EECS 270 *Final Exam*

Spring 2012

Name: _____ unique name: _____

Sign the honor code:

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

_____

Scores:

Page #	Points
2	/20
3	/12
4	/10
5	/15
6	/8
7	/5
8	/8
9	/7
10	/15
Total	**/100**

NOTES:

1. Open book (our text only) and Open notes
2. Calculators are allowed, but no PDAs, Portables, Cell phones, etc.
3. Don't spend too much time on any one problem.
4. You have about 120 minutes for the exam.
5. *Be sure to fully label any MSI device you use.*
   Be sure to show work and explain what you've done when asked to do so

1. Fill in each blank or circle the best answer. **[15 points, -2 per wrong or blank answer, min 0]**

   a. The 4-bit 2's complement representation for -5 is _____.

   b. Write 13.3_4 as a decimal number. _____

   c. A*B + A*C in canonical sum-of-products form is _____.

   d. ***B*C'*D' / A'*C'*D / A*D / A'*C*D'*** is an implicant of A'*B'*C' + B*D.

   e. ***DRAM / SRAM / Neither DRAM nor SRAM / Both DRAM and SRAM*** is/are non-volatile.

   f. ***DRAM / SRAM / Neither DRAM nor SRAM / Both DRAM and SRAM*** generally uses a capacitor to store the value of a cell.

   g. A clock period of 50ns corresponds to a frequency of _____MHz.

   h. A 4-input XOR gate has _____ maxterms.

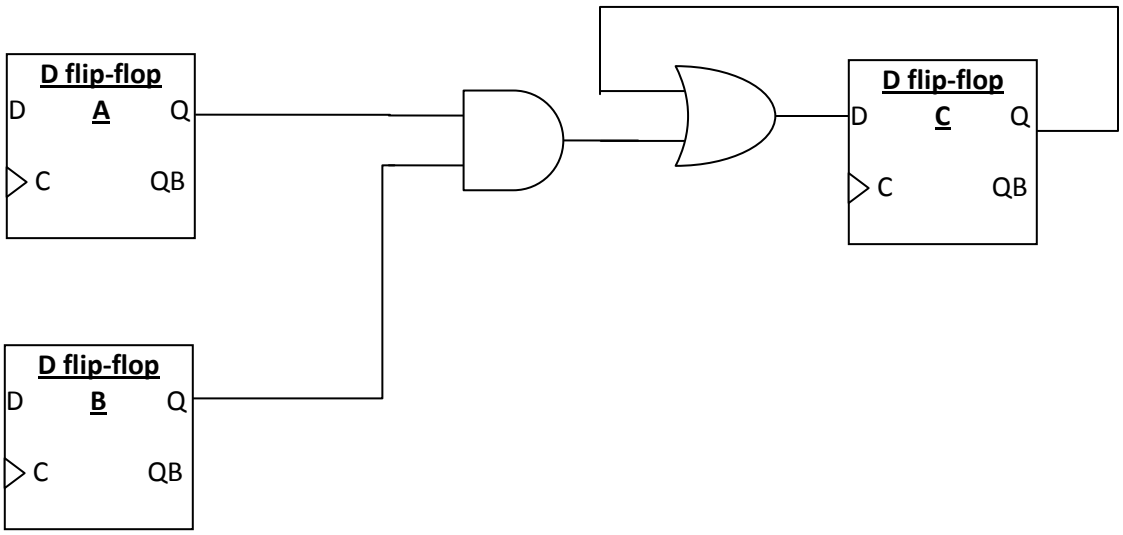
   i. In CMOS you'd need at least _____ transistors to implement a 3-input NOR gate.

   g. Consider a memory device that has 1024 addresses each 16 bits in size. If this was made out of a square memory (equal number of rows and columns in the memory device) the row

      decoder would have _____ inputs while the column MUX would have _____ selection bits.

2. Design a NAND gate using tri-state buffers and inverters. You may freely use power and ground. **[5 points]**

3. Sequential circuit timing **[12 points]**



Consider flip-flops A, B and C each nominally clocked off of the same clock. Assume:

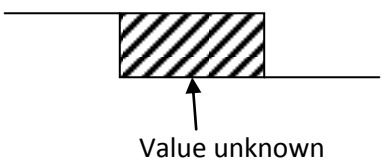
- Each flip-flop has a set-up time of 5ns and a clock-to-Q delay of 2ns to 4ns.
- The AND and OR gates each have a delay of 2 to 4 ns.
- Flip-flop A's rising edge may be as much as 1.25 ns before C's rising edge or as much 1.5 ns after.
- Flip-flop B's rising edge may be as much as 1.3 ns before C's rising edge or as much as 1.2 ns after.

a) What is the fastest clock period you could safely clock this system at? Show your work. **[4]**

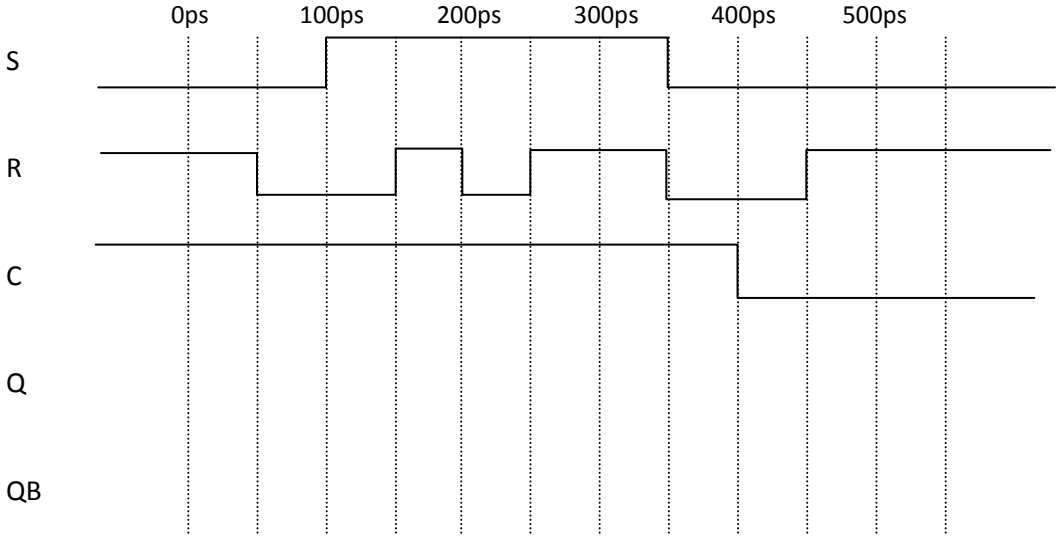
b) What is the (non-negative) range of values for the flip-flop's hold time would be sufficient? Show your work. **[4]**

c) Redo part b) assuming the AND gate always had a delay of exactly 1ns. **[4]**

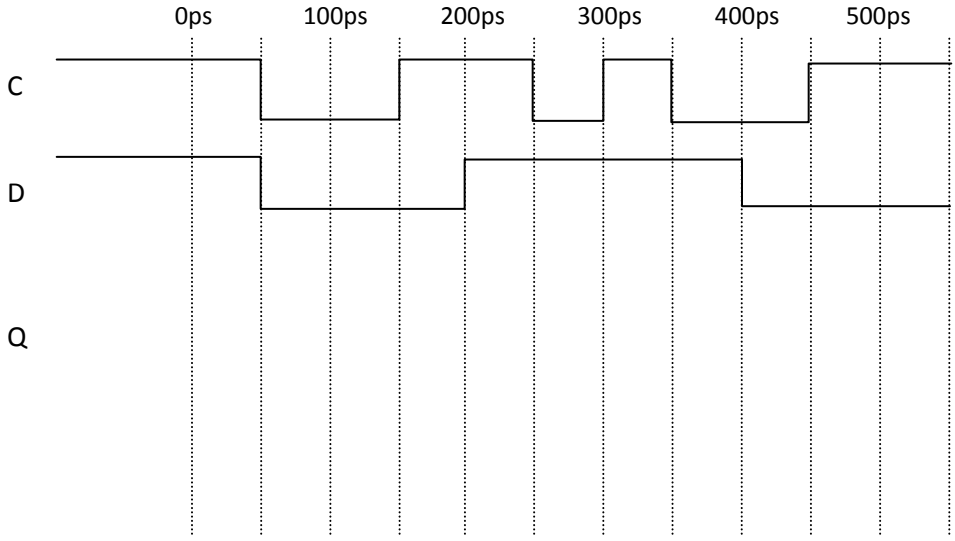
4. Complete the following timing diagrams. If the value is unknown (or oscillating) at some point, clearly indicate that with hashes (as shown). **[10 points]**



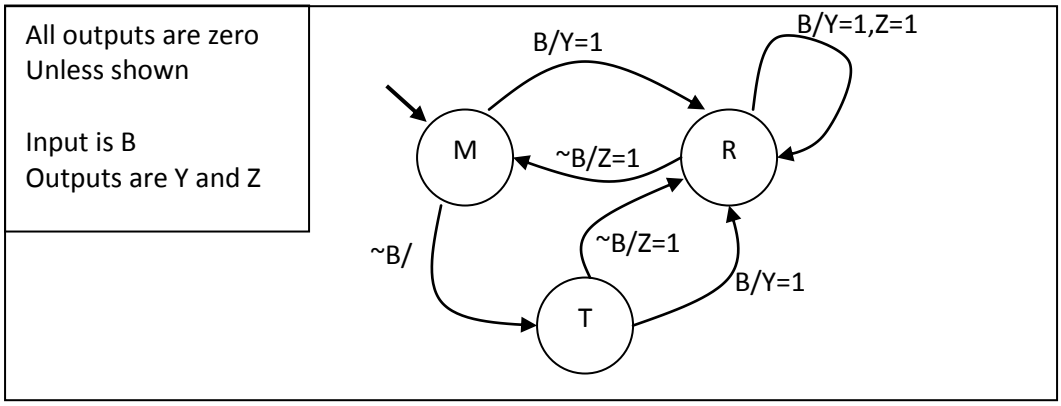
a) Complete the timing diagram below for an SR latch with enable. Start your drawing at 0ps and end it at 550ps. You are to assume the input values before 0ps have been steady for a long time. **[6]**



b) Complete the timing diagram below for a D flip-flop. Assume the flip-flop requires a 25ps hold time and a 75ps setup time and has a 50ns clock to Q delay. Start your drawing at 0ps and end it at 550ps. You are to assume Q is initially 1 at time 0. **[4]**



5. Converting a state transition diagram to logic.



You are to assign state bits S[1:0] as 00 for state M, 10 for state R, and 11 for state T. You are to find logic equations for the next state (NS[1:0]) *in* *minimal* ***sum-of-products form***. For outputs Z and Y you are to find the *minimal* ***product-of-sums form***. Neatly use K-maps to do the minimization. Clearly show your work and provide your final answer where shown. You don't care what happens if the machine somehow gets into state 01. **[15 points]**

NS1=_____ (minimal PoS)

NS0=_____ (minimal PoS)

Y =_____ (minimal SoP)

Z =_____ (minimal SoP)

6. On Logical Minimization and the Jargon Thereof. **[8 points]**

   Consider the following logic function represented in a Karnaugh map:

ab/cd	00	01	11	10
00	0	1	0	0
01	0	0	1	0
11	1	1	1	0
10	0	d	1	1

   a) List all of the Prime Implicants (provide as a comma separated list such as AB, AC', D) for this function:

   _____

   b) List all of the distinguished ones (provide the binary value of each distinguished one) for this function.

   _____

   c) Provide a minimal sum-of-products for this function:

   _____

7. Following the rules for CMOS and using only transistors, design a device that computes A+B'*C'. You may freely use the values A, B, C as well as A', B' and C' as inputs. Your design must use ***eight*** or fewer transistors to receive credit. **[5 points]**

8. Design a state transition diagram for a Moore-type state machine which has one input "A" and one output "Z". Z should go high if the most recent inputs were either "1010" or "110". For full credit, you should use as few states as possible. **[8 points]**

9. Write a Verilog module called **Shift_Right6.** It is to implement a six-bit shift-right register.
    - Inputs are **clock**, **data_in** and **shift_enable**
    - Output is the shift-register's value, **Q[5:0]**.

   The shift register should shift to the right on the rising edge of the **clock** if **shift_enable** is 1, placing the value of **data_in** into the most significant bit. Otherwise the register should hold its value. You will be graded for correctness, syntax and efficiency of you design and code. **[7 points]**

10. You've been tasked with implementing a Mealy-type state machine which has one input "A" and one output "Z". Z should go high if the most recent inputs (including the current input) were either "1010" or "110". Sadly, it turns out you only have a small handful of parts on hand. Implement a circuit which takes the input "A" and generates the output "Z" using on those parts. You (of course) have an input called "clock". ***Part of your grade will be based on the clarity of your solution***, so be neat and be sure to clearly label your parts and their ports! **[15 points]**

    You have ***one*** of each of the following. You may freely use power and ground.
    - 4-input OR gate
    - 4-input AND gate
    - 2 to 1 MUX
    - 4 to 16 decoder
    - 6-bit shift-right register (same behavior and port names as in the previous problem)
    - D flip-flop