Homework 6

Due May 10 by 11:59pm	**Points** 100	**Submitting** a file upload	**File Types** circ, pdf, txt, ys, hex, xls, xlsx, zip, tar, and gz
Available after May 4 at 11:59pm			

In the first part of the term, we worked from electricity up to combinatorial logic circuits. We've since spent some time looking at assembly language and the ISA, the foundational level of software. In the next assignments, you will bridge the gap between these two!

- This week: you will build some missing pieces you will need for your Y86.
- Later: you will
  - plug these into a skeleton dapapath circuit we provide (with shims and a regfile you can use, if you don't want to use yours)
  - wire in the FSM control
  - write a "hello world" program in assembly...
  - load it into LogiSim ROM
  - and see it run!

Q1 IO

Implement (memory-mapped) IO by connecting the LogiSim TTY and keyboard components---and installing some glue so they behave as the KBDR/KBSR and DDR/DSR we discussed in class. That is:

- Build a KBDR, KBSR, DDR, DSR
- that, on the one side, are reachable by the processor as memory addresses in the right places (with DMemReq, DMemWrite, DMemReady)
- and that, on the other side, connect to LogiSim TTY/kb components
- such that they exhibit the correct handshaking, with respect to status bits, displayed characters, and such.
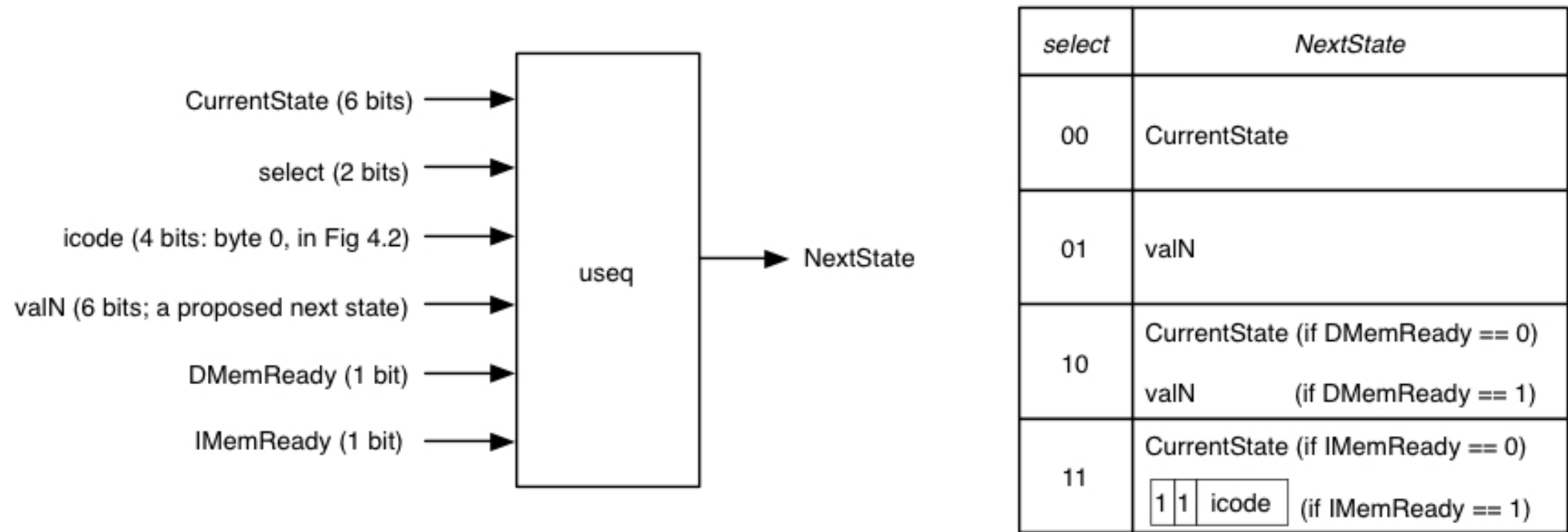
Q2: Microsequencer

You will be building (and then, later, extending) an FSM to drive your Y86.

A 6-bit field should more than enough space to specify the current state.

In the FSM we will build up, most of the time, each state has exactly one "next" state. We can hardcode that into the control ROM. However, there are also exceptions:

- Sometimes, we want to self-loop if DMemReady is not set
- Sometimes, we want to self-loop if IMemReady is not set, but then to branch to one of about 12 states, based on the icode field.

A **microsequencer** is a circuit that helps you calculate next-state transitions like this. Build this microsequencer for your Y86:

select	NextState
00	CurrentState
01	valN
10	CurrentState (if DMemReady == 0) valN (if DMemReady == 1)
11	CurrentState (if IMemReady == 0) 1 1 icode (if IMemReady == 1)

Don't use any internal registers; the outputs should follow directly from whatever the inputs are.

Criteria	Ratings	Pts
Q1. Reasonable attempt		20 pts
Q1. Testing		5 pts
Q1. Correctness **view longer description**		25 pts
Q1. Clarity, neatness, simplicity		25 pts
Q2. Correctness		20 pts
Q2. Clarity, neatness, simplicity		5 pts
		Total Points: 100