The University of Michigan
Department of Electrical Engineering and Computer Science

EECS 270 Fall 2003

Practice Exam #2

Name:_____ UM ID:_____

For all questions, show all work that lead to your answer.

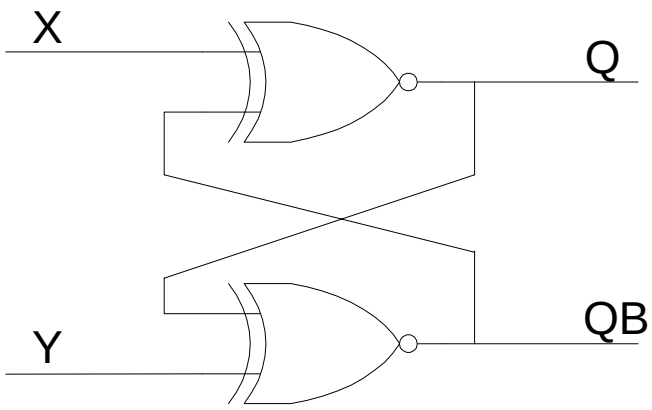
Problem #	Possible Points	Points Earned
1	15	
2	15	
3	25	
4	25	
5	20	
Total	100	

I have neither given nor received any unauthorized aid on this exam.

Signed: _____

Latches and Flip-Flops

1. **(15 pts)** Your manager was quite intrigued by your friend's (incorrect) SR-latch design, and decides to give it a try. He comes up with the following design:

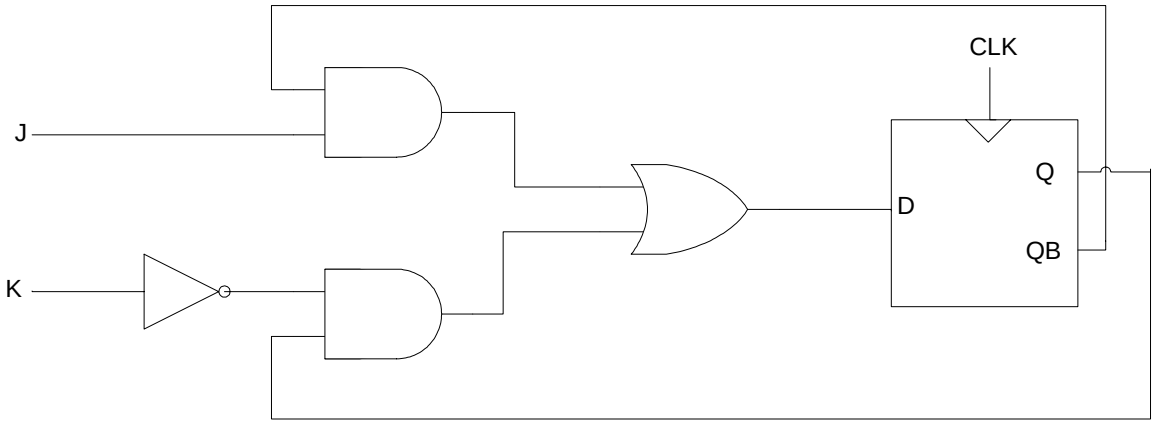


a) Fill in the truth table below:

X	Y	Q	QB
0	0		
0	1		
1	0		
1	1		

b) Is this circuit useful as a latch?

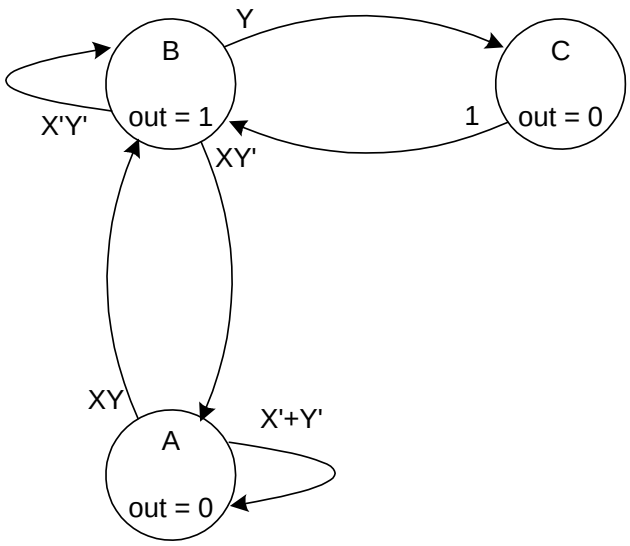
2. **(15 pts)** Derive the characteristic equation for the flip-flop shown below.



Q* =

State Machines

3. **(25 pts)** Given the state machine described by the following state diagram:



a) Assuming that D FFs are used for implementation, and given the following state assignments, create the transition list and derive the minimal expressions for D1, D0 and out.

S	**Q1**	**Q0**
A	1	1
B	0	1
C	0	0

b) Assume that we cluster the unused state e = 10 with state A (i.e., state A = 1X). Derive the new minimal expression for D1.

State Machine Design

4. **(25 pts)**Construct a state diagram for a one input, nine state machine which produces one (Moore-type) output z = 1 whenever the last string of five inputs contains exactly 3 1s **and** the string starts with 2 1s. After each string which starts with 2 1s, analysis of the next string will not start until the end of this 5 character string, whether it produces a 1 or not. For example, the input 11011010 produces an output of 00000000, whereas the input 10011010 produces 00000001

Sequential Block Design

5. **(20 pts)** Construct a counter which produces the following sequence of numbers:

   1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5…15…15, 1, 2, 2, etc.
   (one 1) (two 2's) (three 3's) (four 4's) (five 5's)…(fifteen 15s) (one 1), etc.

   You can use counters that have load, enable, and reset inputs, and counters that can count up as well as down. Use only two counters and no more than 10 basic logic gates (nand, nor, and, or, inverter). *(It is possible for your design to use significantly less than 10 gates.)*