|	
Intro. to Computer Architecture	**Homework 3**
CSE 240 Autumn 2006	**Due: Wed. 4 October 2006**

Write your answers on these pages. Additional pages may be attached (with staple) if necessary. Please ensure that your answers are legible and *show your work*. Write your name at the top of each page. Due at the ***beginning of class***. Total points: 51.

1. [14 Points] **Combinational Logic Circuits and 2s-complement integers.** In this problem we will construct gate-level logic circuits to determine whether: A is zero, A is negative, A is positive, A equals B, A is greater than B, and A is less than B. For each part, below, assume the inputs A (A_3, A_2, A_1, A_0) and B (B_3, B_2, B_1, B_0) represents a 4-bit 2's complement integer (with A_0 and B_o representing the least significant digits A an B, respectively). You may use any logic gates discussed in the textbook (including XOR and gates with more than two inputs, *e.g.,* n-input AND gates).

   (a) Output Z is 1 if and only if the input A is zero. Construct the gate-level logic circuit to produce output Z from the input. Be sure to label your inputs (A_3, A_2, A_1, A_0) and output (Z).

   (b) Output N is 1 if and only if the input A represents a negative number. Construct the gate-level logic circuit to produce output N from the input. Be sure to label your inputs and output. (Hint: This may not sound like a hint, but... This one is *really* as easy as you think!)

   (c) Output P is 1 if and only if the input A represents a positive number. Construct the gate-level logic circuit to produce output P from the input. You may use the logic circuit for Z and N from above by drawing a box with "Z?" or "N?" in it (with four inputs and one output). Be sure to label all your inputs and the output.

(d) This circuit has three outputs. Output E is 1 if and only if A is equal to B. Output L is 1 if and only if $A < B$. Output G is 1 if and only if $A > B$. Construct the gate-level logic circuit to produce these three outputs (E, L, G) from the inputs $(A_3, A_2, A_1, A_0, B_3, B_2, B_1, B_0)$. Be sure to label your inputs and output. As before, you can use the logic functions by drawing a box with "Z?", "N?", or "P?" in it. **You may also use an adder and/or a incrementer when constructing this circuit.**

(e) Although the above circuit calculates "A equal B", it does so inefficiently. Create a new circuit for calculating just "A equal B" *without* using an adder or incrementer by using bit-wise comparison. (Hint: XOR is your friend.) As before, be sure to label your inputs and output.

(f) Give two reasons why this second implementation of "A equal B" is greatly preferred to the first implementation, especially if A and B were, say, 64-bit numbers?

2. [4 Points] **Memory.** Memories come in many shapes and sizes. If we need a memory of a particular size and addressability, we can often construct it out of other, differently-structured memories. Construct a 2^4-by-1-bit memory (below, left) using a mux and a 2^2-by-4-bit memory (below, right), as well as any gates you might need. Assume that the memories are read-only memories (ROMs), so you need only be concerned with reading. Be sure to label the inputs ($A[0]$ - $A[3]$) and output (D).

A[3] →
A[2] →
A[1] →
A[0] →

D

A[1] →
A[0] →

D[3] D[2] D[1] D[0]
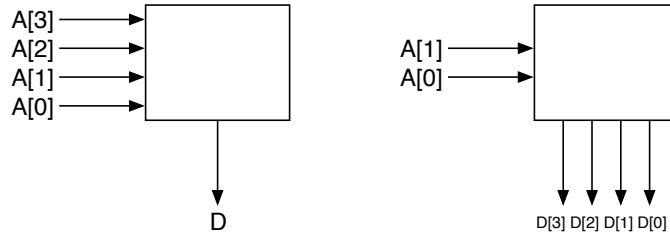
3. [4 Points] **State Machines I.** Consider the following sequential circuit implementing a state machine. I and O are the input and output, respectively, and S is a 1-bit D flip-flop (with the initial value of 0). Draw the state diagram describing the circuit. Make sure you label edges (with input) and nodes (with their state value from S and output).

I

O

S
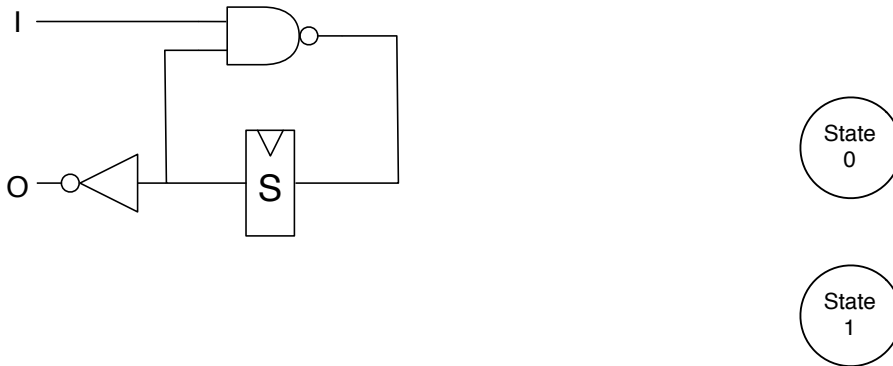
State 0

State 1

4. [12 Points] **State Machines II.** Design a state machine for an elevator for a four-floor building with no basement. To simplify the problem, the system has a single input: I, the floor to which the elevator has been summoned by someone pressing a button, either inside or outside the elevator. (We're simplifying the controller by ignoring the cases in which multiple buttons are pressed.) The desired floor is represented by a 2-bit unsigned number. The state machine has two outputs: (i) a one-bit "move" signal, M (1 represents the elevator should move, 0 represents the elevator should not move) and (ii) a one-bit "direction" signal, D (1 represents up, 0 represents down).

The state element, F, is a 2-bit unsigned number that encodes the current floor of the elevator (initially zero representing the ground floor). **One clock cycle corresponds to the time it takes the elevator to move one floor**.

(a) Complete the state machine diagram (below, right) for the elevator control state machine. Unlike the other state machine diagrams you've drawn thus far, this machine's outputs are on the arcs (and not on the nodes). Therefore, label the edges with a "input / M, D" (For example, "1 / 1, 0"). The D output is ignored whenever M is zero, so put a mark the D for those states with an "X". You may label an arc with multiple input/output pairs.

(b) Complete the truth table for the state machine. If the D output is ignored whenever M is zero, so put an mark the D for those states with an "X" (as is done for you for the first line):

$F_{current}$	I	F_{next}	M	D
0	0	0	0	X
0	1			
0	2			
0	3			
1	0			
1	1			
1	2			
1	3			
2	0			
2	1			
2	2			
2	3			
3	0			
3	1			
3	2			
3	3			

State 3

State 2

State 1

State 0

(c) Consider implementing this state machine using a single memory. Doing so would require 16 entries (rows) in the memory and each entry would be 4 bits wide for a total 64 bits. Now consider a version of the state machine for a building with 2^n-floors. In terms of n, how many rows would be required? How wide would the entries be? How many total bits in the memory?

5. [8 Points] **Sequential Logic Circuits I.** Consider the following circuit. Components are labeled according to the following key: (A) A 4-bit register representing an unsigned binary number (initially set to 0). (B) A 4-bit incrementer. (C) An 8-bit register (initially set to 0). (D) An 8-bit adder. (E) A 2^4-by-8-bit memory.
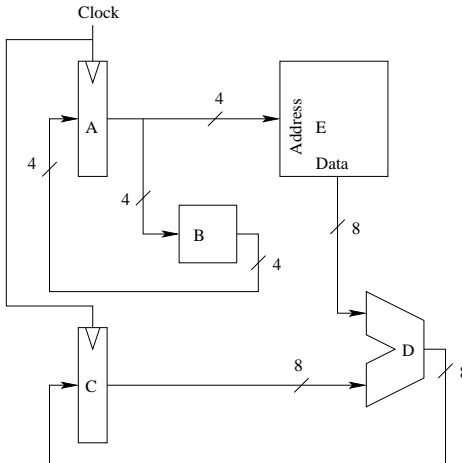
Assume the memory has the following contents:

Address	Contents		Address	Contents
0	4		8	2
1	2		9	2
2	1		10	2
3	3		11	1
4	2		12	6
5	7		13	4
6	1		14	2
7	2		15	1

(a) Determine the values stored in the A and C registers at the end of each clock cycle for 6 cycles. Complete the following table with this information.

Cycle:			0	1	2	3	4	5	6
Register:	*A*		0						
	C		0						

(b) In a sentence, what does this circuit do?

6

6. [8 Points] **Sequential Logic Circuit II.** Consider the following sequential circuit. *A* is a 3-bit register, *M* is a $2^3 \times 8$-bit memory that can be both read and written in a single cycle (like the register file made from flip-flops that we looked at in class), and the boxes labeled "+1" are incrementers. The register and memory are connected to a common clock. *A* initially contains the value 0 and the memory contains 2s-complement signed integers.
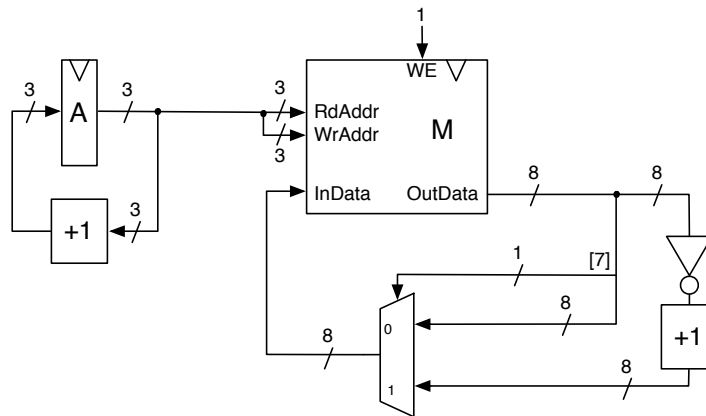
(a) Trace the first 6 cycles of this sequential circuit. Leave a box empty if the value for that location has not been written that cycle (that is, fill in a box only if that location was written that cycle).

Cycle:		0	1	2	3	4	5	6
Register:	*A*	0						
Memory:	0	1						
	1	-1						
	2	0						
	3	9						
	4	-5						
	5	-3						
	6	23						
	7	-17						

(b) In a sentence, what does this circuit do?

Name: _____ 7. [1 Point] **Last and Most Important Question!** Give us your feedback.

    (a) How many hours did you spend on this assignment?

    (b) On a scale of 1-5, how difficult did you find this assignment? (1-easiest, 5-most difficult)

    (c) Do you have any other comments on your experience completing this assignment? What are they?