Q1) Find the output of z from the following verilog code when inputs are x=1 and y =1

```
module check(
input x,
input y,
output z
);
assign z = (~x & ~y) |(x & y);
endmodule
```

Solution

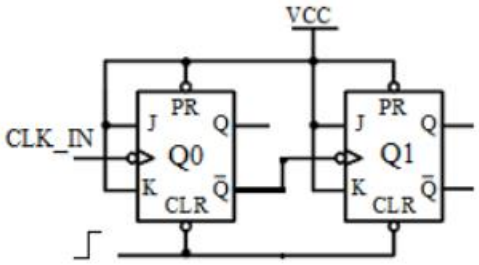
Z = 1

Q2) Fill the missing statement of the following verilog code which implements D Flip Flop

```
module RisingEdge_DFlipFlop_SyncReset(D,clk,sync_reset,Q);
_____ D; //
input clk; //
input sync_reset; // synchronous reset
_____reg Q; // output Q
always @(posedge___)
begin
 if(sync_reset==1'b1)
  Q <= ____
 else
  Q <= _____
end
endmodule
```

Solution:

```
module RisingEdge_DFlipFlop_SyncReset(D,clk,sync_reset,Q);
input D; // Data input
input clk; // clock input
input sync_reset; // synchronous reset
output reg Q; // output Q
always @(posedge clk)
begin
 if(sync_reset==1'b1)
  Q <= 1'b0;
 else
  Q <= D;
end
endmodule
```

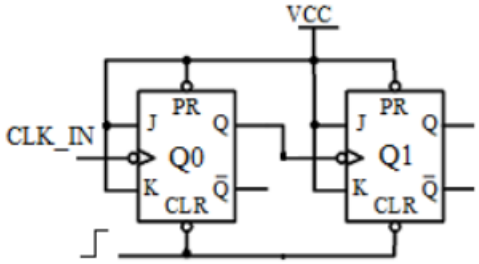


Q3) Shown above is a simple two-bit binary counter circuit. The Q output of the first flip-flop constitutes the least significant bit (LSB), while the second flip-flop's Q output constitutes the most significant bit (MSB).

Based on a timing diagram analysis of this circuit, determine whether it counts in an *up* sequence (00, 01, 10, 11) or a *down* sequence (00, 11, 10, 01). Then, determine what would have to be altered to make it count in the other direction.

Solution: 1) This counter circuit counts in the *down* direction.

2) Diagram:

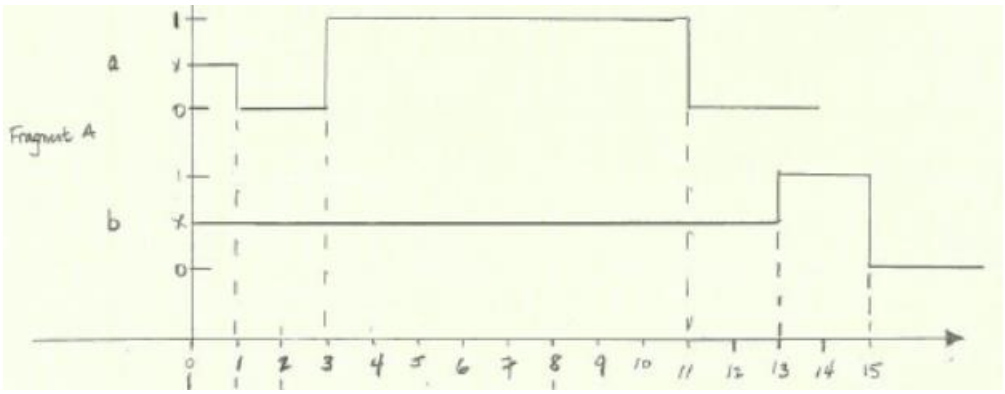


Q4) Draw the waveform of following Verilog code outputs

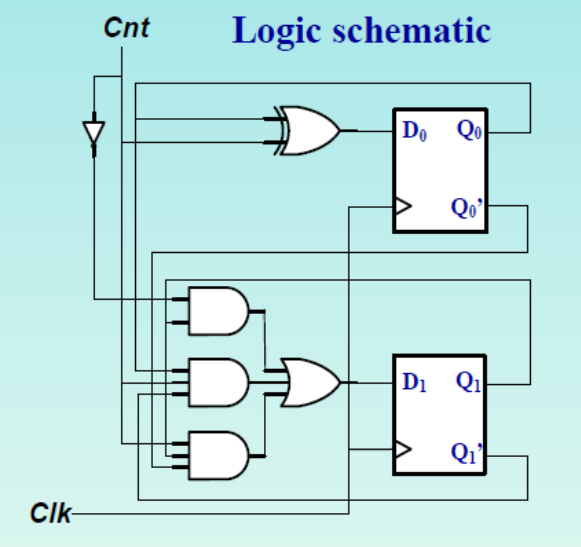
```
`timescale 1ns/1ps

Module (a,b,clk)

Input clk;

Output reg a,b;

Initial begin

a=1'bx;

b=1'bx;

end

always @(posedge clk)
```

```
begin
a = #1 0;
a = #2 1;
a=#8 0;
b=#2 1;
b=#2 0;
end
endmodule
```

ANS:



Q5. Derive the state table and the state diagram for the modulo-4 FSM shown below



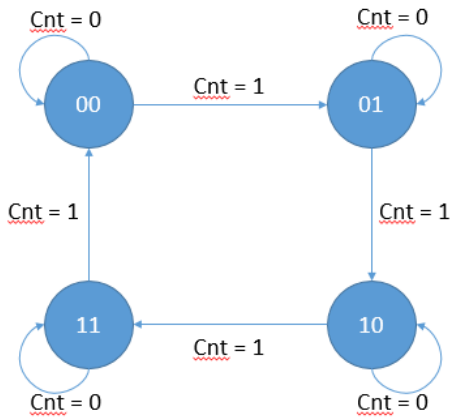
Solution:

State Table:

Based on the circuit operation, fill in the values for the table as shown

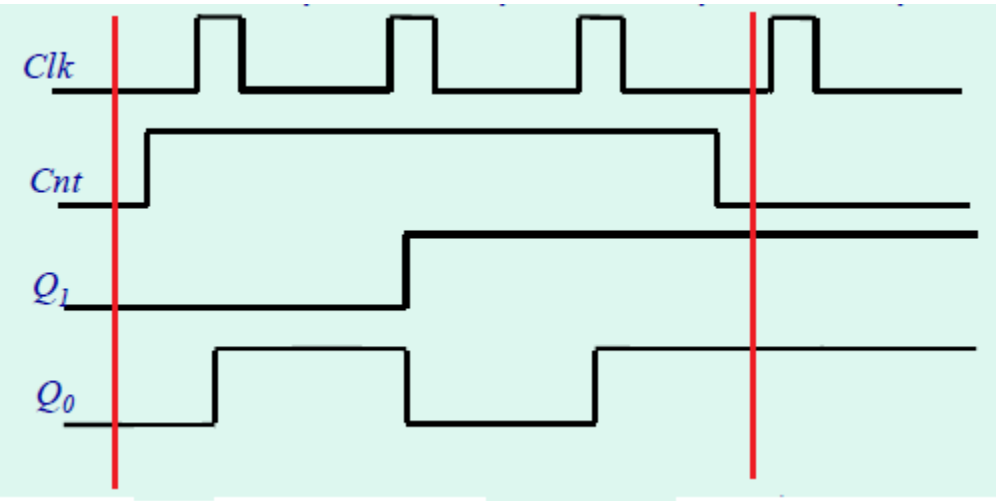
Present State	**Next State**	
$Q_1 Q_0$	Q_1(next) Q_0(next)	
	Cnt = 0	Cnt = 1
00	00	01
01	01	10
10	10	11
11	11	00

State Diagram



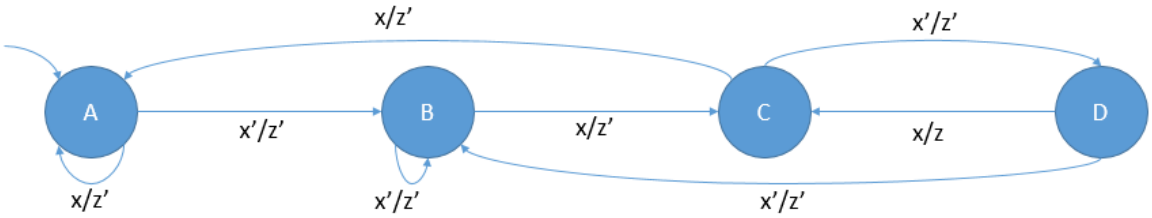
Based on the table we can create a state transition diagram. We observe that the values depend on Cnt, so value of Cnt = 0 or 1 determines the direction of the state change

Timing Diagram



We can draw the timing graph based on the flip flop action for state change and the parameter that dictates the state change. Based on these values we trace the patterns for Q_1(new) and Q_0(new). The red lines in the timing graph denotes the values when Cnt = 0, which is the normal count operation. Q_1(new) = Q_1; Q_0(new) = Q_0.

Q6. Answer the questions for the FSM shown below:



X – Input

Z – Output

a) What is the minimum number of Flip-Flops required for the FSM implementation?
Answer: Since the FSM has 4 state transitions we would require 2 Flip Flops

b) Is this a Moore or a Mealy FSM?
Answer: Since the FSM operation depends on the output state Z and the input X, the FSM is a Mealy FSM.

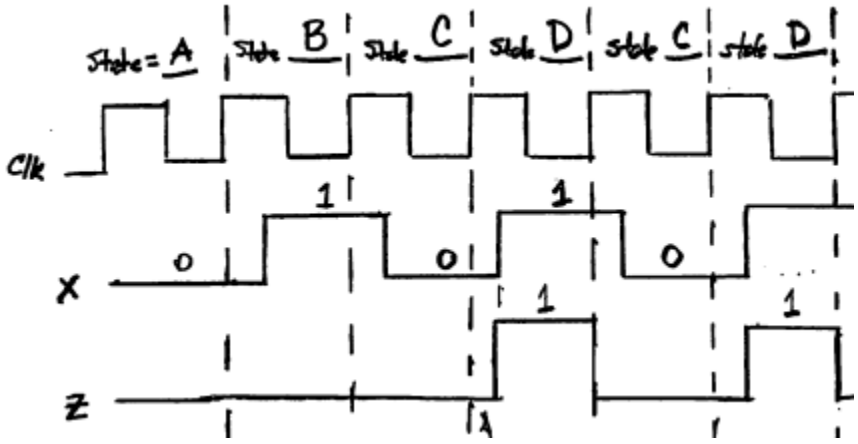
c) Complete the Truth Table (state table) for values of Z and next states (N3, N2, N1, N0). The present states are (S3, S2, S1, S0). States are A=0001, B=0010, C=0100, D=1000

State	S3	S2	S1	S0	X		Z	N3	N2	N1	N0	State
A	0	0	0	1	0		***0***	***0***	***0***	***1***	***0***	**B**
A	0	0	0	1	1		***0***	***0***	***0***	***0***	***1***	**A**
B	0	0	1	0	0		***0***	***0***	***0***	***1***	***0***	**B**
B	0	0	1	0	1		***0***	***0***	***1***	***0***	***0***	**C**
C	0	1	0	0	0		***0***	***1***	***0***	***0***	***0***	**D**
C	0	1	0	0	1		***0***	***0***	***0***	***0***	***1***	**A**
D	1	0	0	0	0		***0***	***0***	***0***	***1***	***0***	**B**
D	1	0	0	0	1		***1***	***0***	***1***	***0***	***0***	**C**

For Z: The output of the FSM is only when the state is D and input is X=1

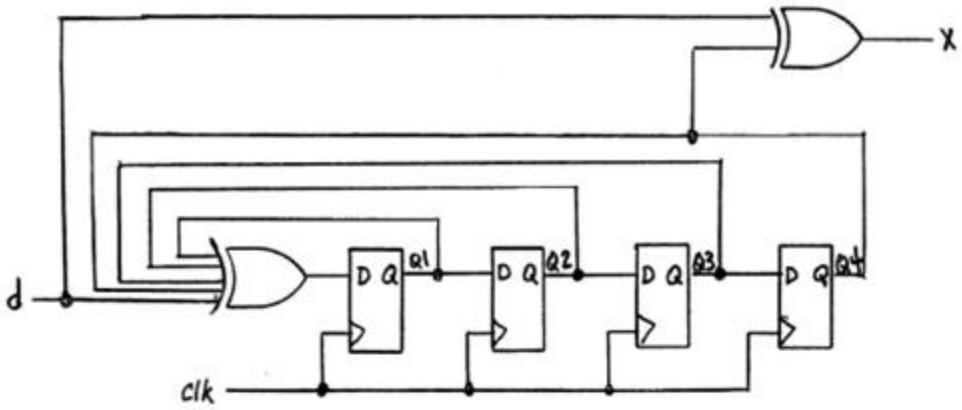
The values of N3->N0 can be inferred from the state diagram state transitions.
E.g. For the first two entries in the table (Corresponding to state A), if X=1 the state remains A, else if X=0 then the state transitions to B. (First two entries for next state)

d) Draw the timing Graph for the FSM if the values of clk and X are given as shown:
(Assume initially Z =0 and state = A)
Answer:



The timing diagram can be inferred based on the input and the state transition. E.g. If X = 0 and Z = 0 for state A, the next state is B.

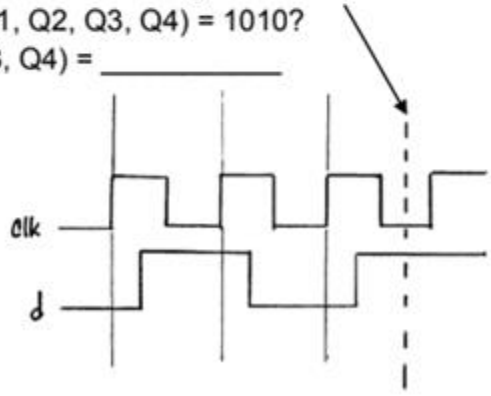
Q7) For the sequential circuit shown below, answer the questions (a) and (b)



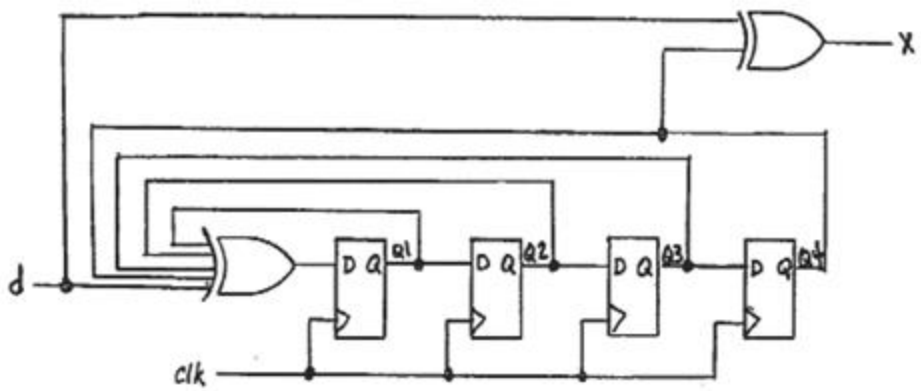
(a) (12%) Fill in the table for the next state values of the four flip-flops and the output x given the current state of the flip-flops and the input d. Assume setup and hold times are met for the flip-flop inputs.

Q1	Q2	Q3	Q4	d	$Q1_{Next}$	$Q2_{Next}$	$Q3_{Next}$	$Q4_{Next}$	X
0	1	0	1	0					
1	1	0	0	1					
1	1	1	1	1					

(b) (8%) For the timing diagram below, what is the value of Q1, Q2, and Q3 *at the time indicated by the dashed line* in the figure if the initial value for (Q1, Q2, Q3, Q4) = 1010?
(Q1, Q2, Q3, Q4) = _____



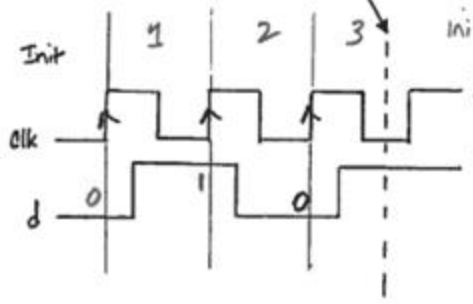
Solution:



(a) (12%) Fill in the table for the next state values of the four flip-flops and the output x given the current state of the flip-flops and the input d. Assume setup and hold times are met for the flip-flop inputs.

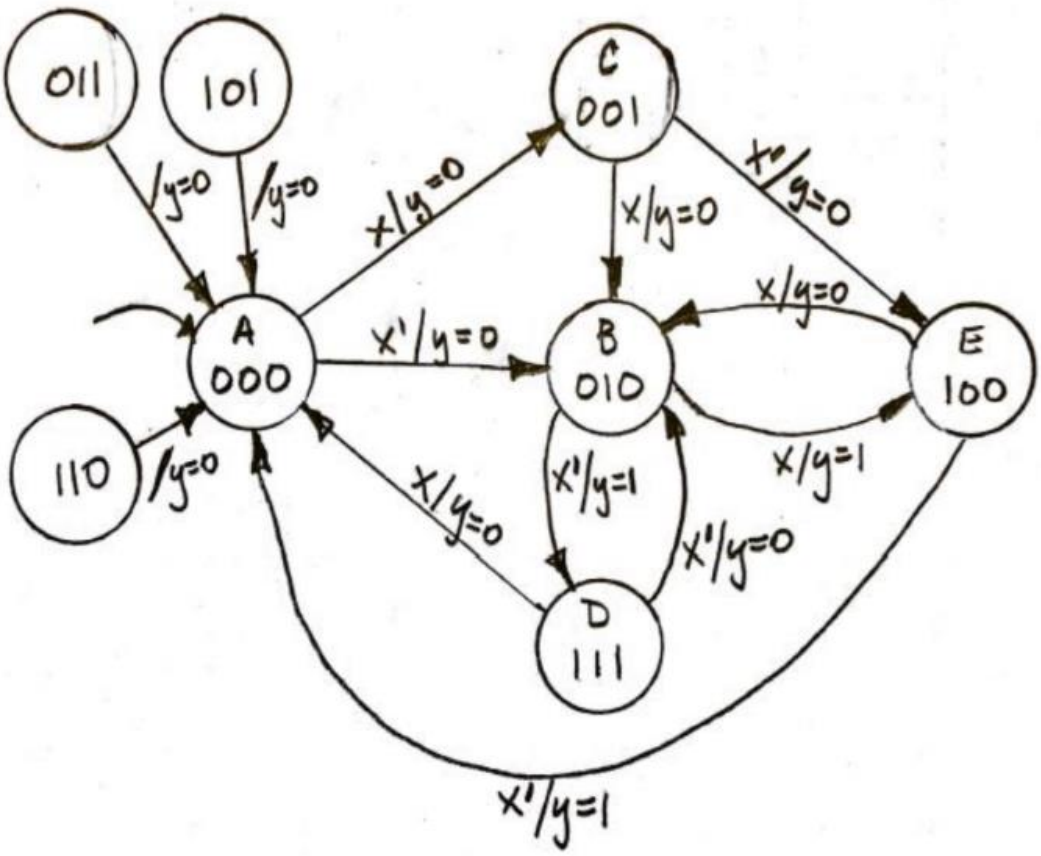
Q1	Q2	Q3	Q4	d	Q1Next	Q2Next	Q3Next	Q4Next	X
0	1	0	1	0	0	0	1	0	1
1	1	0	0	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	0

(b) (8%) For the timing diagram below, what is the value of Q1, Q2, and Q3 *at the time indicated by the dashed line* in the figure if the initial value for (Q1, Q2, Q3, Q4) = 1010?
(Q1, Q2, Q3, Q4) = 0101



	Q1	Q2	Q3	Q4	d	Q1n	Q2n	Q3n	Q4n
Init	1	0	1	0	0	0	1	0	1
1	0	1	0	1	1	1	0	1	0
2	1	0	1	0	0	0	1	0	1

Q8) Given the sequential circuit below, answer parts (a) and (b).



a) Fill in the table for the next state values of the three flip flops given the current state of the flipflops and the input x. Assume setup and hold times are met for the flip-flop inputs.

S2	S1	S0	X	N2	N1	N0	y
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
1	0	0	0				
1	0	0	1				
1	1	1	0				
1	1	1	1				

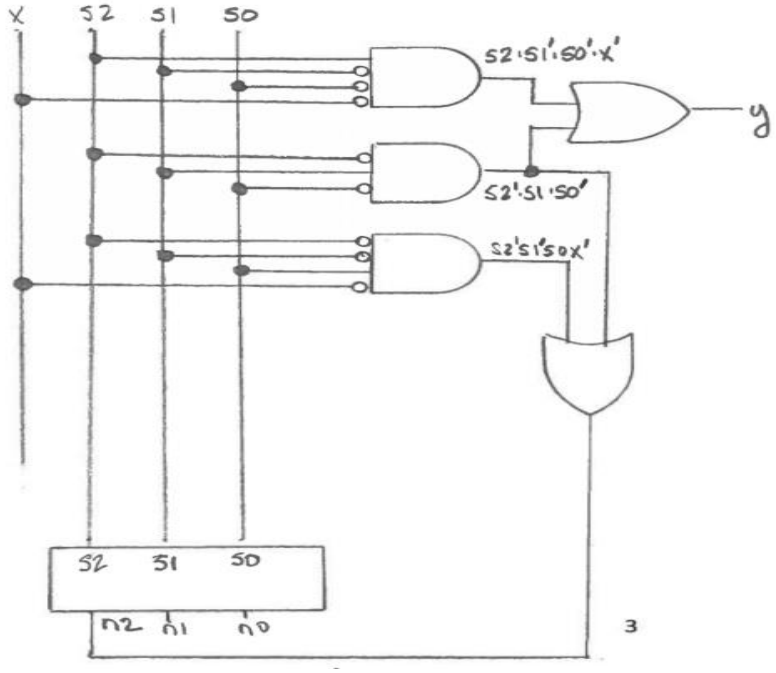
b) Using K-Map minimize the combinational logic for n2 and y. Show the hardware implementation of n2 and y using AND, OR and NOT gates.

Solution:

A)

s2	s1	s0	x	n2	n1	n0	y
0	0	0	0	0	1	0	0
0	0	0	1	0	0	1	0
0	0	1	0	1	0	0	0
0	0	1	1	0	1	0	0
0	1	0	0	1	1	1	1
0	1	0	1	1	0	0	1
1	0	0	0	0	0	0	1
1	0	0	1	0	1	0	0
1	1	1	0	0	1	0	0
1	1	1	1	0	0	0	0

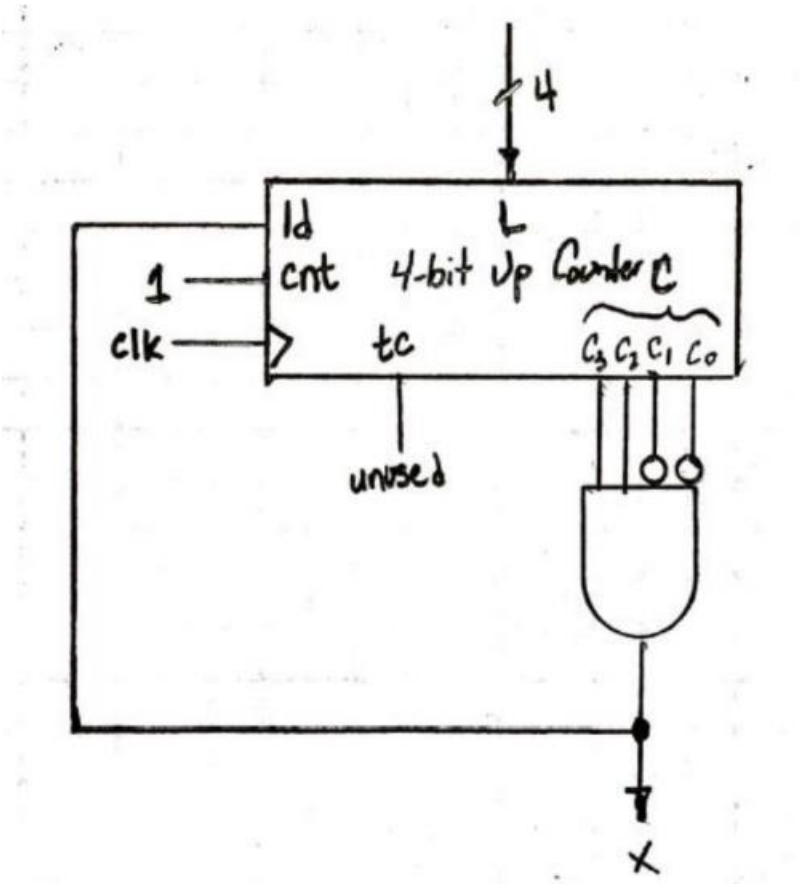
B)



Q9) Consider the 4-bit up counter as shown below.

a) Show the count sequence for the given input until the output X becomes 1.
b) If the frequency of clock is 900 MHz, what is the output frequency of X.

0100



Q10) The Verilog code on the following two pages describes the Finite State Machine for Problem 2 on this exam. There is missing Verilog code in five places that are boxed. Write the Verilog statements to go where the questions marks (i.e., ??) are in the code .

```
module ExamIIProb4_2019 (y,x,Clk,Rst)
    output y;
    input x, Clk, Rst;
    reg y;

    parameter A=3'b???, B=3'b???, C=3'b???, D=3'b111; E=3'b100
    reg [??:0] State, NextState;                                    (a)

//Combinational Logic
    always @(State, x)
    begin
      case(State)
        A: begin
             if(x==0)begin
                 y ? ??;                                           (b)
                 NextState ? ??;
             end
             else begin
                 y=0;
                 NextState=C;
             end
           end
        B: begin
             if(x==0) begin
                 y=1;
                 NextState=D;
             end
             else begin
                 y=1;
                 NextState=E;
             end
           end
```

```
        C: begin
             y=??;
             if(x==??)NextState=??;
             else NextState=??;
           end
        D: begin
             y=0;
             if(x==1) NextState=A;
             else NextState=B;
           end
        E: begin
             if(x==0) begin
             y=1;
             NextState=A;
             end
             else begin
             y=0;
             Nextstate=B;
             end
           end
        Default: begin
             NextState ? ??;
             y ? ??;
           end
        endcase
     end // end of combinational always @
//Sequential
   always @(??? Clk)
   begin
     if(Rst == 1) State ? ??;
     else
     State ? ??;
   end // end of sequential always @
endmodule
```

(c)

(d)

(e)

Solution:

(a) (4%)

A = 3'b 000 B = 3'b 010 , C = 3'b 001

reg [2:0] state, NextState

(b) (4%)

```
y = 0;
NextState = B;
```

(c) (4%)

y = 0

```
if (x == 0) NextState = E;
else NextState = B;
```

or

```
if (x == 1) NextState = B;
else NextState = E;
```

(d) (4%)

```
NextState = A;
y = 0;
```

(e) (4%)

```
always @ (posedge Clk)
if (Rst == 1) State <= A;
else
State <= NextState
```

Q11) Find the output of z from the following verilog code when inputs are x=1 and y =0

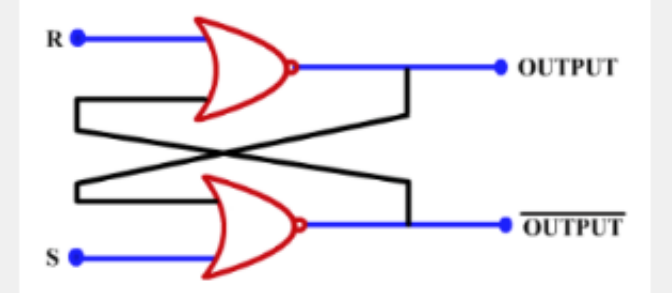
```
module check(
input x,
input y,
output z
);

assign z = (~x & ~y) |(x & y);

endmodule
```

Solution:

z= 0

Q12) Fill in the blanks for the state table of given circuits.

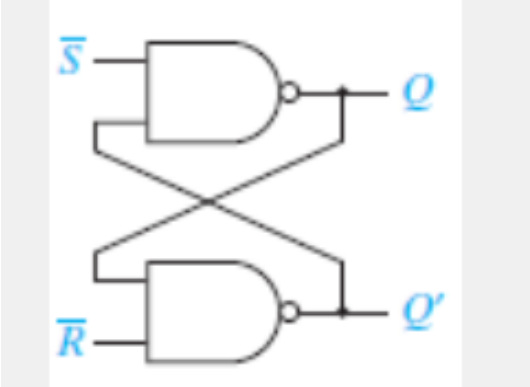
a)



b)

Input		Current Output	Next Output
S	R		
0	0	1	
0	1		
1	0		

b)



Input		Current Output	Next Output
S'	R'		
1	1	1	
0	1		
1	0		

Solution:

a)

Input		Current Output	Next Output
S	R		
0	0	1	1
0	1	1	0
1	0	0	1

b)

Input		Current Output	Next Output
S'	R'		
1	1	1	1
0	1	1	1
1	0	1	0