

Homework 7

Due	May 18 by 11:59pm	Points	100	Submitting	a file upload
File Types	circ, pdf, txt, ys, yo, hex, xls, xlsx, c, h, s, zip, gz, tar, and tgz			Available	after May 9 at 11:59am

Hello World!

Submit online via Canvas.

We accept only the following file formats:

- .circ
- .pdf
- .txt
- .ys
- .hex
- .xls, .xlsx
- (and zip/tar/gz of the above)

Resources:

- [A sample microarchitecture skeleton for the Y88 \(https://ssl.cs.dartmouth.edu/~sws/cs51-s15/hw7/y86-uarch-skeleton.zip\)](https://ssl.cs.dartmouth.edu/~sws/cs51-s15/hw7/y86-uarch-skeleton.zip) (with sample shims, ALU, CND, register file---although you don't have to use ours).
 - We recommend you use "tick once" to tickle clocks, so that the fast rises when the slow does
 - Also, be sure to ook at the internals of the isplitter
- [the latest update of the FSM graph \(https://ssl.cs.dartmouth.edu/~sws/cs51-s15/hw7/y86fsm.pdf\)](https://ssl.cs.dartmouth.edu/~sws/cs51-s15/hw7/y86fsm.pdf)
- On the class [resources page](#), you'll find [RAM and ROM Tools](#) (to let you turn .yo files and Excel spreadsheets into RAM and ROM files for LogiSim).

Note that as you go through the below, you might need to iterate a bit on your circuit. If you can do it cleanly, please submit *one* copy of the circuit, your ROM code, etc, rather than one for each question.

Q1: 30 pts

Take our sample circ (or yours), and off the right, wire in a few kilobytes of RAM and your memory-mapped IO. You'll need to make copies of the tunnels hanging of the shims, and work out how they dance with your RAM and IO "fake RAM".

(Note that our FSM won't have the two shims assert RAMuse simultaneously.)

Q2: 20pts

Build the FSM control ROM circuit sketched in class (e.g., [here \(https://ssl.cs.dartmouth.edu/~sws/cs51-s15/18-uarch-pipeline/slides.pdf\)](https://ssl.cs.dartmouth.edu/~sws/cs51-s15/18-uarch-pipeline/slides.pdf)), and wire that into your Y86. Include the microsequencer you built in Homework 6. (Make sure you connect your "current state" register to reset!)

Q3: 40pts

Work out the control ROM!

- Create a spreadsheet, with a row for each state and a column for each bit you need in the microinstructions or such.
- (Label the rows and columns!)
- Use the fsm2hex tool to program this into your ROM
- Keep in mind that our FSM chart does not show what should happen on reset

(Please include the spreadsheet and the .hex file)

Q4: 10pts

Write a simple "hello.js" that sends "hello world!" to the display. Assemble it, install it in your ROM, and watch it run! Be amazed.

Fix bugs, if necessary. (Please include the hello.js and the .hex file)

Hwk 7 rubric		
Criteria	Ratings	Pts
Q1. Only writes to physical RAM when the address is in range		5 pts
Q1. Correctly forwards RAM controls view longer description		15 pts
Q1. Clarity, etc.		10 pts
Q2. Correctness of flow view longer description		10 pts
Q2. Clarity, etc.		10 pts
Q3. Installing microcode to FSM		3 pts
Q3. Instructions view longer description		27 pts
Q3. Clarity view longer description		10 pts
Q4. Writing hello.js		4 pts
Q4. Providing the .hex file view longer description		2 pts
Q4. Clarity, etc.		2 pts
Q4. It runs!		2 pts
		Total Points: 100