E85: Digital Electronics & Computer Engineering

Spring 2022

Syllabus

This syllabus supersedes anything on the MOOC except due dates.

Teaching Team

Professors:	David Harris	David_Harris@hmc.edu	
E85 Class:		TuTh 1:15-2:30 pm	Shan B460
Office Hours:		M 2:35-3:55 (Digital Lab), Tue 2:35-3:55 (Digital Lab)	

Class and office hours will use the following Zoom meeting while virtual (at least first 2 weeks)
https://hmc-edu.zoom.us/j/93680599145?pwd=dXRsZGZhWTUySFVOa1YwV1Z2YjFFZz09

Tutoring:	Sat 7-8, Sun 7-8, Mon 7-8, Wed 4:15-5:15	
	Kevin Kim	kekim@g.hmc.edu
	Madeline Masser-Frye	mmasserfrye@g.hmc.edu
	Kevin Wang	kewan@g.hmc.edu

Tutoring will use the following Zoom link while virtual, and then meet in the Digital Lab.
https://hmc-edu.zoom.us/j/96498694822

ТВП: Ben Bracker bbracker@g.hmc.edu Tue 7-8
https://hmc-edu.zoom.us/j/96498694822

Text

Harris & Harris, *Digital Design and Computer Architecture, RISC-V Ed.*, Morgan Kaufmann 2021.

Electronic Communication

Class web page: http://pages.hmc.edu/harris/class/e85
Class email list: eng-85-1-2022-sp@g.hmc.edu, eng-85a-1-2022-sp@g.hmc.edu
MOOC: You received a link to sign up for the HMC-only version of ENGR85A Digital Design. Make sure you aren't in the self-paced public Digital Design.

I encourage the class to set up a student-only Discord channel, as has been done in the past.

Lab Kits

E85 lab kits include a microcontroller board, breadboard, and various components. The Engineering Department has partially subsidized the kits so your cost is $50. To purchase a board, complete this E85 Kit Payment Form below and then go to the stockroom to pick the board up. If the kit fee presents a financial hardship, complete the E85 Kit Payment Waiver Request; waivers are handled by the staff and are never seen by the instructor.

You can do your lab assignments on the computers in the Digital Lab in the Parsons basement, or on your own computer. You will be using the Mentor Graphics ModelSim, which runs on Windows. For Mac or Linux users, they also run on Windows using Parallels or VirtualBox. For the second half of the class, you'll also be using Segger Embedded Studio, which runs on Windows, Mac, and Linux. Running on your own computer may be more convenient, but if your computer is old or slow, you may find it easier to go to the lab.

If other methods aren't working, you can also access the lab computers from your own computer via Remote Desktop. Directions to reserve a remote computer and access it are at https://docs.google.com/document/d/1oWzKPpmTTOvulhTgIueBPiO-JDj1eXMiZ1bdg2EzDZM/edit

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 levels of abstraction from transistors through circuits and logic.
- Manage complexity using the digital abstraction, static and dynamic disciplines, and hierarchical design.
- Design and implement combinational and sequential digital circuits using schematics and hardware description languages.
- Analyze and trade off performance, cost, and power consumption of digital circuits.
- Begin the practice of implementing and debugging digital systems with appropriate lab techniques including breadboarding and interpreting datasheets.
- Simulate digital circuits

Grading

Lecture Practice:	5%
Class Activities	5%
Labs:	30%
Problem Sets:	25%
Exams:	35%

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.

Assignment deadlines are shown on the Dates tab of the MOOC. Your lowest problem set and lab grade of the semester will each be dropped, as will your three lowest lecture practice assignments. Please ration your drops carefully lest you find yourself ill at the end of the semester and out of options. The class moves quickly and it is difficult catching up if you fall behind. Contact the Associate Dean of Academic Affairs if you have a protracted emergency.

Handling extensions in the MOOC is cumbersome, so I do not accept late lecture practice questions. The dropped lab and problem set are intended to accommodate illness, travel, and personal emergencies. However, if you have already used your drop and need flexibility on a second lab or problem set, please contact the instructor at least two days in advance for an extension, or work with the Academic Deans after the fact if it was impractical to request an extension in advance.

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. 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. **It is an honor code violation to simply copy someone else's work.**

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 class time and come with questions, and then reread as necessary when you work your problem sets and labs.

This class is flipped in that you will see the content first in the MOOC and then work examples in class. The examples are intended to reflect the concepts you need for the problem sets and labs, and previous semester students said that the time spent attending class generally is more than compensated by a reduction in time on the assignments. You will receive 5% of your grade to incentivize participating in the in-class activities.

If you feel that class is not a good use of your time, you can get this credit by doing an independent project instead. One project is required in the first half of the class (E85A) and another in the second half to substitute for the activities. For the first half, your project should involve either Verilog code containing a FSM, or a breadboard with at least one digital component, achieving something useful, artistic, or fun. For the second half, your project should either involve the microcontroller, or involve adding at least 3 significantly different RISC-V instructions (and a test that exercises them) to your multicycle processor Verilog. Complete a writeup not to exceed 1 page of text covering the functions, principles of operation, and results, plus appropriate schematics and/or code sufficient for somebody to understand and replicate your project. Turn in each project no later than the midterm or final exam for credit.

Tentative Schedule

Lecture	Date	MOOC Topics Due	Class Activity	Readings	Assignment
0	1/18	Introduction	Welcome: numbers	1.1-1.5	
1	1/20	Gates	***soldering	1.6-1.9, A1-A7	
10	1/25	Combinational logic design		2.1-2.8	PS 1 due
11	1/27	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	2/3	Dynamic discipline, metastability		3.5-3.7	Lab 2 due Comb Logic
110	2/8	Verilog, Part I		4.1-4.3	PS 3 due
111	2/10	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	2/17	Fixed and floating-point numbers		5.3	Lab 4 due Behavioral FSM
1010	2/22	Sequential building blocks, arrays		5.4-5.7	PS 5 due
1011	2/24	Midterm Review			Lab 5 due Multicycle Control
	3/1	Slack			
	3/3	Midterm			
1100	3/8	C Programming		C.1-C.7	
1101	3/10	C Programming		C.8-C.11	
	3/15	Spring Break			
	3/17	Spring Break			
1110	3/22	Microcontrollers: Memory-mapped I/O		9.1-9.3.3	PS 6 due
1111	3/24	Parallel & serial interfacing, ADCs		9.3	Lab 6 due C Programming
10000	3/29	I/O libraries and examples		9.3	PS 7 due
10001	3/31	Assembly language		6.1-6.3.6	Lab 7 due Simon
10010	4/5	Function calls, machine language		6.3.7-6.9	
10011	4/7	Single-cycle processor datapath		7.1-7.3.1	Lab 8 due SPI
10100	4/12	Single-cycle processor control, Verilog	Single cycle Verilog	7.3, 7.6	PS 8 due
10101	4/14	Multicycle processor	Add mc instructions	7.4	Lab 9 due Sound & Light
10110	4/19	Pipelining	Pipelined execution	7.5.1-2	PS 9 due
10111	4/21	Advanced architecture: a sampler		7.7	
11000	4/26	Advanced RISC-V Features			PS 10 due
11001	4/28	Class summary and review	Final Review		Lab 10 due Multicycle CPU
	5/13	Take home exam due 5 pm			