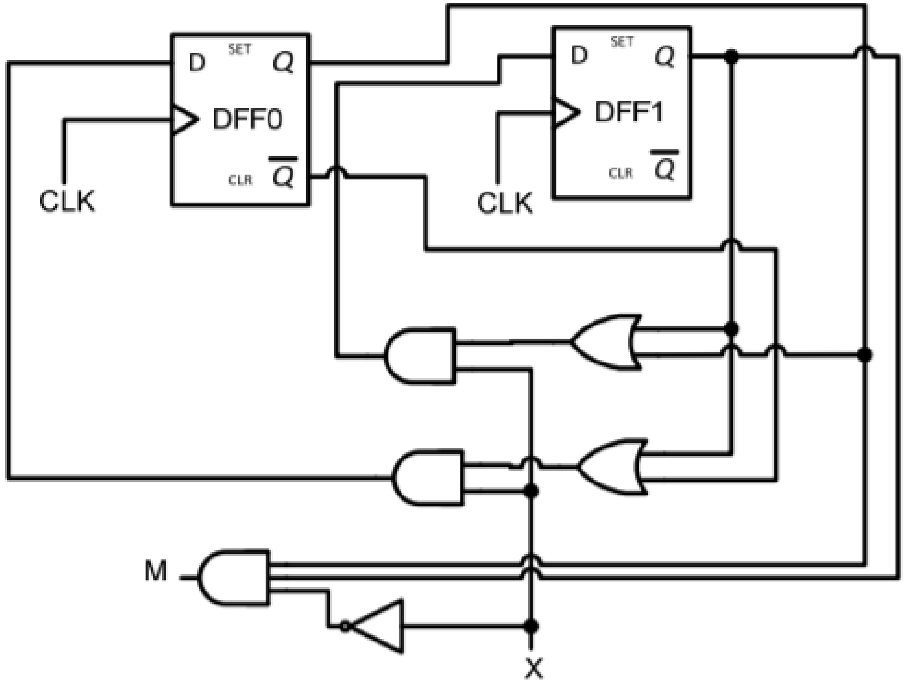
Recitation 5

EE 102, Elnaz Mahmoudi

Problem 1

A sequential circuit design is shown in the following diagram.



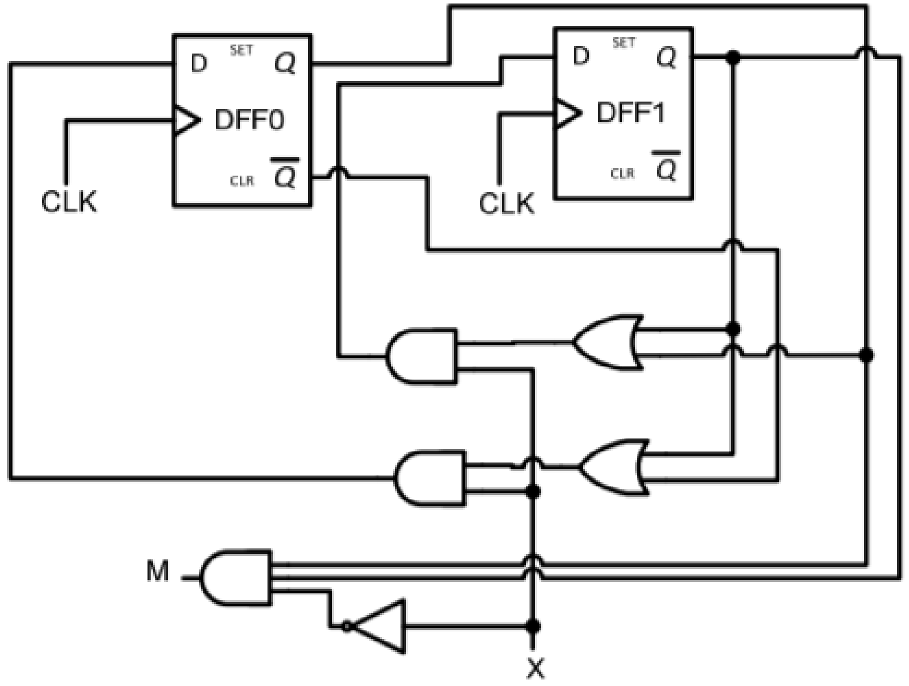
a) Write down Boolean equations for the next state and the output.

a) Write down Boolean equations for the next state and the output.

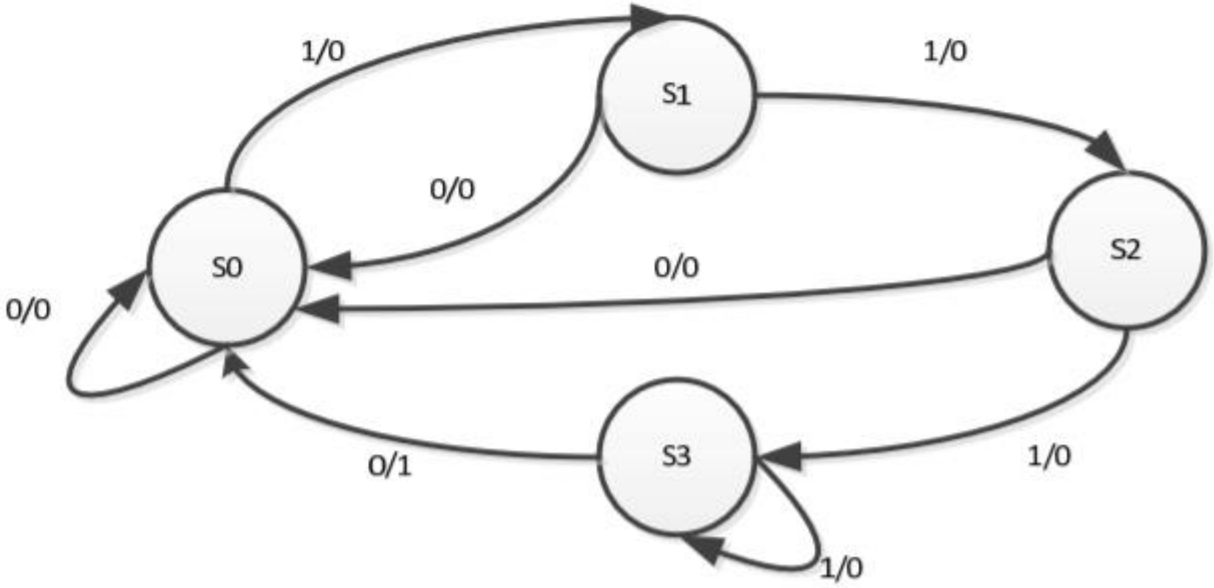
$Q1+= X (Q1+Q0) = XQ1 + XQ0$

$Q0+= X (Q1+Q0') = XQ1 + XQ0'$

$M = X'Q1Q0$



b) Draw the finite state machine representing this design.



c) What does this circuit do? Describe.

This FSM outputs M = 1 when the input has the pattern X = 1110.

Problem 2

a) Design a finite state machine (FSM) for a counter that counts through the 3-bit prime numbers downwards. Assume the counter starts with initial prime value set to 010 as its first 3 bit prime number. You need to provide the state transition table and the state transition diagram. Assume that the state is stored in three D-FFs.

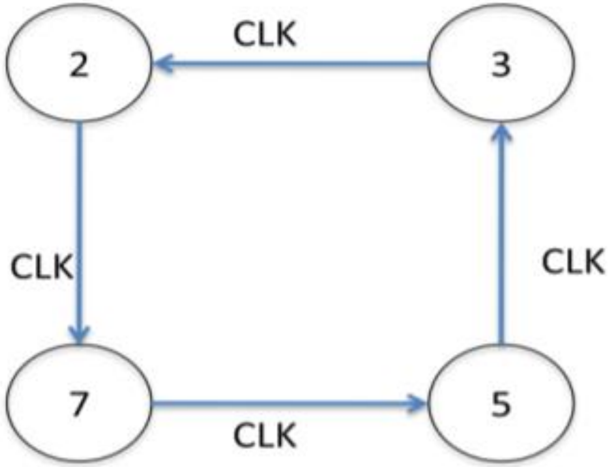
Hint: The set of all 3-bit prime numbers includes 2, 3, 5 and7.

Since there are three bit prime numbers, we will require 3DFF. There are 4 valid states. The rest of the state shouldn't appear in the counter since the counter will always start with a prime number.

State transition table

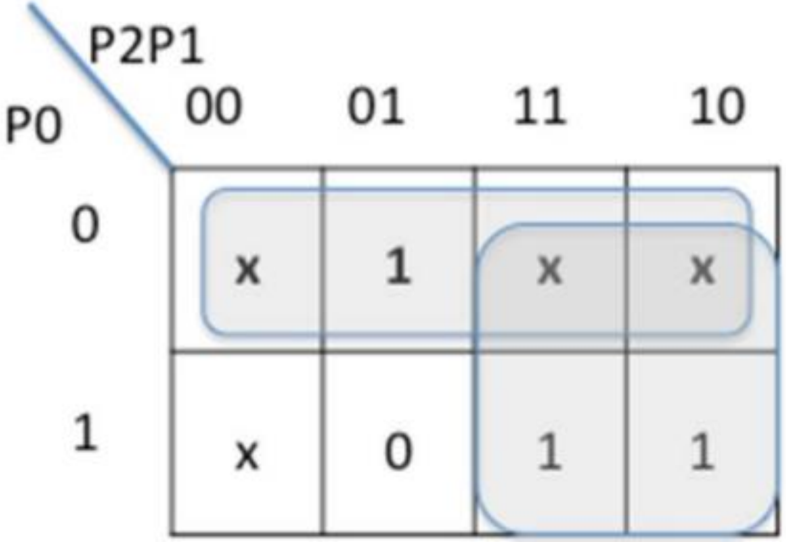
previous state				next state			
value	P2	P1	P0	value	N2	N1	N0
0	0	0	0	x	x	x	x
1	0	0	1	x	x	x	x
2	0	1	0	7	1	1	1
3	0	1	1	2	0	1	0
4	1	0	0	x	x	x	x
5	1	0	1	3	0	1	1
6	1	1	0	x	x	x	x
7	1	1	1	5	1	0	1

State transition diagram, the transition occurs when a CLK signal arrives.

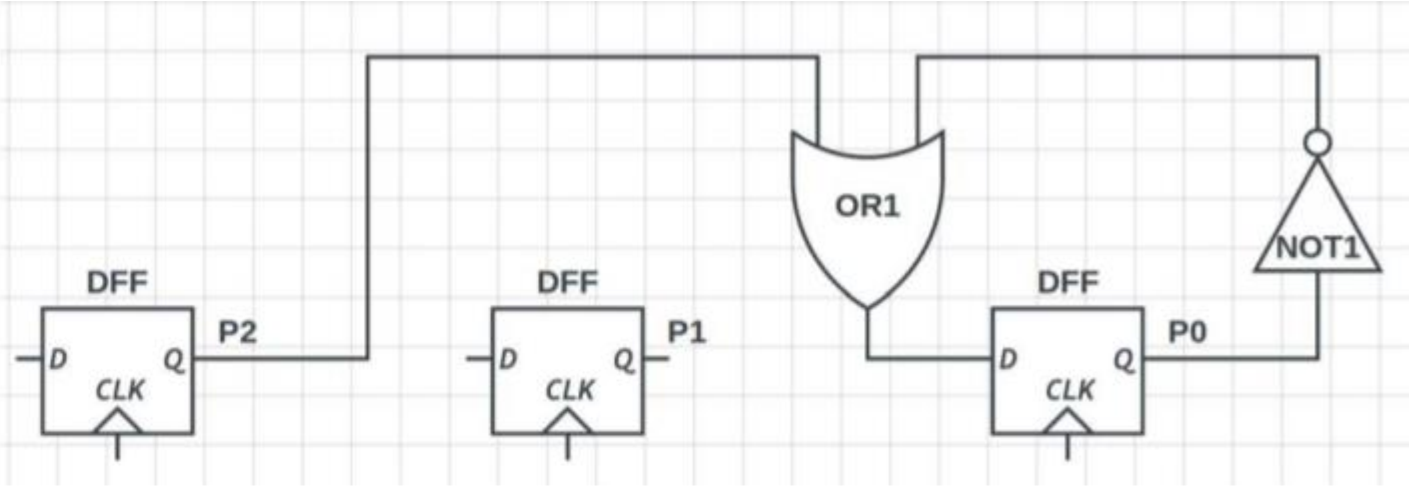


b) Design the circuit for the least significant bit (LSB) of the next state.

For N0, we draw the following Karnaugh map. N0=P2 + P0’



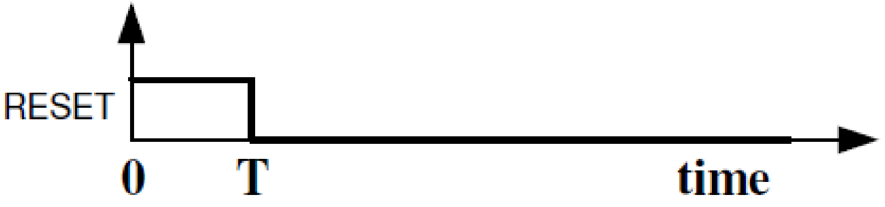
P0 \ P2P1	00	01	11	10
0	x	1	x	x
1	x	0	1	1



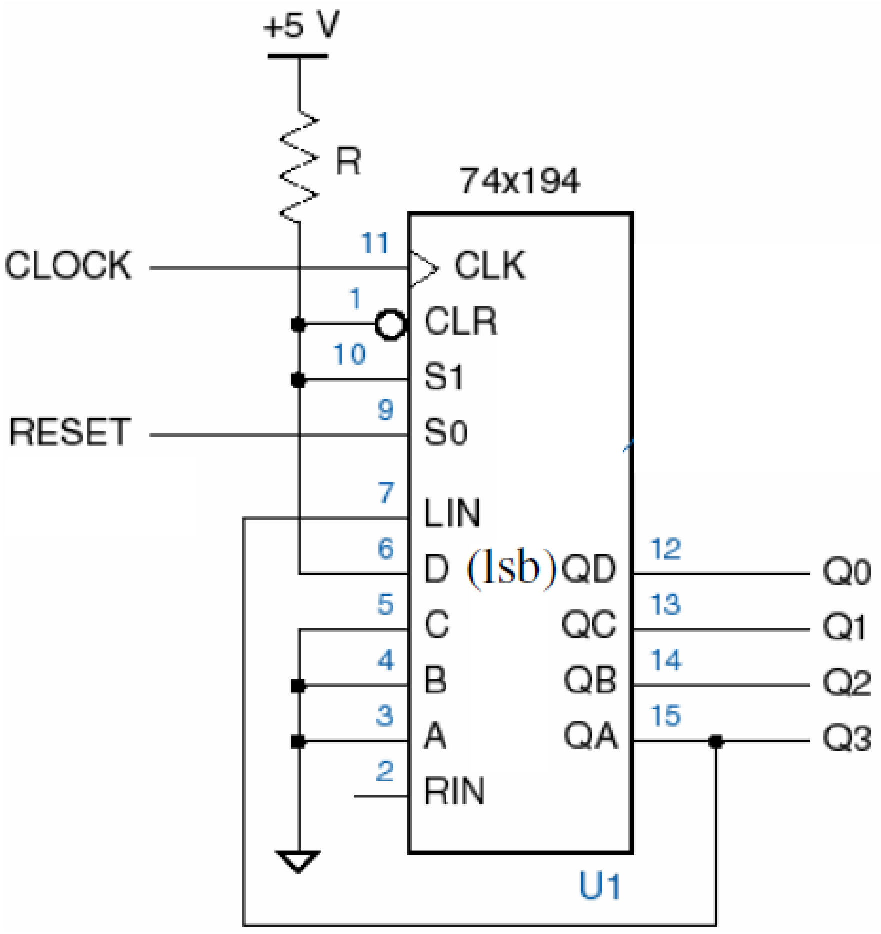
Q1.

What is the counting sequence of the below ring counter assuming that the following RESET signal is applied to S0. Is this counter self correcting? If yes, show that it is self correcting. Otherwise, modify the circuit so that it becomes self correcting.

S1	S0	Function
0	0	LAST Q
0	1	Shift Right
1	0	Shift Left
1	1	LOAD

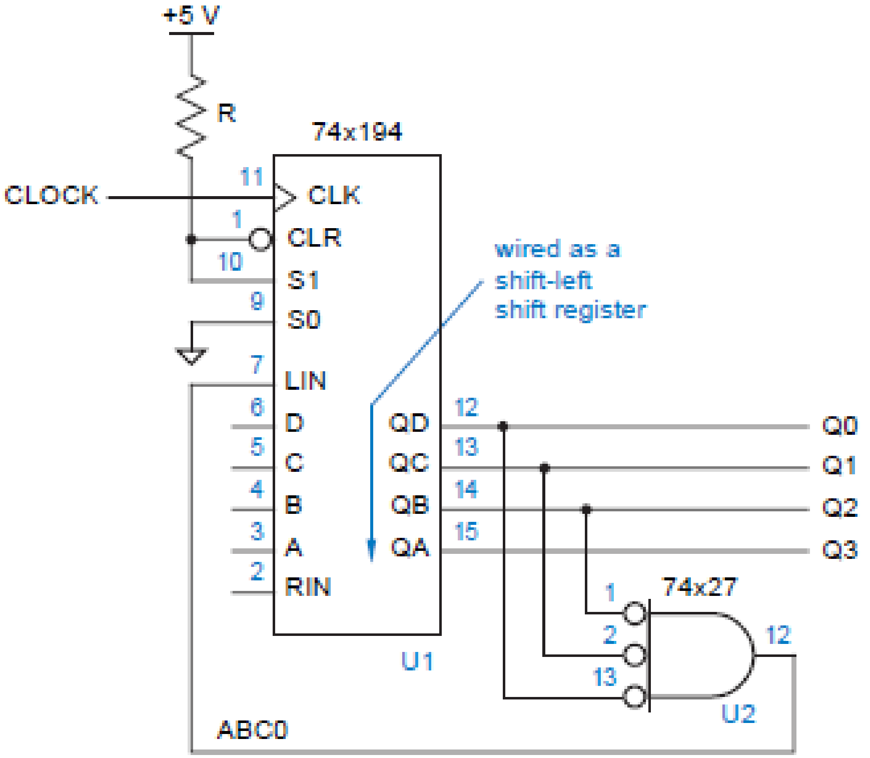


T is sufficiently long to accomodate at least one clock tick



Solution:

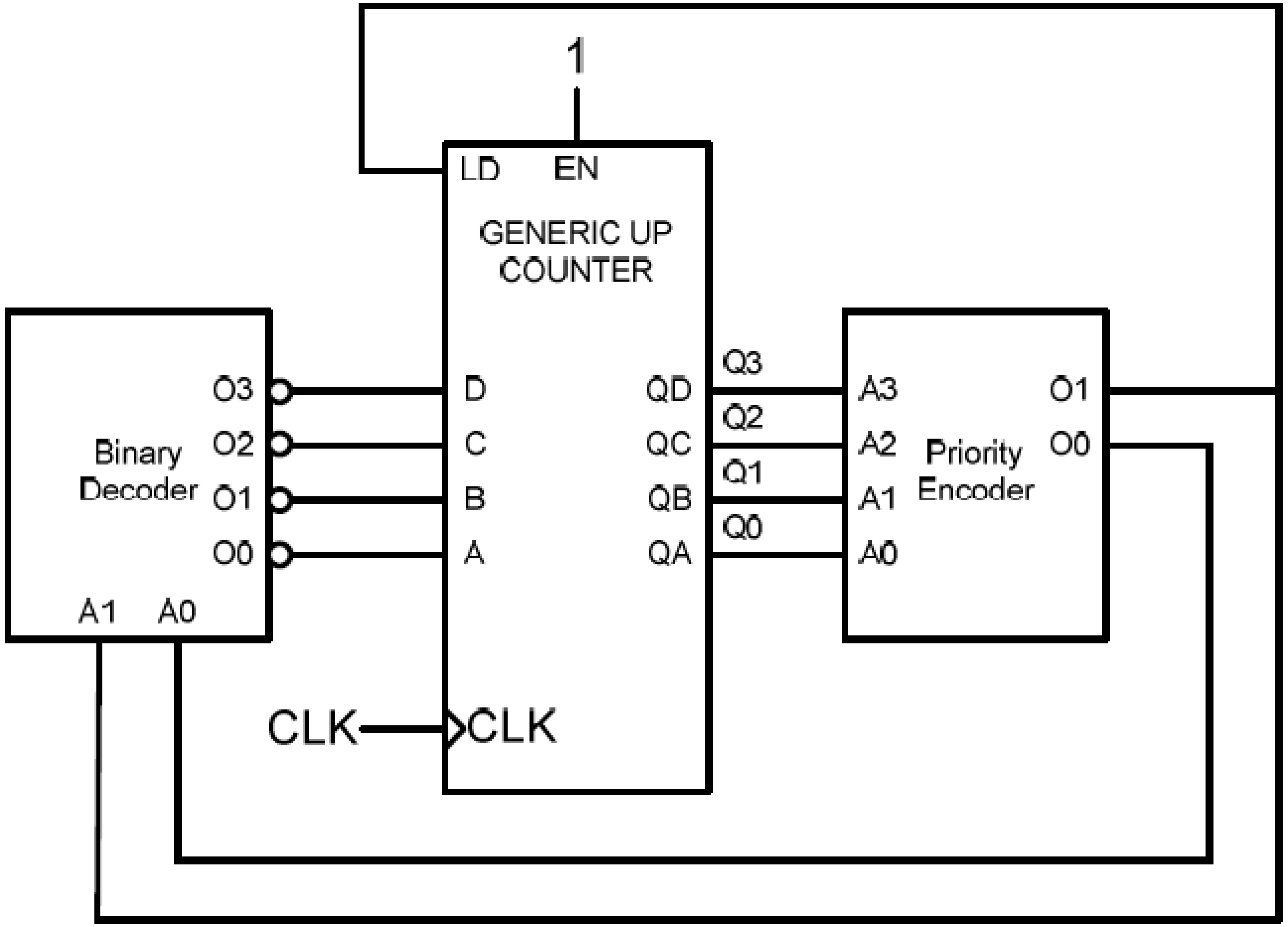
When RESET = 1, the shift register is at the LOAD mode. Thus with the first clock tick and also for the subsequent clock ticks which arrive during when RESET = 1, it loads 0001 to the output. After RESET is unasserted it goes into LEFT SHIFT mode, and counts as 2,4,8,1,... . This circuit is not self correcting and it can be made self correcting as explained in Figure 8-45 in Wakerly 4th edition.

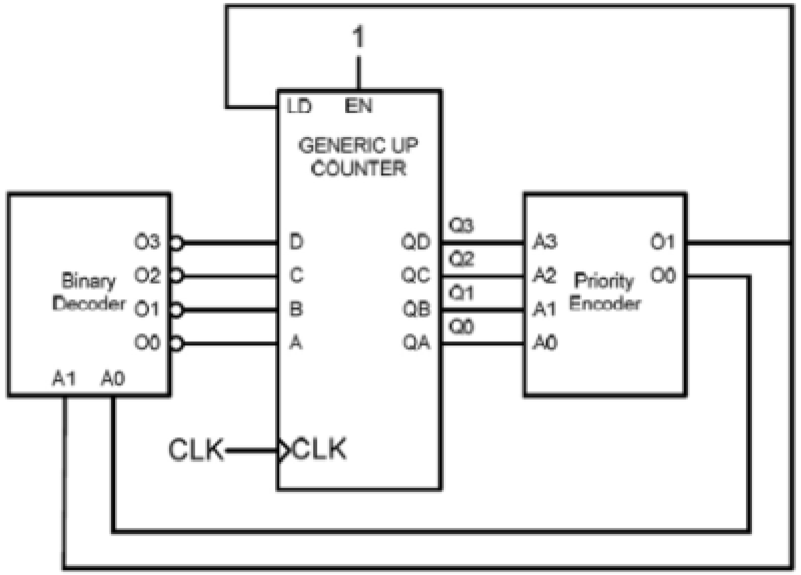


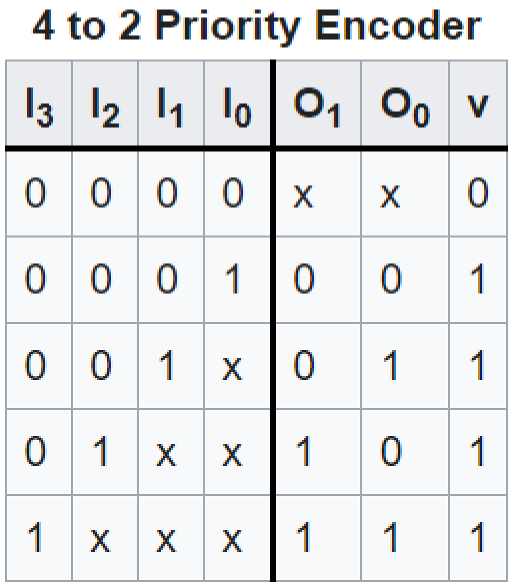
A *self-correcting counter* is designed so that all abnormal states have transitions leading to normal states. Self-correcting counters are desirable for the same reason that we use a minimal-risk approach to state assignment in Section 7.4.3: If something unexpected happens, a counter or state machine should go to a “safe” state.

Q2.

Determine the counting sequence of the following configuration and fill in the following table. Assume that the counter is initially 0. Priority order is A3, A2, A1, and A0 from high to low in the priority encoder.





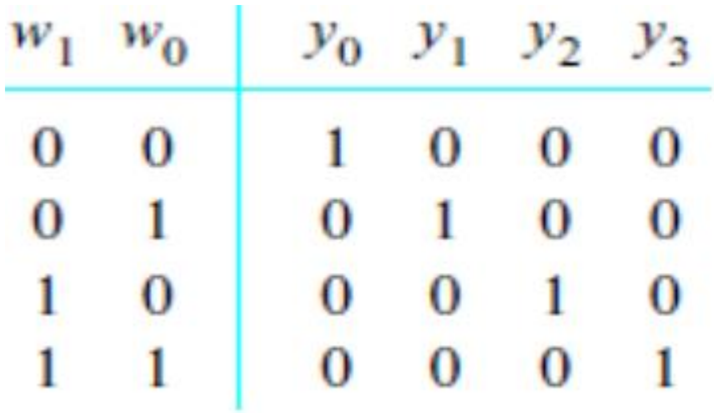


4 to 2 Priority Encoder

I_3	I_2	I_1	I_0	O_1	O_0	v
0	0	0	0	x	x	0
0	0	0	1	0	0	1
0	0	1	x	0	1	1
0	1	x	x	1	0	1
1	x	x	x	1	1	1

Next State Table to determine the counting sequence:

Q3	Q2	Q1	Q0	LD	Q3*	Q2*	Q1*	Q0*
0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	0	1	1
0	0	1	1	0	0	1	0	0
0	1	0	0	1	1	0	1	1
1	0	1	1	1	0	1	1	1
0	1	1	1	1	1	0	1	1
1	0	1	1	1	0	1	1	1

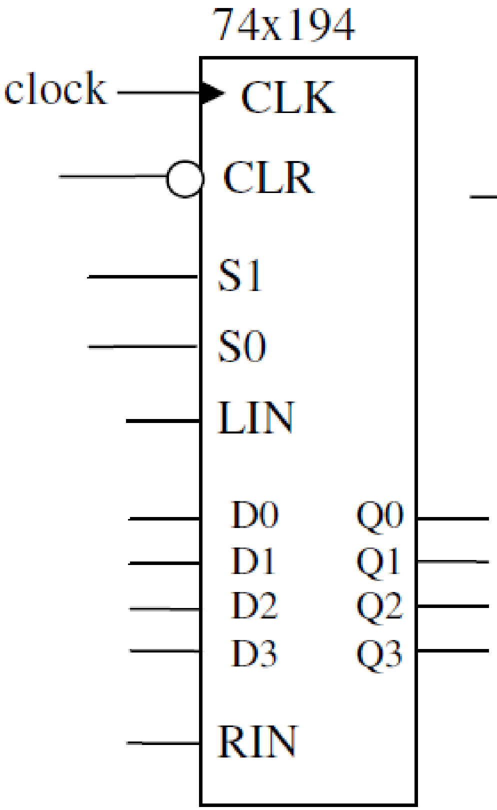


w_1	w_0	y_0	y_1	y_2	y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

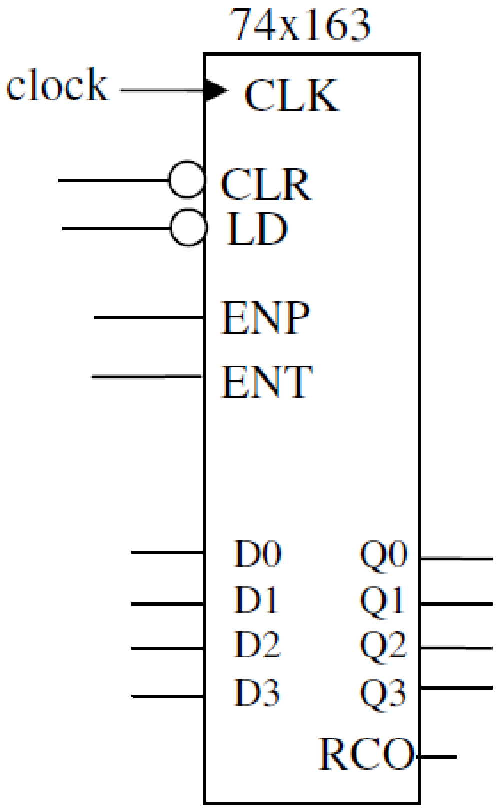
Counting sequence: 7 – 11 – 7 – 11 – 7 – 11…..

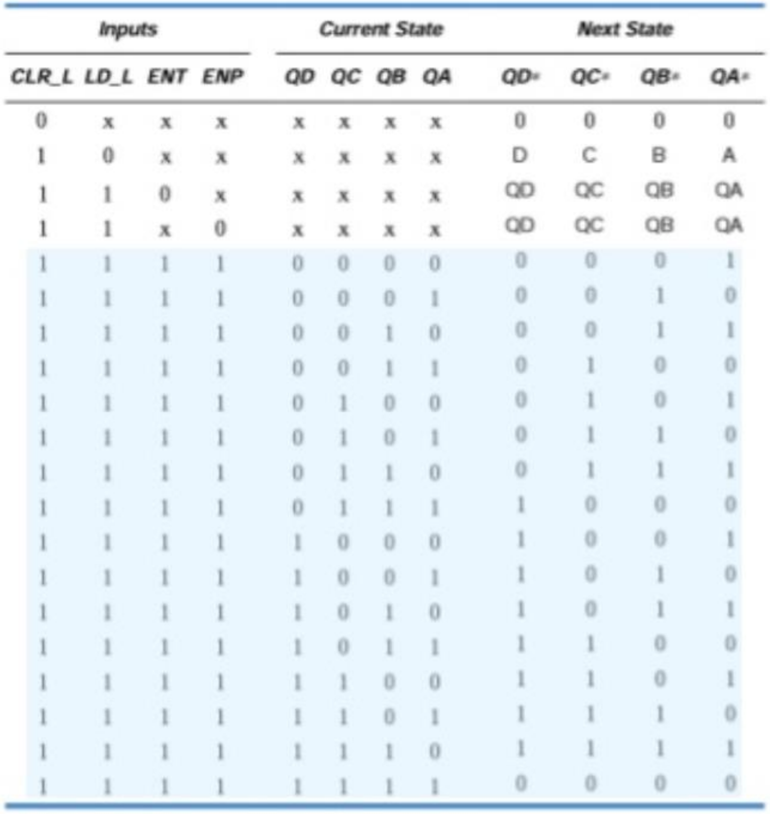
Q3.

a) You are to design a circuit with a clock input, another input X, and one output Y. Right after every 4th clock tick Y becomes ’1’ if the values of X at the last four clock ticks have been “1101”, and ‘0’ otherwise. Y becomes ‘0’ right after other clock ticks. Y does not change in between clock ticks (not considering the changes due to delays). Use the universal shift register 74x194, the 4-bit counter 74x163, and additional components to design this circuit. Assume that 74x194 and 74x163 outputs are all ‘1’ at power ON.



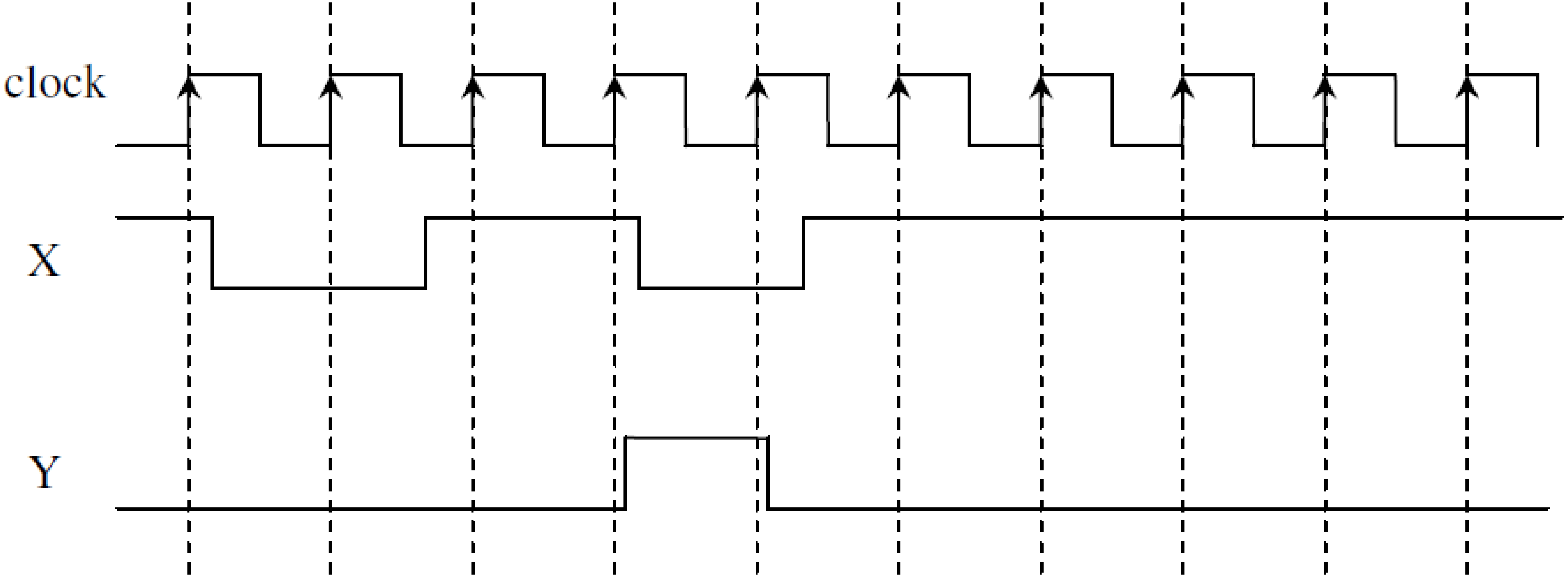
S1	S0	Function
0	0	hold
0	1	shift right
1	0	shift left
1	1	load

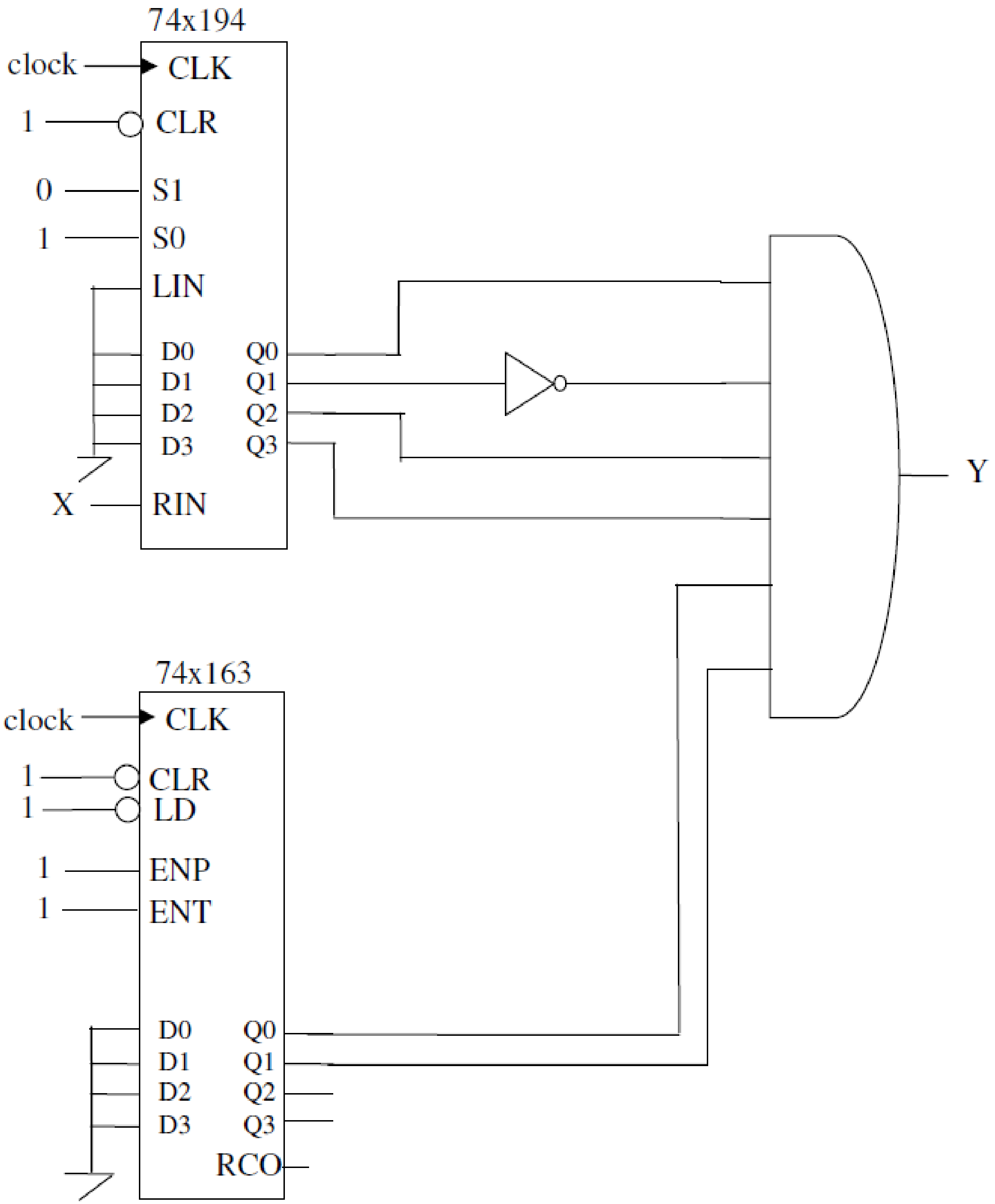




Inputs				Current State				Next State			
CLR_L	LD_L	ENT	ENP	QD	QC	QB	QA	QD*	QC*	QB*	QA*
0	x	x	x	x	x	x	x	0	0	0	0
1	0	x	x	x	x	x	x	D	C	B	A
1	1	0	x	x	x	x	x	QD	QC	QB	QA
1	1	x	0	x	x	x	x	QD	QC	QB	QA
1	1	1	1	0	0	0	0	0	0	0	1
1	1	1	1	0	0	0	1	0	0	1	0
1	1	1	1	0	0	1	0	0	0	1	1
1	1	1	1	0	0	1	1	0	1	0	0
1	1	1	1	0	1	0	0	0	1	0	1
1	1	1	1	0	1	0	1	0	1	1	0
1	1	1	1	0	1	1	0	0	1	1	1
1	1	1	1	0	1	1	1	1	0	0	0
1	1	1	1	1	0	0	0	1	0	0	1
1	1	1	1	1	0	0	1	1	0	1	0
1	1	1	1	1	0	1	0	1	0	1	1
1	1	1	1	1	0	1	1	1	1	0	0
1	1	1	1	1	1	0	0	1	1	0	1
1	1	1	1	1	1	0	1	1	1	1	0
1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0

To illustrate the problem further the following timing diagram is provided:

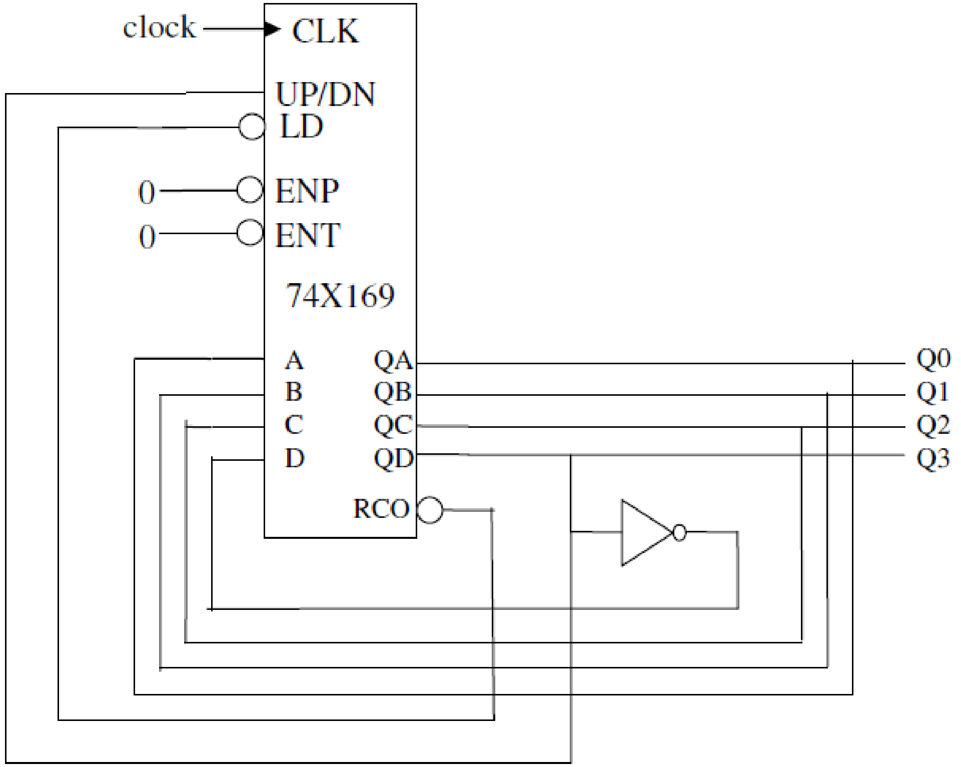



74x194
clock
CLK
1
CLR
0
S1
1
S0
LIN
D0
D1
D2
D3
Q0
Q1
Q2
Q3
X
RIN
Y
74x163
clock
CLK
1
CLR
1
LD
1
ENP
1
ENT
D0
D1
D2
D3
Q0
Q1
Q2
Q3
RCO

Q4.

Problem 8.13 from Wakerly 4th edition. In 74X169, which is an up/down counter, LOAD operation takes priority over other operations. RCO becomes 1 when 0 is reached while in DOWN count mode, and it becomes 1 when 15 is reached while in UP count mode. It counts up if UP/DN is 1 and counts down if it is 0. What is the counting sequence of the following circuit?

To make life easy for you I suggest you assume that at Power-ON the output is decimal 10. Explain your solution. Preferably write a next state table to clearly explain from what state the counter goes to what state. The next state of course depends on LD and UP/DN as well as load inputs (A,B,C,D).



Present State											**Next State**				
Decimal	**QD**	**QC**	**QB**	**QA**	**D=QD'**	**C=QC**	**B=QB**	**A=QA**	**UP/DN=QD**	**LD=RCO**	**QD***	**QC***	**QB***	**QA***	**Decimal**

SOLUTION:

Next State Table to determine the counting sequence starting from 10:

Note that RCO is a function of the present state and present UP/DN value. LD has precedence over UP/DN. Note that this next state table is different from what you are accustomed to. In the first line present state is decimal 10 and the next state is decimal 11. In the second line I have taken the present state to be decimal 11 so that the next state after decimal 11 is determined, and so on. In any case, it covers all present states. The counting sequence is therefore 10,11,12,13,14,15,7,6,5,4,3,2,1,0,8,9,10,…

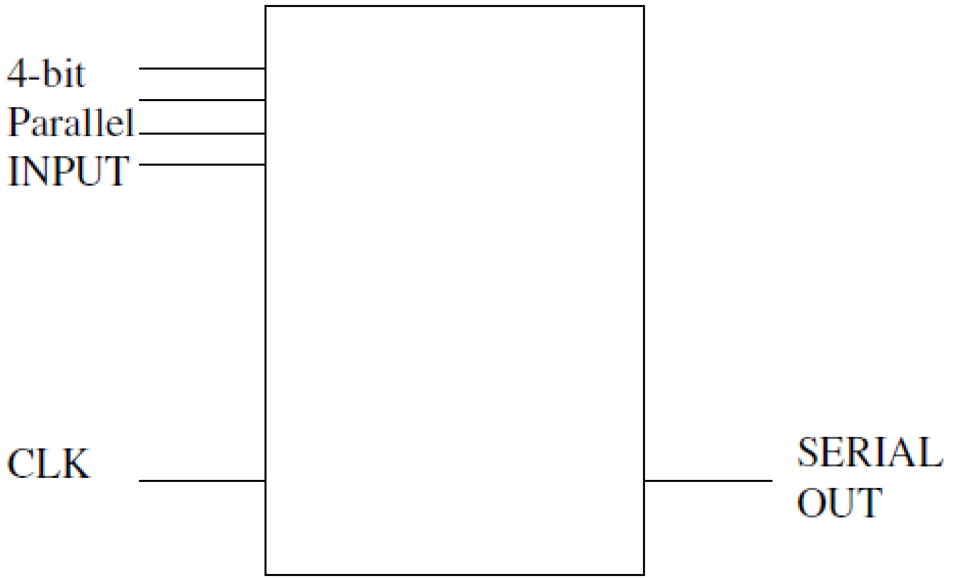
Present State											Next State				
Decimal	QD	QC	QB	QA	D=QD’	C=QC	B=QB	A=QA	UP/DN=QD	LD=RCO	QD*	QC*	QB*	QA*	Decimal
10	1	0	1	0	0	0	1	0	1	0	1	0	1	1	11
11	1	0	1	1	0	0	1	1	1	0	1	1	0	0	12
12	1	1	0	0	0	1	0	0	1	0	1	1	0	1	13
13	1	1	0	1	0	1	0	1	1	0	1	1	1	0	14
14	1	1	1	0	0	1	1	0	1	0	1	1	1	1	15
15	1	1	1	1	0	1	1	1	1	1	0	1	1	1	7
7	0	1	1	1	1	1	1	1	0	0	0	1	1	0	6
6	0	1	1	0	1	1	1	0	0	0	0	1	0	1	5
5	0	1	0	1	1	1	0	1	0	0	0	1	0	0	4
4	0	1	0	0	1	1	0	0	0	0	0	0	1	1	3
3	0	0	1	1	1	0	1	1	0	0	0	0	1	0	2
2	0	0	1	0	1	0	1	0	0	0	0	0	0	1	1
1	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	8
8	1	0	0	0	0	0	0	0	1	0	1	0	0	1	9
9	1	0	0	1	0	0	0	1	1	0	1	0	1	0	10
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Q5.

a) You are to design a parallel-to-serial converter which loads 4-bit parallel data at every 4th clock tick and shifts them in between. Thus for example at the nth clock tick it loads the data "ABCD" and its serial output becomes D, C, B, and A at the nth, n+1st, n + 2nd, and n + 3rd clock ticks respectively. At the n + 4th tick it loads a new 4-bit parallel data, and so on.

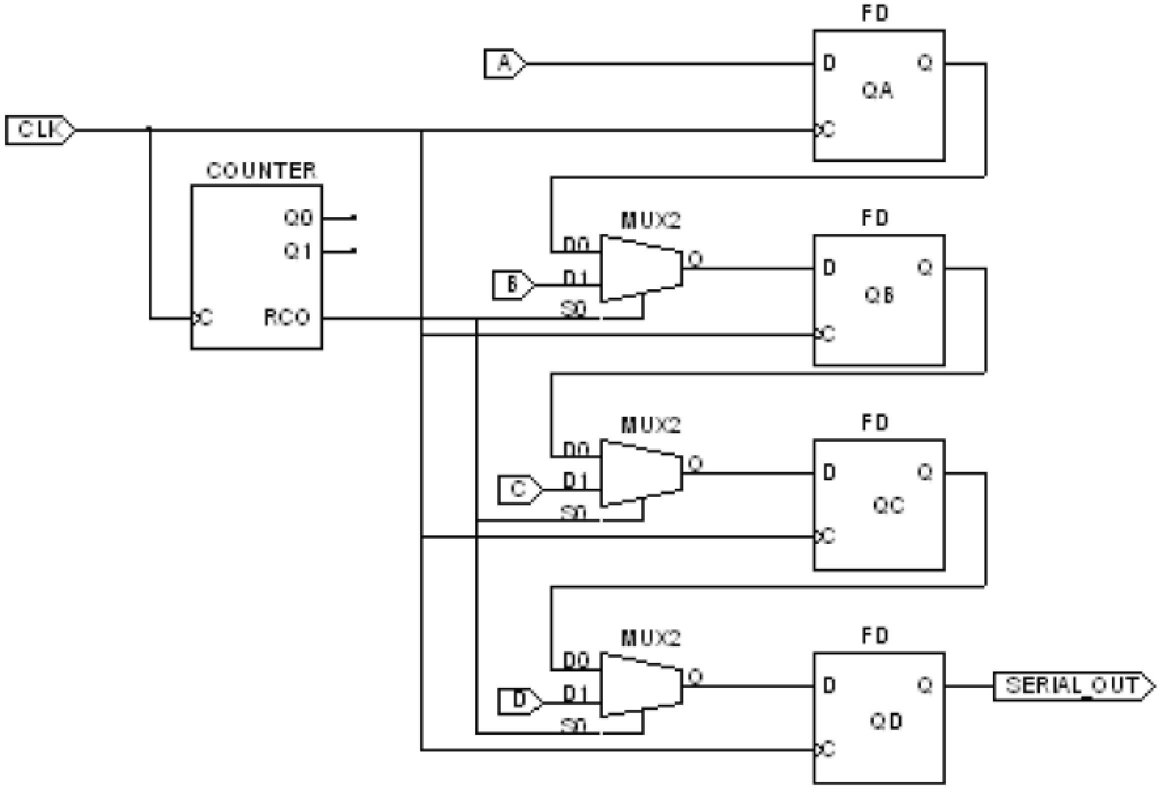
To achieve this, first modify the serial-in serial-out shift register structure in page 727 using three 2-to-1 multiplexers. And then use a 2-bit counter with RCO output to obtain the parallel-to-serial converter. Draw the timing diagram of your final circuit to illustrate the parallel to serial conversion.

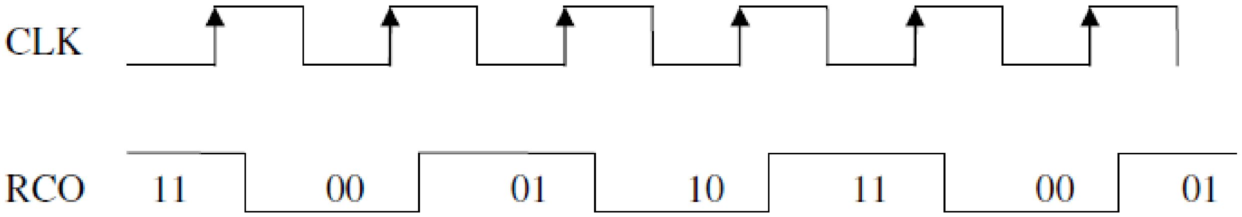
Your final design should have the following pin diagram:



Solution

a)





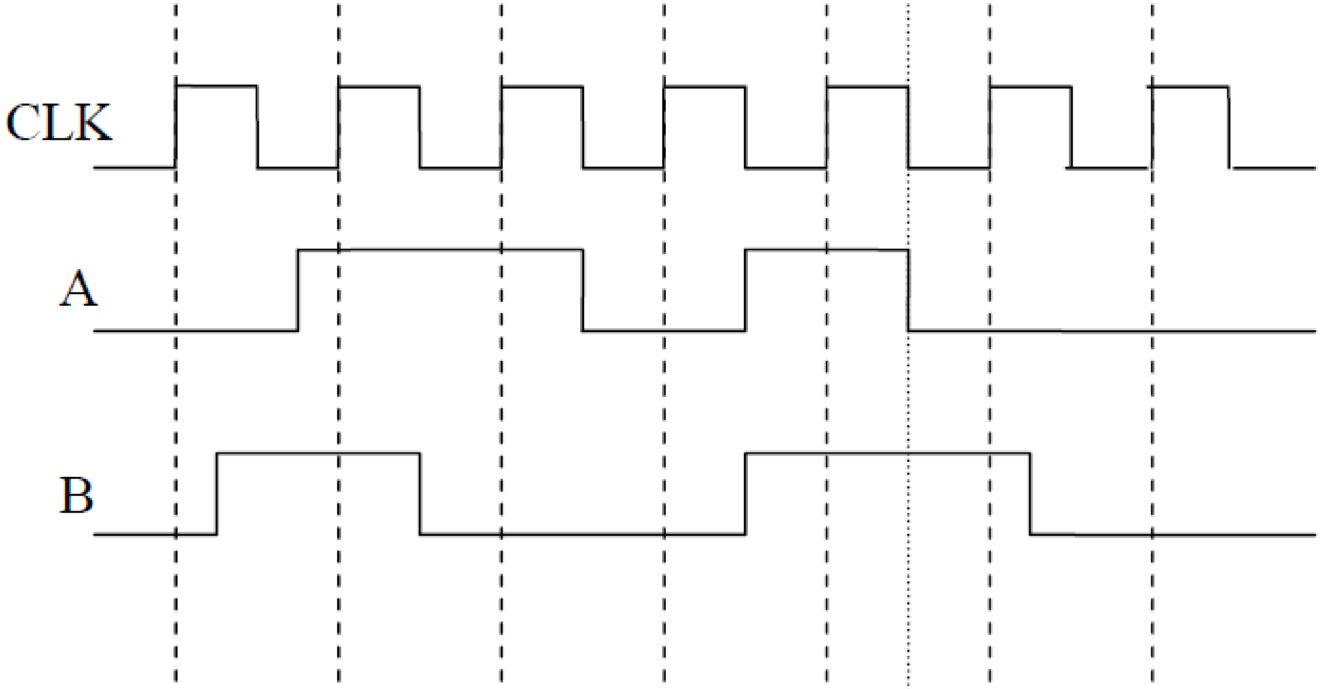
QA	*	A1	*	*	*	A2	*
QB	*	B1	A1	*	*	B2	A2
QC	*	C1	B1	A1	*	C2	B2
QD	A0	D1	C1	B1	A1	D2	C2

Q4. The VHDL code of an FSM is given below.

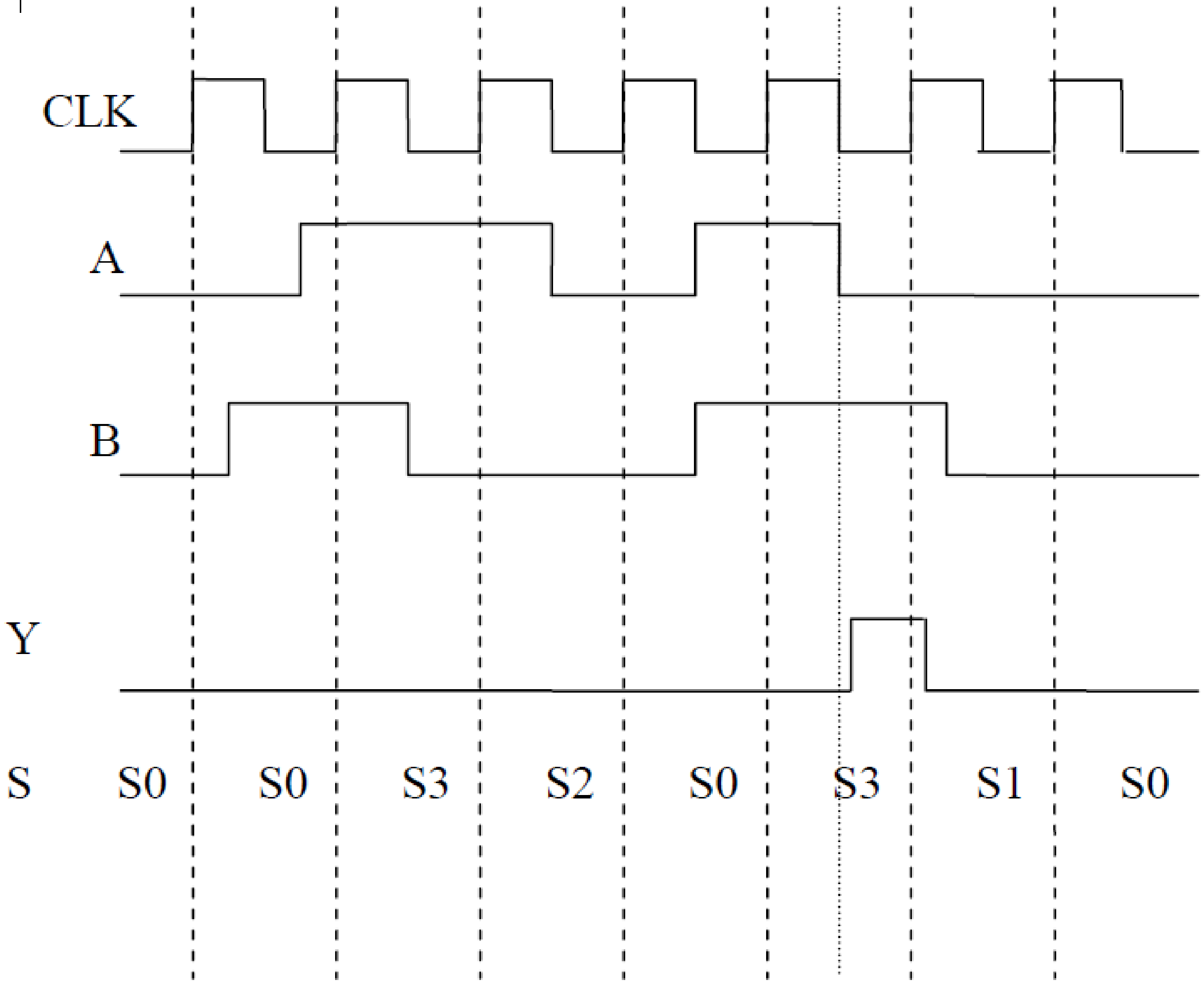
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity TwoInputFSM is
  Port ( CLK : in std_logic;
        A : in std_logic;
        B : in std_logic;
        Y : out std_logic);
end TwoInputFSM;

architecture Behavioral of TwoInputFSM is
type newtype is (S0,S1,S2,S3);
signal S:newtype :=S0;
signal N:newtype;

begin
S<=N when CLK'event and CLK='1' else S;
N<=S1 when (not A and B)='1' else S2 when (A and not B)='1' else S3 when (A and
B)='1' else S0 when (not A and not B)='1';
Y<='1' when S=S3 and A='0' else '0';
end Behavioral;
```

a) Draw the waveform of Y for the given waveforms for A and B.

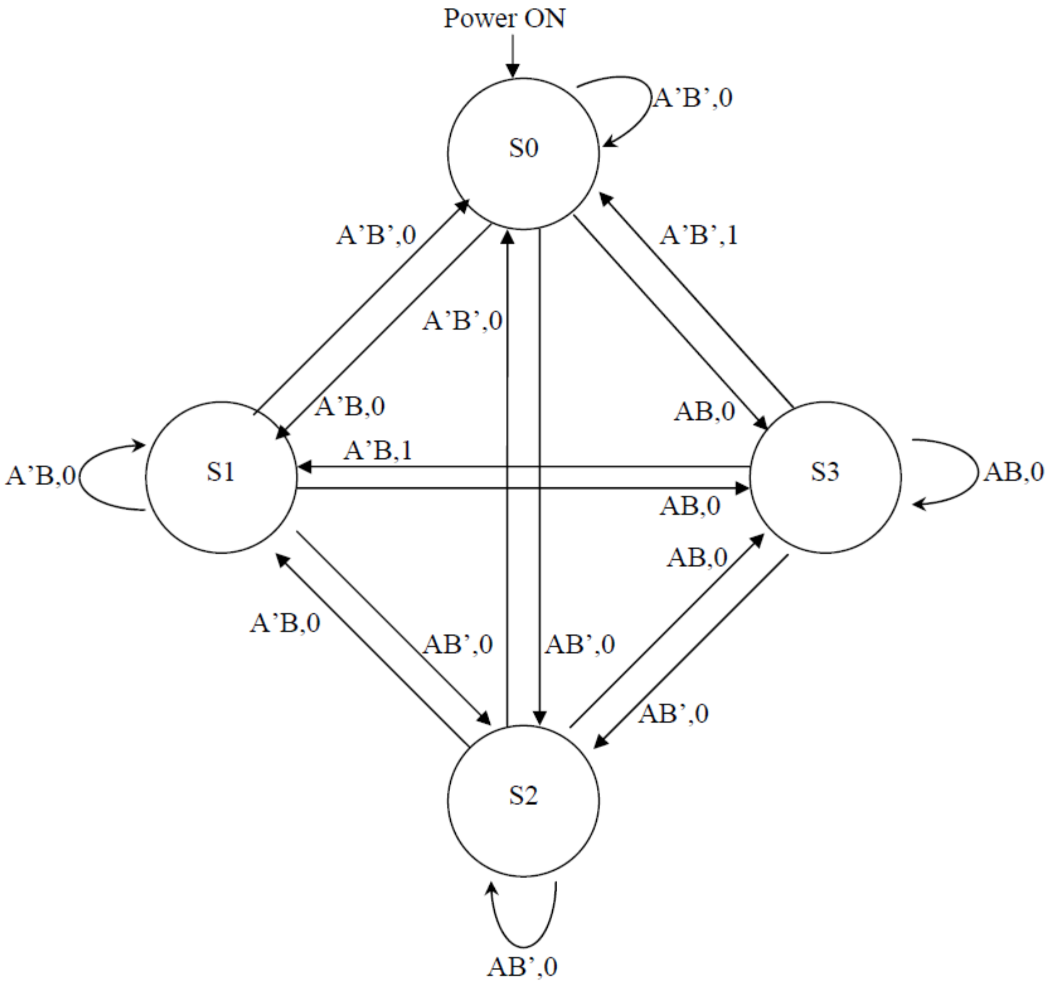
Solution:



a) Draw the waveform of Y for the given waveforms for A and B.
Solution:



b) Draw the state/output diagram of this FSM



Q14.

Design and draw an FSM with two inputs (X, Y), and one output (F) that works as follows:

If at the last clock tick X>Y, then FSM goes to state 1
If at the last clock tick X<Y, then FSM goes to state 2
If at the last clock tick X=Y, then FSM goes to state 3
F = 1 if X=Y at the last clock tick and also at present.
Power-ON state is state 0.

Solution:
Since X and Y are single bit numbers,
X>Y means X = 1 and Y = 0,
X<Y means X = 0 and Y = 1,
X=Y means (X = 1 and Y = 1) or (X = 0 and Y = 0).

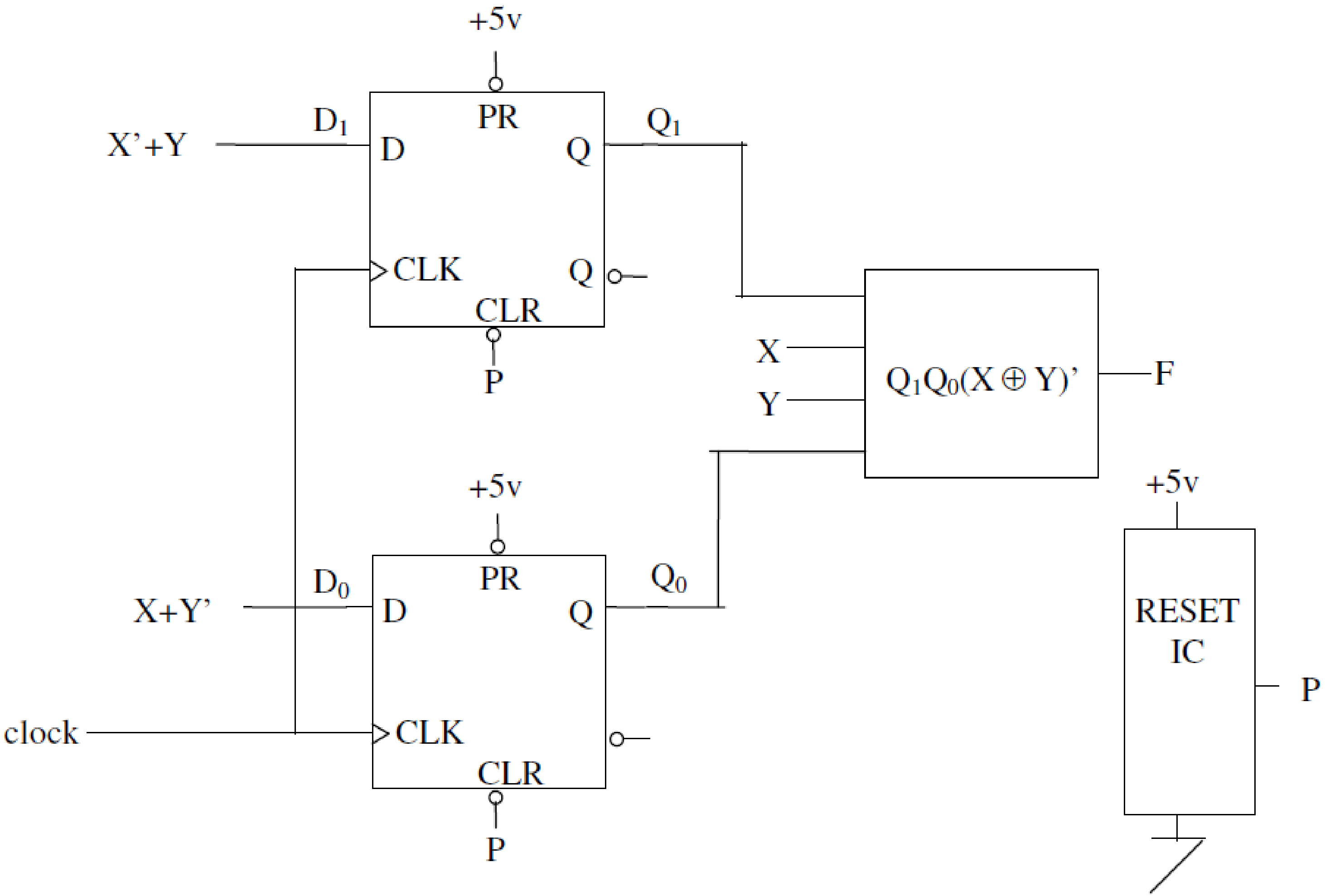
There are three states
S0:Power ON
S1: at the last clock tick X>Y
S2:at the last clock tick X<Y
S3:at the last clock tick X=Y
We need 2 flip flops. Say Q1 and Q0.
State encoding:
S0: Q1Q0 = 00
S1: Q1Q0 = 01
S2: Q1Q0 = 10
S3: Q1Q0 = 11

As we notice the next state does not depend on the present state but only on the last received X and Y values. The next state table is

X	Y	Q1*	Q0*
0	0	1	1
0	1	1	0
1	0	0	1
1	1	1	1

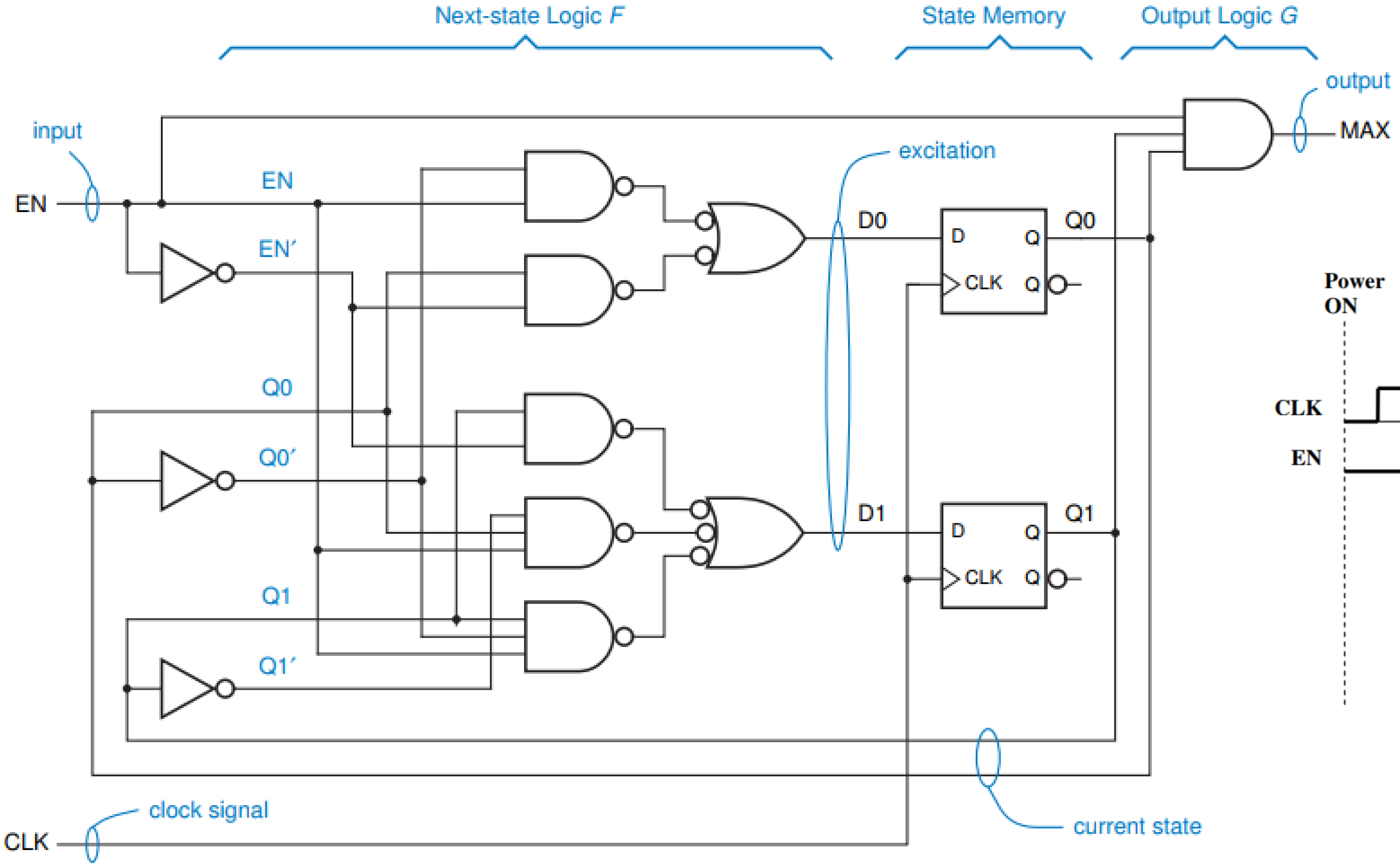
If we use D ffs we have D1 = X'+Y, and D0 = X+Y'.

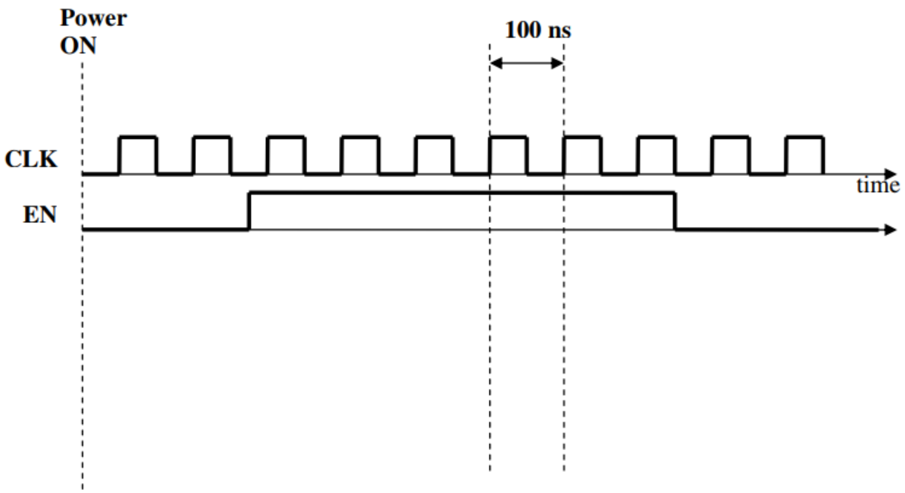
Output is 1 when we are at state S3 and also the present values of the inputs are equal. Therefore F = Q1Q0(X ⊕ Y)'



Q1.

The following is a clocked synchronous state machine of Mealy type. Draw timing waveforms for D0, Q0, D1, Q1, MAX for the waveforms given for EN and CLK below. Assume that at power ON Q0 = Q1 = 0. Assume that the D ffs have delays of 25 ns but the other gates do not have delays. Also comment on the meanings of the input EN, the binary number Q1Q0, and the output MAX.



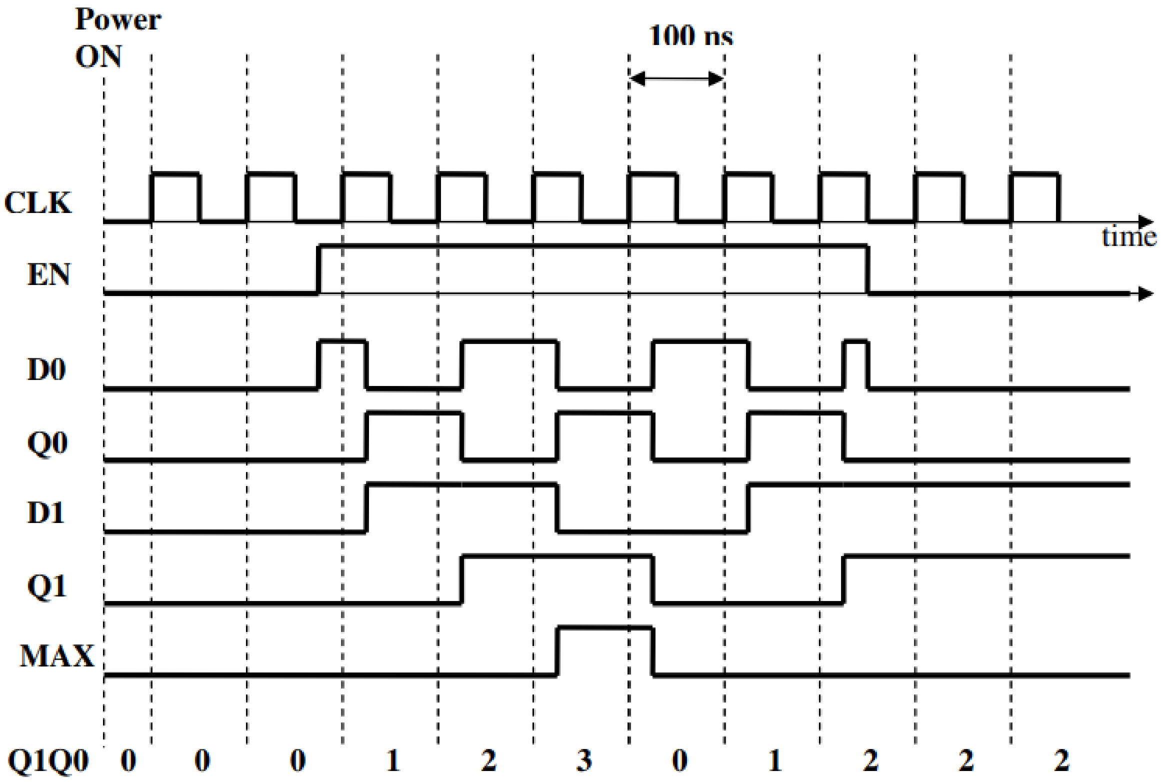


Therefore if we consider the binary number Q_1Q_0 as an output then this circuit is a counter which counts as 0,1,2,3,0,... if EN = 1. However if EN = 0 then the counter stops at its last count. If EN were to become 1 again it would continue counting from where it had stopped. Note that even if EN = 1, ff outputs change only at clock ticks. MAX is the actual output and it is 1 if and only if $Q_1Q_0 = 11$ and EN = 1.

This circuit is of Mealy type because the output MAX depends on the instantaneous value of the input EN.

SOLUTION:

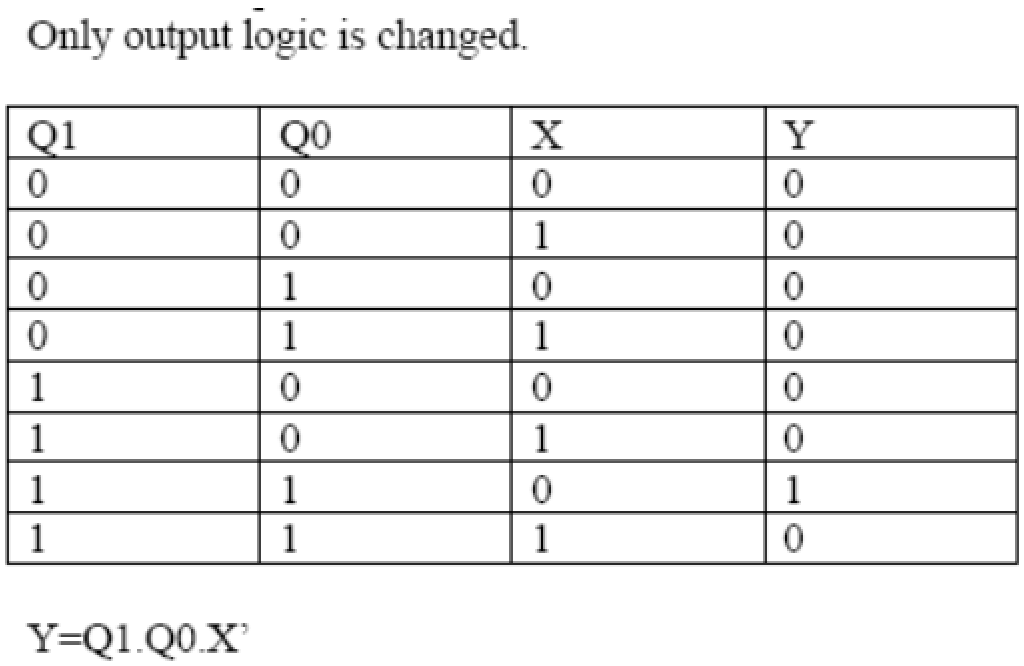
Note that $D_0 = ENQ_0' + EN'Q_0$ and $D_1 = EN'Q_1 + ENQ_1'Q_0 + ENQ_1Q_0'$



Q7.
Design an FSM to perform the following operation:

- **An FSM has 4 states, S0, S1, S2, and S3 one input, X, and one output, Y.**
- **If X = 1 at the last tick, then the FSM goes to state S2.**
- **If X = 0 at the last tick, then the FSM goes to state S1 if before the tick it was in state S0 or S2.**
- **If X = 0 at the last tick, then the FSM goes to state S3 if before the tick it was in state S1 or S3.**
- **Power ON state is S1.**
- **The output, Y, is "1" if and only if the FSM is in state S3.**

Q8.
Modify the above FSM such that the output, Y, is "1" if the FSM is in state S3 and the current input X=0.



Only output logic is changed.

Q1	Q0	X	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Y=Q1.Q0.X’