King Fahd University of Petroleum and Minerals
Department of Computer Engineering

DIGITAL LOGIC DESIGN COE 202

Homework 4, January 13, 2009

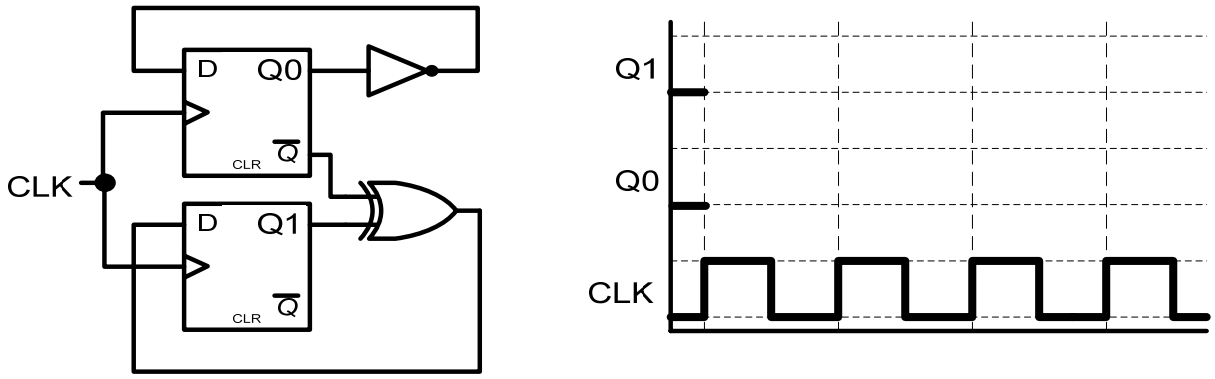
To be returned on January 27, 2009

Problems	Grading
1	
2	
3	
4	
TOTAL	

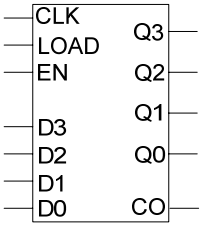
Student Name:……

Student ID:……

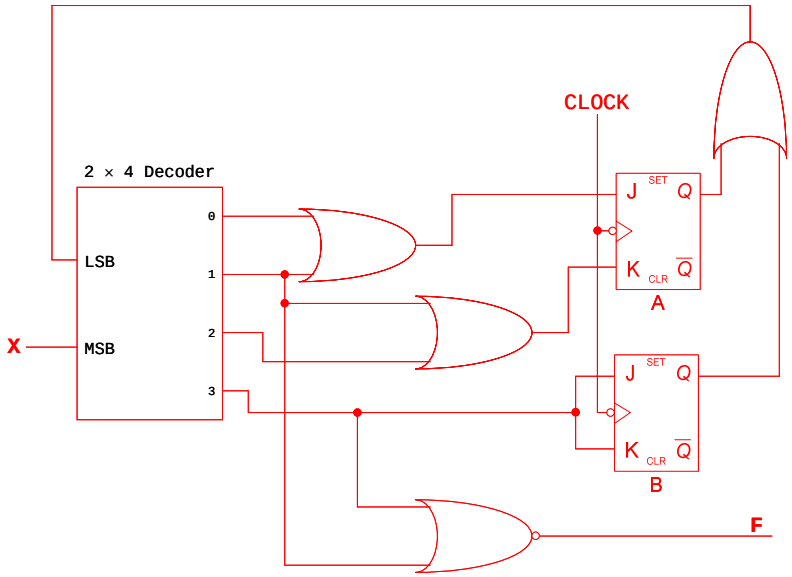
Problem # 1: Two positive edge-triggered D flip-flops are connected as shown in the figure. Both flip-flops are initialized to 0 at the beginning. Fill in the timing diagram for the two outputs Q1 and Q0.



Problem # 2: Use as many as necessary of the following counter with minimal external gates to design a counter that counts from 0 to 20.



Problem # 3: Drive the state diagram for the following circuit (show all the steps of your work):



Problem # 4: Design a *Mealy* sequential circuit that receives data serially on input *X* and produces an output *Y*. The output *Y* is equal to 1 when **all** of the following three conditions are satisfied:

1. The input sequence "10" is detected at least once.
2. At least one "1" is received since the sequence "10" was detected.
3. The total number of "1"s received so far is *odd*.

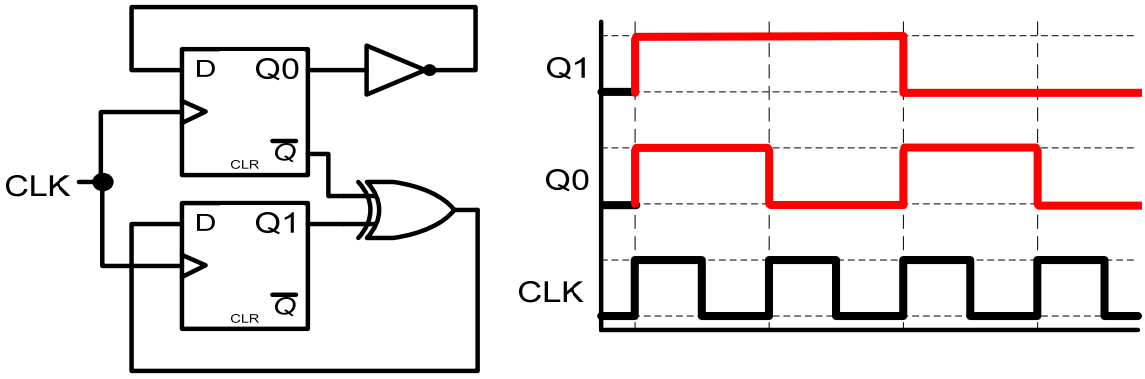
Use rising-edge triggered *JK* flip-flop(s) and a non-inverted decoder to design the circuit. **Show all steps of the design including state minimization**. The following are sample traces to help you in the design of the state diagram as well as verifying your final design:

1. Trace 1:
   X = 00**10**1**1**1110011…
   Y = 000001011101…

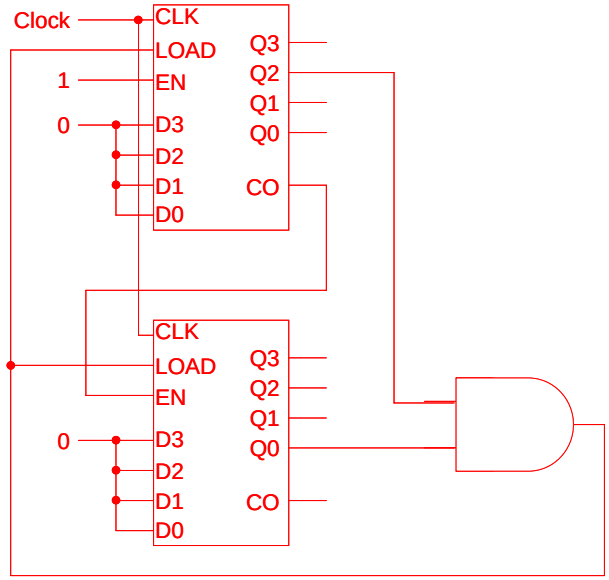
2. Trace 2:
   X = 001**10**00**1**1110011…
   Y = 000000010100010…

Solution:

Problem # 1: Two positive edge-triggered D flip-flops are connected as shown in the figure. Both flip-flops are initialized to 0 at the beginning. Fill in the timing diagram for the two outputs Q1 and Q0.

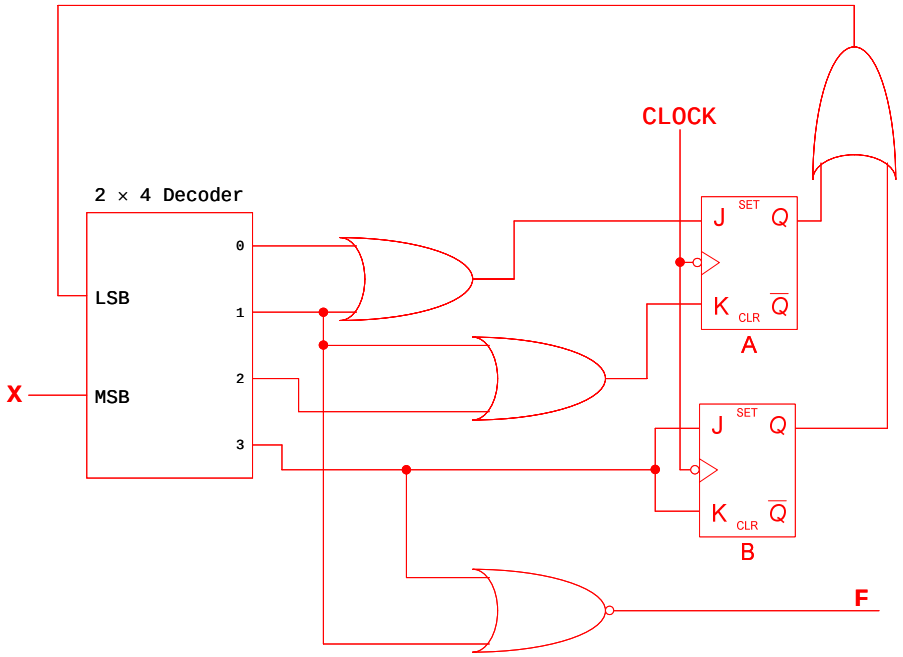


Problem # 2: Use as many as necessary of the following counter with minimal external gates to design a counter that counts from 0 to 20.



Note that the counter designed uses 5 bits $b_4b_3b_2b_1b_0$ to be able to count from 0 to 20. The counter at the top of the diagram will represent b_3, b_2, b_1, and b_0, using Q3, Q2, Q1, and Q0, respectively. On the other hand, the counter at the bottom of the diagram will represent b_4 using Q0.

Problem # 3: Drive the state diagram for the following circuit (show all the steps of your work):



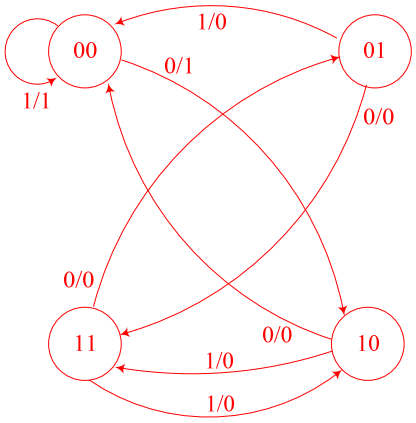
$$J_A = \sum(0,1) = \overline{X} \cdot \overline{(Q_A + Q_B)} + \overline{X} \cdot (Q_A + Q_B) = \overline{X}$$

$$K_A = \sum(1,2) = \overline{X} \cdot (Q_A + Q_B) + X \cdot \overline{(Q_A + Q_B)} = \overline{X} \cdot Q_A + \overline{X} \cdot Q_B + X \cdot \overline{Q_A} \cdot \overline{Q_B}$$

$$J_B = K_B = m_3 = X \cdot (Q_A + Q_B) = X \cdot Q_A + X \cdot Q_B$$

$$F = \overline{\left(\sum(1,3)\right)} = \overline{\left(\overline{X} \cdot (Q_A + Q_B) + X \cdot (Q_A + Q_B)\right)} = \overline{(Q_A + Q_B)} = \overline{Q_A} \cdot \overline{Q_B}$$

Q_A	Q_B	X	J_A	K_A	J_B	K_B	Q_A^+	Q_B^+	F
0	0	0	1	0	0	0	1	0	1
0	0	1	0	1	0	0	0	0	1
0	1	0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	0	0	0
1	0	0	1	1	0	0	0	0	0
1	0	1	0	0	1	1	1	1	0
1	1	0	1	1	0	0	0	1	0
1	1	1	0	0	1	1	1	0	0



Problem # 4:

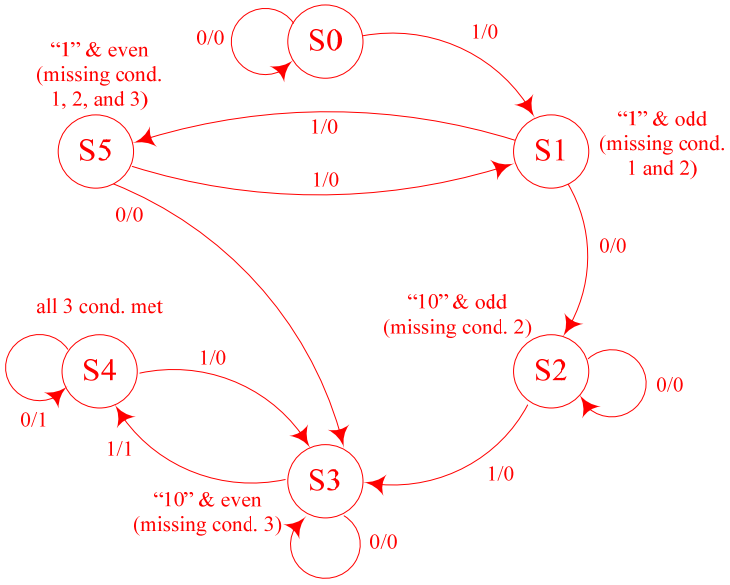
Design a *Mealy* sequential circuit that receives data serially on input X and produces an output Y. The output Y is equal to 1 when **all** of the following three conditions are satisfied:

4. The input sequence "10" is detected at least once.
5. At least one "1" is received since the sequence "10" was detected.
6. The total number of "1"s received so far is *odd*.

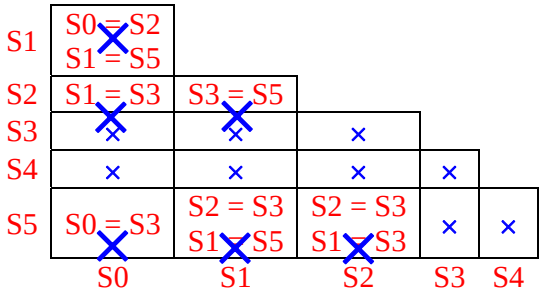
Use rising-edge triggered *JK* flip-flop(s) and a non-inverted decoder to design the circuit. **Show all steps of the design including state minimization**. The following are sample traces to help you in the design of the state diagram as well as verifying your final design:

3. Trace 1: X = 00**10**1**1**1110011…
   Y = 000001011101…

4. Trace 2: X = 001**10**00**1**1110011…
   Y = 000000010100010…



P.S.	N.S.		Output	
	X = 0	X = 1	X = 0	X = 1
S0	S0	S1	0	0
S1	S2	S5	0	0
S2	S2	S3	0	0
S3	S3	S4	0	1
S4	S4	S3	1	0
S5	S3	S1	0	0



State Assignment:
S0 = 000 S3 = 011
S1 = 001 S4 = 100
S2 = 010 S5 = 101
Unused states will be assigned to the 000 state for all combinations of the input with the output *Y* set to 0.

State	Q_A	Q_B	Q_C	X	Q_A^+	Q_B^+	Q_C^+	J_A	K_A	J_B	K_B	J_C	K_C	Y
S0	0	0	0	0	0	0	0	0	×	0	×	0	×	0
	0	0	0	1	0	0	1	0	×	0	×	1	×	0
S1	0	0	1	0	0	1	0	0	×	1	×	×	1	0
	0	0	1	1	1	0	1	1	×	0	×	×	0	0
S2	0	1	0	0	0	1	0	0	×	×	0	0	×	0

	0	1	0	1	0	1	1	0	×	×	0	1	×	0
S3	0	1	1	0	0	1	1	0	×	×	0	×	0	0
	0	1	1	1	1	0	0	1	×	×	1	×	1	1
S4	1	0	0	0	1	0	0	×	0	0	×	0	×	1
	1	0	0	1	0	1	1	×	1	1	×	1	×	0
S5	1	0	1	0	0	1	1	×	1	1	×	×	0	0
	1	0	1	1	0	0	1	×	1	0	×	×	0	0
	1	1	0	0	0	0	0	×	1	×	1	0	×	0
	1	1	0	1	0	0	0	×	1	×	1	0	×	0
	1	1	1	0	0	0	0	×	1	×	1	×	1	0
	1	1	1	1	0	0	0	×	1	×	1	×	1	0

$J_A = \sum(3,7)$ $K_A = \prod(8)$ $J_B = \sum(2,9,10)$ $K_B = \prod(4,5,6)$ $J_C = \sum(1,5,9)$ $K_C = \sum(2,7,14,15)$ $Y = \sum(7,8)$