
10100	4/13	Single-cycle processor control, Verilog	7.3, 7.6	
10101	4/15	Multicycle processor	7.4	PS 8 due
10110	4/20	Pipelining	7.5.1-2	
10111	4/22	Advanced architecture: a sampler	7.7	PS 9 due, Lab 10 (optional)
11000	4/27	Case study: Processors	6.7, 8.7, 8.5	
11001	4/29	Class summary and review		PS 10 due, Lab 11 (optional)