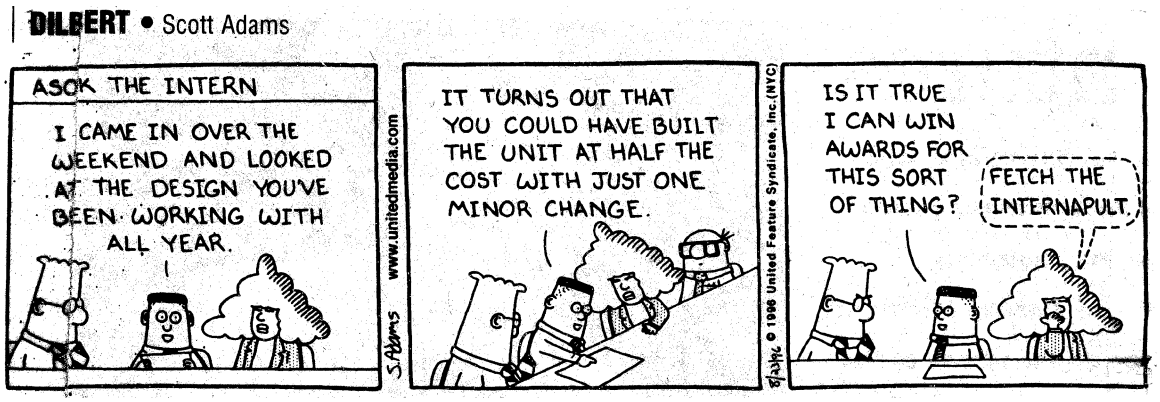
Introduction to Computer Engineering (E85)

Harris Spring 2001

Problem Set 8 **Due: Friday, April 6**

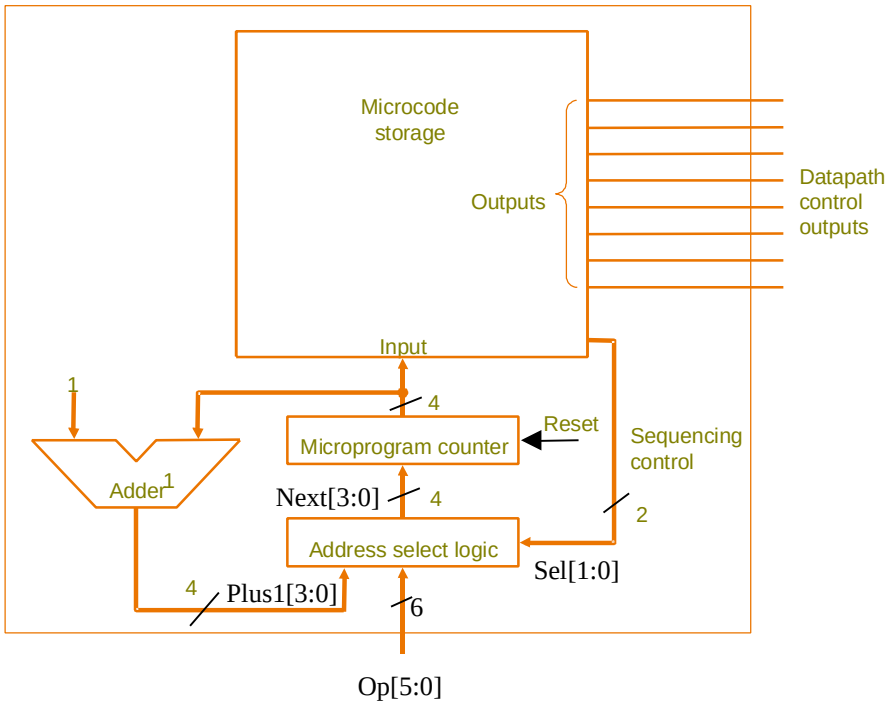


1) Enhancing the MIPS Processor

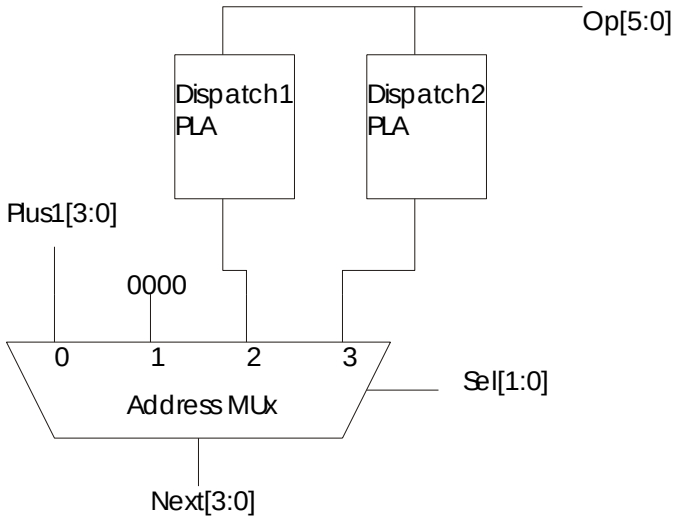
Modify the multi-cycle processor datapath and control FSM to handle the `lb` instruction. Diagrams are attached for your convenience.

2) Microprogramming

Ben Bitdiddle is building a multi-cycle MIPS processor. He's decided to build his cbox control unit with microprogramming, as described in Section 5.5. The cbox receives Reset and Op[5:0] (from the top 6 bits of the instruction) and is responsible for producing all of the control signals to the datapath. The overall architecture is shown below:



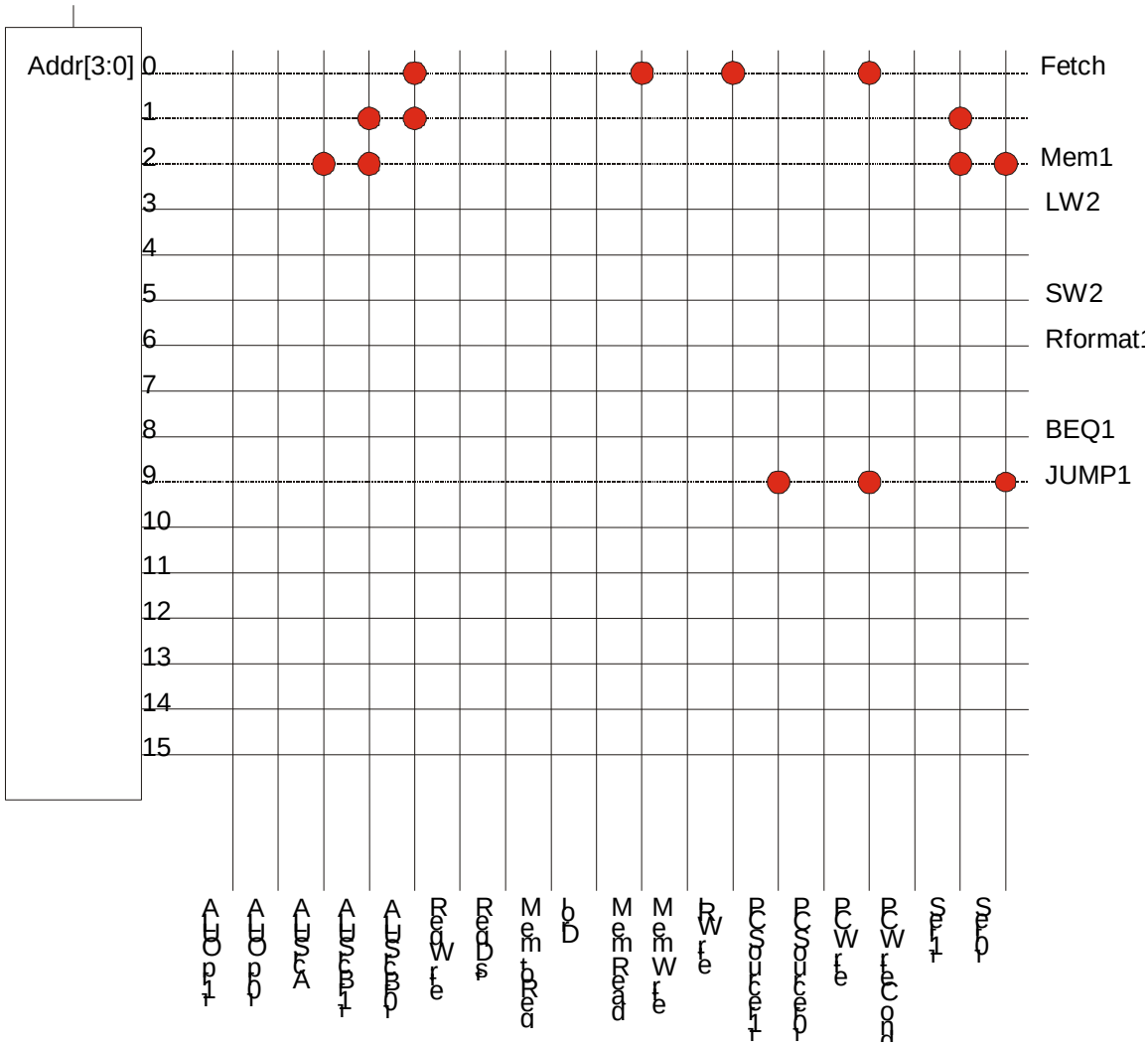
The microprogram counter is just a 4-bit register with reset. The microcode storage is a 16 word x 18 bit ROM producing 16 bits of control signals for the datapath and 2 bits of sequencing control used by the Address Select Logic. The Address Select Logic, shown below, is responsible for computing the next value of the microprogram counter. This next value may be Seq, Fetch, Dispatch1, or Dispatch2, as listed in Figure 5.46. Depending on the sequencing control lines Sel[1:0], the next address is chosen from the old microprogram counter plus 1, zero (to go back to fetch), or the outputs of two PLAs containing the dispatch logic.



Help Ben by designing the ROM and two DISPATCH PLAs.

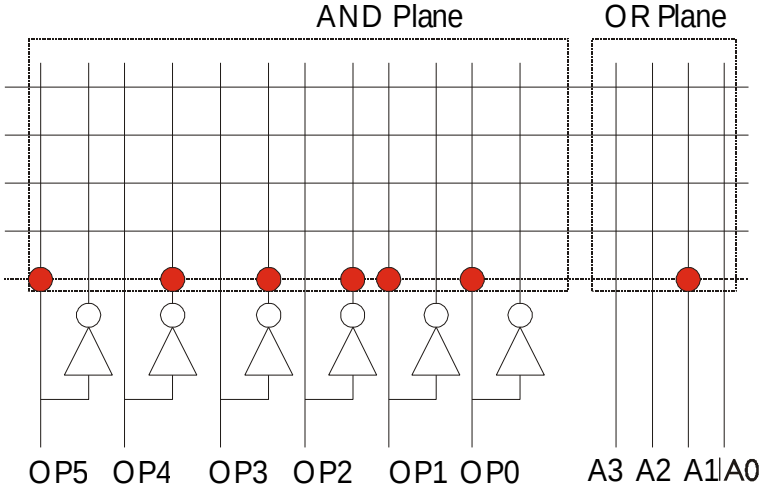
a) Microcode ROM

Refer to Figures 5.42 and 5.46 to complete the Microcode ROM below. Place a dot on the entries where the ROM should contain 1's. The dotted rows have been completed for you. Since the microcode program is fewer than 16 words long, the last six rows will be empty. A copy of the ROM is on the last page for your convenience.


Addr[3:0]
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Fetch
Mem1
LW2
SW2
Rformat1
BEQ1
JUMP1

b) Dispatch1 PLA

The Dispatch1 PLA uses the 6 bits of the Op field to jump to the appropriate line in the microcode program. The dispatch could be constructed as a 64 word by 4 bit ROM. The six bits of Op could choose one word of the ROM containing the 4-bit destination in the microcode program for that Op. However, this would be wasteful of space because very few instructions are actually implemented, so most of the words in the ROM would be blank. A more efficient approach is to use a PLA. The AND plane of the PLA decodes the instruction. The OR plane produces the microcode address given the instruction. Since we handle five types of instructions (LW, SW, R-format, BEQ, and JUMP), the PLA only requires five minterms. Complete the PLA below. The first minterm for LW (Op =100011) has been done for you; the Dispatch1 table jumps to MEM1 (address 0010) to process LW. A copy of the PLA is on the last page for your convenience.



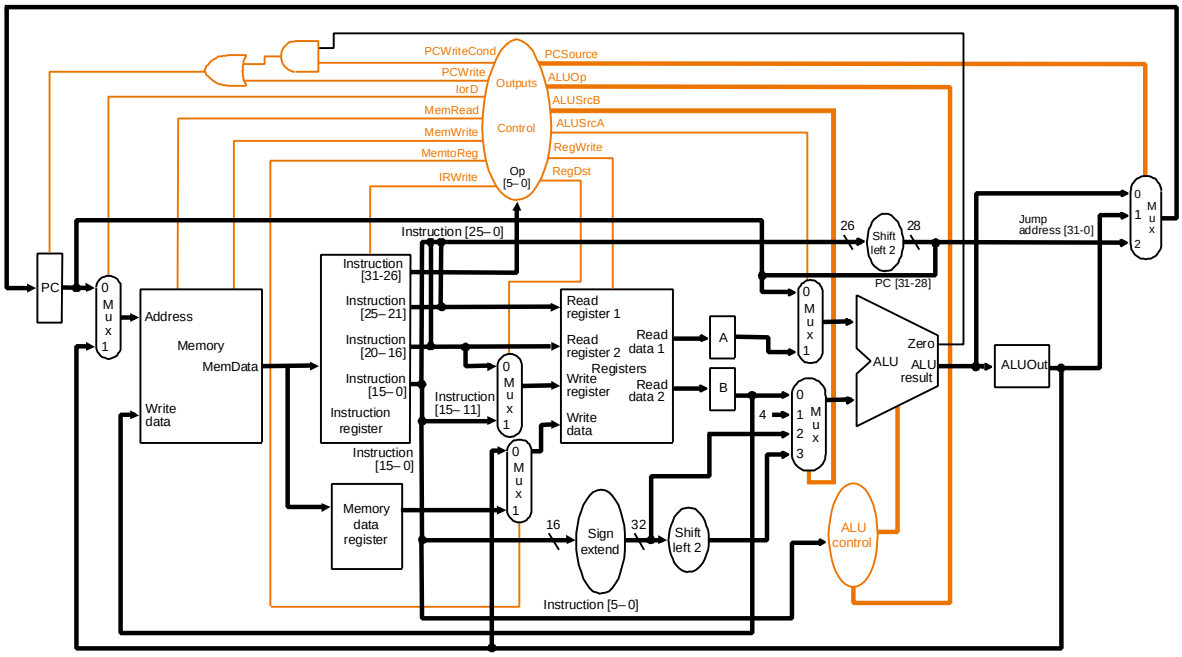
c) Dispatch2 PLA

Finally, the Dispatch2 PLA also uses 6 bits of Op to jump to either LW2 or SW2. Again, it could be constructed as a ROM, but the PLA is more efficient. Sketch a circuit for the Dispatch2 PLA on a separate sheet of paper.

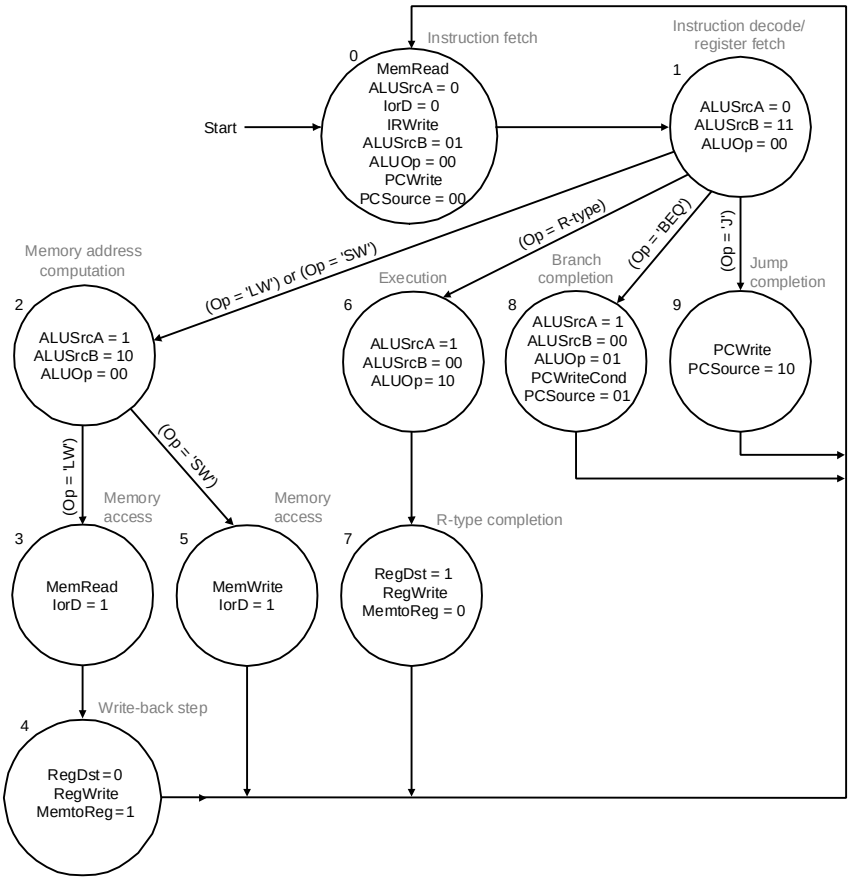
3) Time

Please indicate how many hours you spent on this problem set. This will not affect your grade, but will be helpful for calibrating the workload for next semester's class.

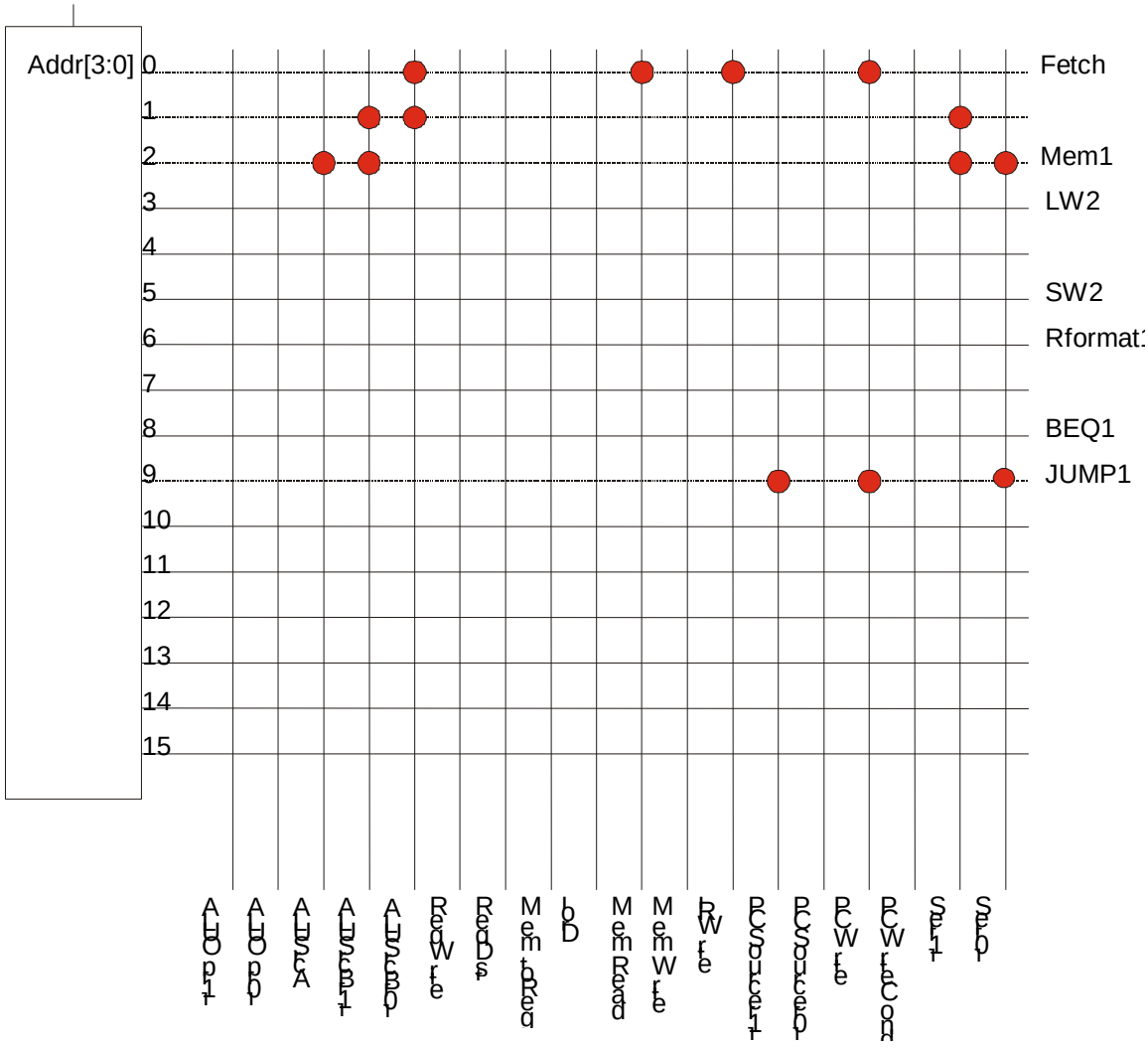
Multicycle Datapath (Problem 1)



Multicycle FSM (Problem 1)



Microcode ROM



Dispatch1 PLA