EECS 270 Midterm Exam 2

Fall 2009

Name: _____ unique name: _____ UMID:_____

Sign the honor code:

I have neither given nor received aid on this exam nor observed anyone else doing so.

_____

Scores:

Problem #	Points
1&2	/11
3	/9
4	/7
5	/7
6	/7
7	/10
8	/8
9	/8
10	/10
11	/6
12	/17
Total	**/100**

NOTES:

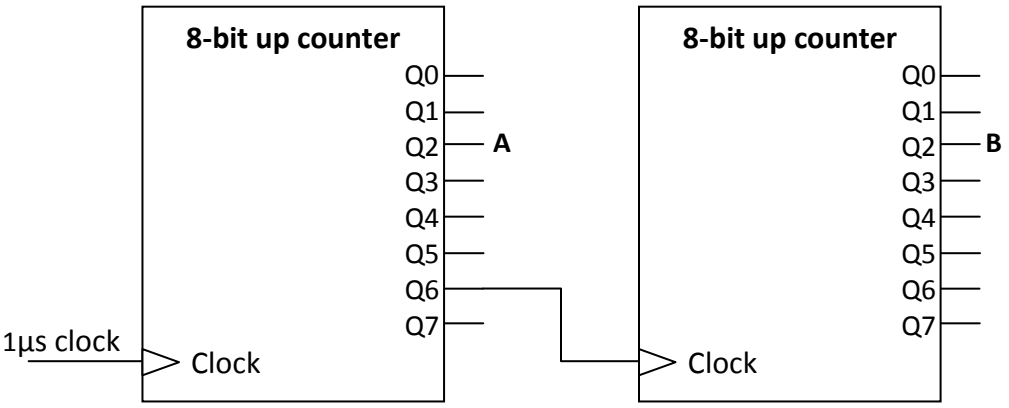
1. Open book and Open notes
2. There are **13** pages total. Count them to be sure you have them all.
3. Calculators are allowed, but no PDAs, Portables, Cell phones, etc.
4. This exam is fairly long: *don't spend too much time on any one problem*.
5. You have about 120 minutes for the exam.
6. **Be sure to show work and explain what you've done when asked to do so.**

1. Multiple choice /fill in the blank. Circle the correct answer or fill in the blank. **[7 points, -2 per wrong or blank answer, minimum 0]**

   - A single bit of ***SRAM/DRAM/ROM*** consists of a transistor and a capacitor.
   - In a 32 by 32 memory array, if the output is to be 4 bit per address, you need

     _____ address lines total, _____ of which will go to the row decoder.
   - In base-2 arithmetic, if you clear all but the four least-significant digits you are: ***subtracting / dividing by / adding / multiplying by /taking the modulo by*** the number_____.
   - 123 base 4 is _____ in base 2 representation.

2. Two 16-bit counters are connected as follows. Assume that these two counters are both working as "counting up" (note that Q7 is the MSB for the output of a counter); also, the period of the input clock signal for the 1st counter (the one on the left-hand side) is equal to 1 µs. Calculate the *frequency* of the output signals at A and B, respectively. Be sure to include units! **[4 points]**



Frequency of A:_____

Frequency of B:_____

3. For the following problem indicate if each statement is “true”, “false” or “unknown if true or false” by writing a T, F or U as appropriate. **[9 points, -2 for each wrong or blank answer, minimum 0]**

   If wxy’ is a prime implicant of the function F(w,x,y,z) then

   a. _____ wxy’z is an implicant of F.

   b. _____ wxyz is an implicant of F.

   c. _____ wy’ is an implicant of F.

   d. _____ wz is an implicant of F.

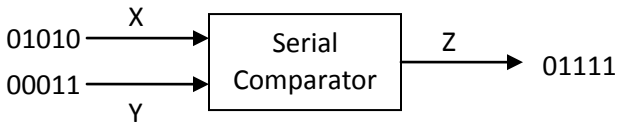
   e. _____ if xy’z’ and wz are prime implicants of F then wxy’ is an essential prime implicant.

   f. _____ if the only other prime implicants of F are y’z’ and w’x’y’ then wxy’ is an essential prime implicant.

4. Find the **<u>minimal *product-of-sums*</u>** for the following function $F=\sum_{W,X,Y,Z}(0,2,3,6,8,9,12)+d(1,11,13)$. Use a K-map and clearly show your work. **[7 points]**

5. For this problem you will be designing a serial comparator. This device will take two numbers as input, with one bit of each number being provided each cycle (most significant bit first). It is then to figure which value is the greater one. More formally:

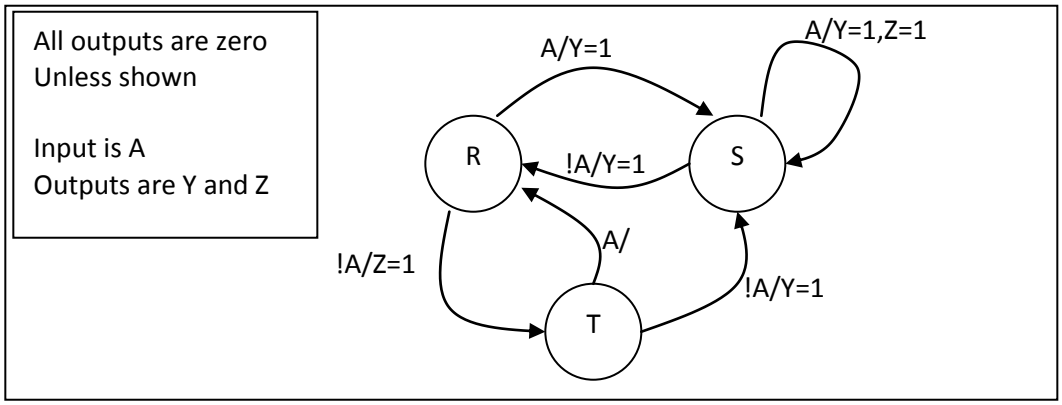
   The machine has three inputs (X, Y and reset) and one output (Z) in addition to the clock. If the reset input is a 1, the output should be 0, otherwise the output will be determined by X and Y. The X and Y inputs are binary numbers given one bit at a time with the most significant bit given first and are valid on the rising edge of the clock (as well as well before and after that edge.) The output Z is to be 1 if the bits of X seen so far are larger (as a binary number) than the bits of Y seen so far (since reset last went low). Consider the sample input and output sequence below (note: the leftmost bits are the oldest!):

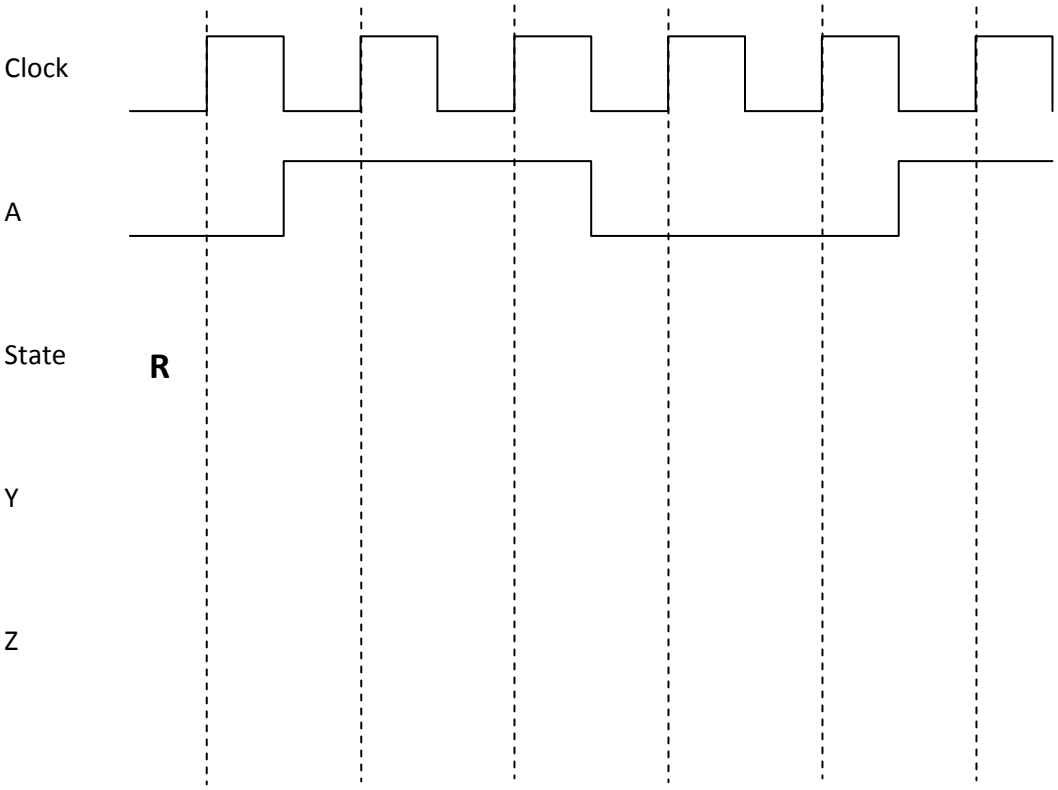


   In this figure first X and Y are both zero, so X is not greater than Y and therefore Z=0. On the next rising edge of the clock X is now 1 (for a running value of 01) and Y is 0 (for a running value of 00) so the running value of X is greater than the running value of Y and therefore Z=1.

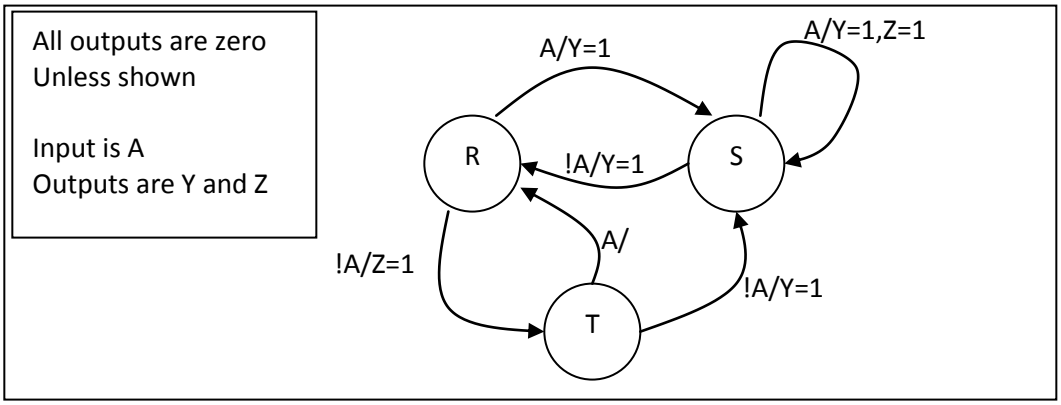
   Design a *Moore-type state diagram* which solves this problem. *To receive more than half credit you must use as few states as possible.* **[7 points]**

6. Complete the timing diagram for the state machine shown below. For the state, simply write the state name. **[7 points]**





7. For this problem, assign state bits S[1:0] as 00 for state R, 01 for state S, and 10 for state T. *Using a K-map*, find the *minimal sum-of-products* for next state (NS[1:0]) and the outputs (Y and Z). You are to assume that any output not shown is zero and that you will never reach the state S[1:0]=11, so you don't care what happens in that case. *You must show your work to get **any** credit!* **[10 points]**



NS1=

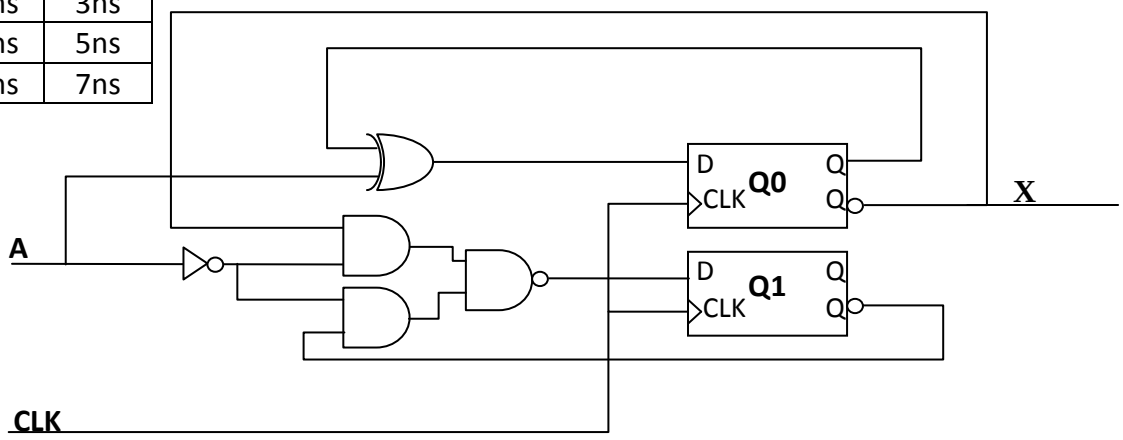
NS0=

Y=

Z=

8. Say you have the following values associated with the process you are using:

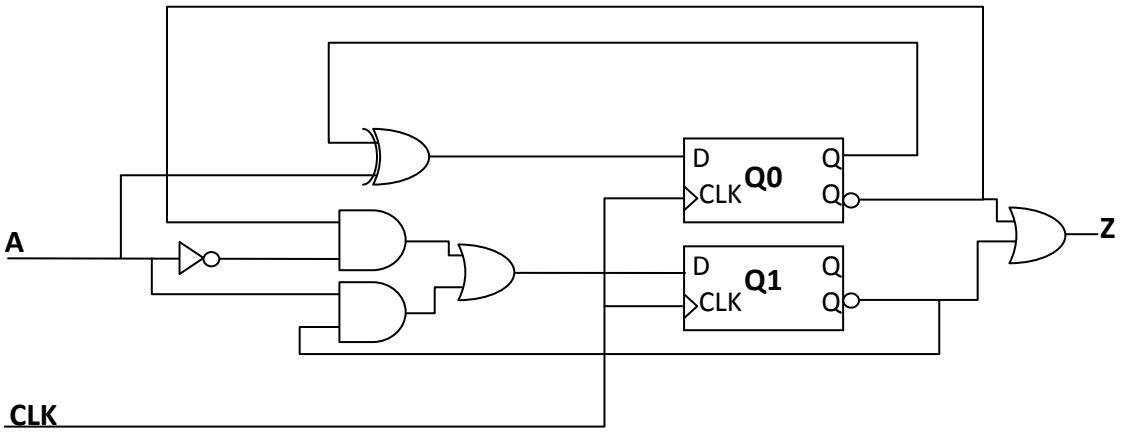
Device	Min	Max
DFF:		
Clock to Q	1ns	4ns
Set-up time	4ns	
Hold time	5ns	
OR/AND	2ns	6ns
NOT	1ns	3ns
NAND/NOR	1ns	5ns
XOR	3ns	7ns



Assume that the input A is coming from a flip-flop that has the same properties as the flip-flops that are shown and is clocked by the same clock.

a. Add inverter pairs as needed to the above figure to avoid any "fast path" problems. Do so in a way that has least impact on the worst-case delay (as a first priority) and which keeps the number of inverter pairs needed to a minimum (as a second priority). **[3 points]**
b. *After* you've made your changes in part a, compute the maximum ***frequency*** at which this device can be safely clocked. *Clearly* show your work. **[5 points]**

9. Draw the state transition diagram which is implemented by the following circuit. **[8 points]**



10. Design a *Mealy-type* state transition diagram for the following problem. There are 2 inputs, X and Y, and one two-bit output M[1:0]. You are to output

    ((# of 1's in X since reset)+(# of 0's in Y since reset))mod 3

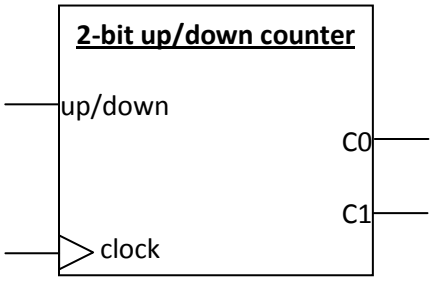
    The "mod 3" simply means to return the remainder when divided by 3. So if there had been 2 ones in X and 5 zeros in Y since reset the output would be one, or M[1:0]=01. The table below shows an example set of inputs and outputs. Just to be helpful, the value (# of 1's in X since reset)+(# of 0's in Y since reset) is listed under *sum* in the table. **[10 points]**

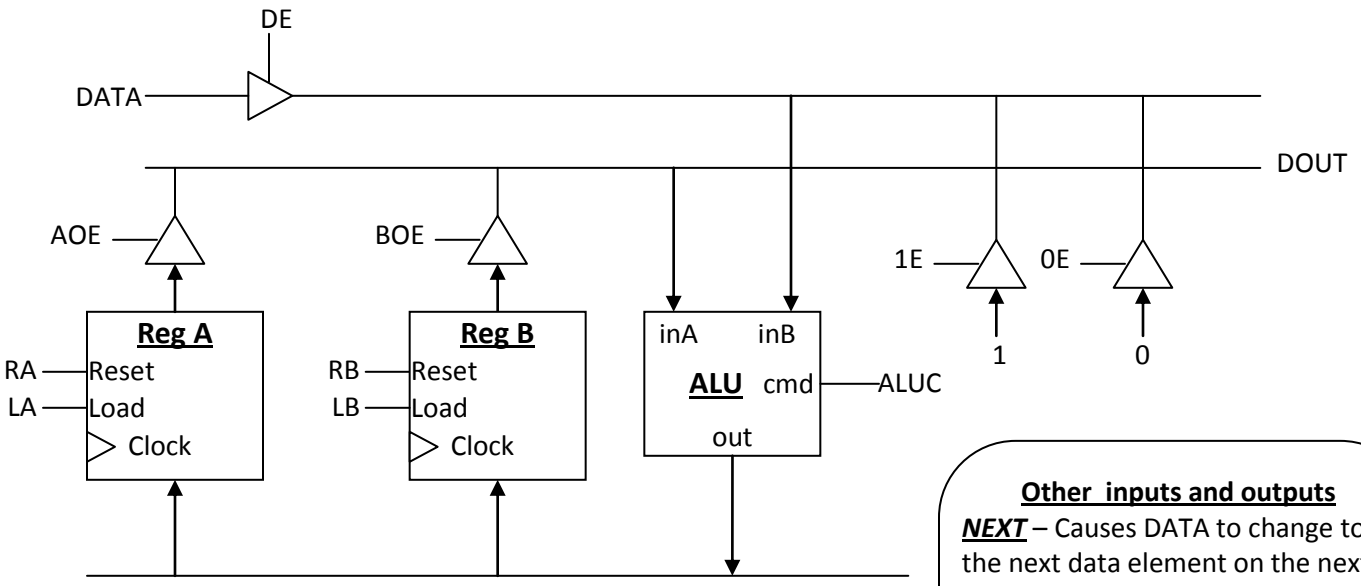
X	0	0	1	0	1	0	1	0	1
Y	1	0	1	1	0	0	1	1	0
Sum	*0*	*1*	*2*	*2*	*4*	*5*	*6*	*6*	*8*
M1	0	0	1	1	0	1	0	0	1
M0	0	1	0	0	1	0	0	0	0

Be very sure that:

- You have clearly labeled your inputs and outputs on the machine.
- You have indicated which state to start in
- Your answer is readable.

11. Using only 2 T-flip flops and no more than 2 standard gates (AND, NAND, NOT, XOR etc.) design a 2-bit up/down counter. The device is to count up when "up/down" is a 1 and down when "up/down" is a 0. C1 is the MSB. **[6 points]**





Other inputs and outputs

NEXT – Causes DATA to change to the next data element on the next rising edge of the clock.

DONE – Indicates the result is available on DOUT. Only needs to be asserted for one clock period.

START – You are to start finding the average.

B4 – True if RegB==4

B8 – True if RegB==8

12. You are to design a state machine which causes the above datapath to find the average of 4 numbers (rounding down). When the START signal goes high, you are to begin to compute the average. Once done, the result should be placed on DOUT and the DONE signal should be set high. START will not go high again until after DONE is asserted.

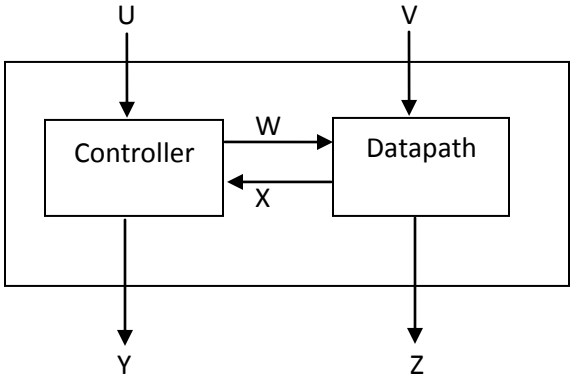
    The data to be averaged is available on DATA. The first element will be there when START is high. Each successive data element will be placed on DATA after any rising edge where the NEXT signal is asserted.

ALU

cmd	Operation
0	out=inA+inB
1	out=inA-inB
2	out=inA<<inB
3	out=inA>>inB

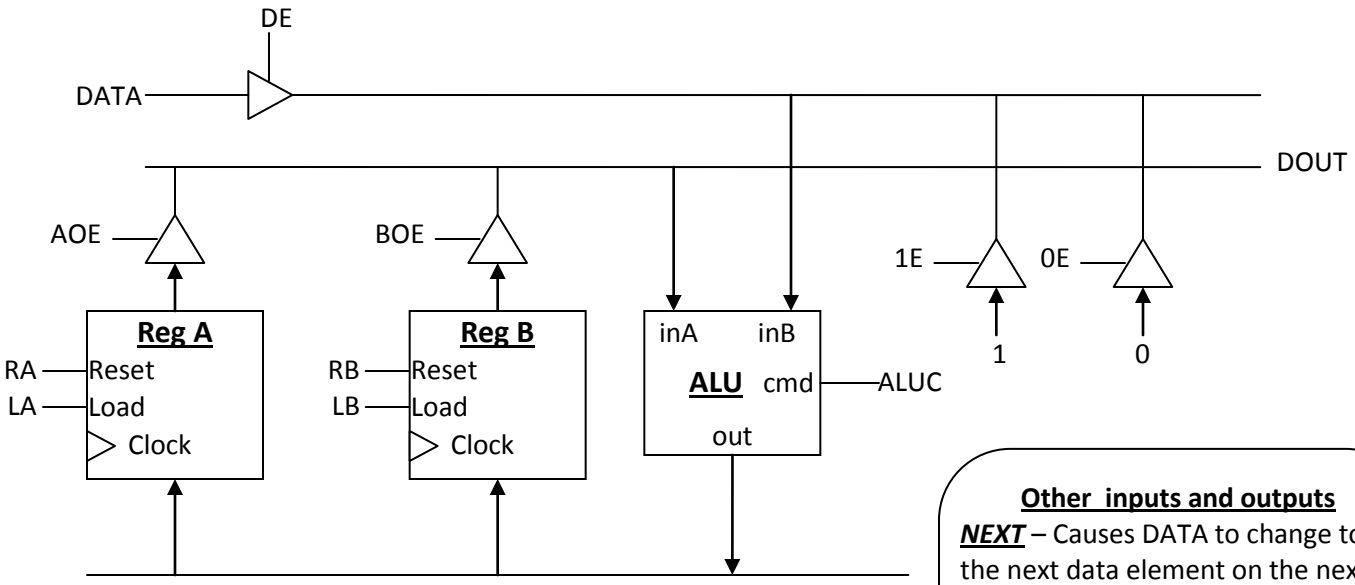
Note X>>Y means to shift X Y places to the right

A) Consider the diagram found below. For each entry in the table, indicate where in the diagram it belongs by placing exactly one of U,V,W,X,Y or Z next to the table entry. **[4 points]**



DATA:	AOE:	B4:
START:	ALUC:	DOUT:

(Problem continues on next page).



B) Draw a Moore-type state diagram which causes the datapath to perform the specified function. Any value not specified in a given state will be assumed to be zero. You may assume there will be no problems with overflow. **[13 points]**

Other inputs and outputs

NEXT – Causes DATA to change to the next data element on the next rising edge of the clock.

DONE – Indicates the result is available on DOUT.

START – You are to start finding the average.

B4 – True if RegB==4

B8 – True if RegB==8

ALU

cnt	Operation
0	out=inA+inB
1	out=inA-inB
2	out=inA<<inB
3	out=inA>>inB

Note X>>Y means to shift X Y places to the right