

King Fahd University of Petroleum and Minerals
College of Computer Science and Engineering
Computer Engineering Department

COE 202: Digital Logic Design (3-0-3)
Term 161 (Fall 2016)
Final Exam
Wednesday, December 11, 2017

Time: 120 minutes, Total Pages: 14

Name: _____ **ID:** _____ **Section:** _____

Notes:

Do not open the exam book until instructed

Calculators are not allowed (*basic, advanced, cell phones, etc.*)

Answer all questions

All steps must be shown

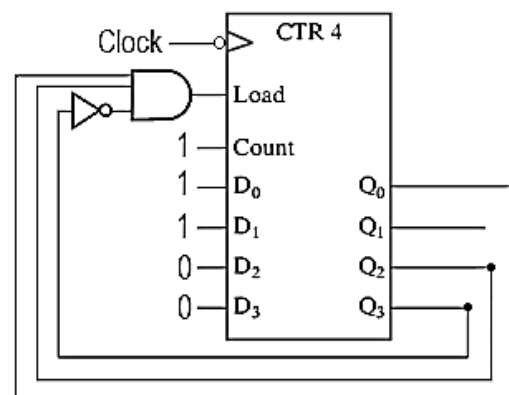
Any assumptions made must be clearly stated

Question	Maximum Points	Your Points
1	16	
2	7	
3	6	
4	15	
5	9	
6	6	
7	6	
8	15	
9	10	
Total	90	

Question 1: Fill in the Spaces: (Show all work needed to obtain your answer)**(16 points)**

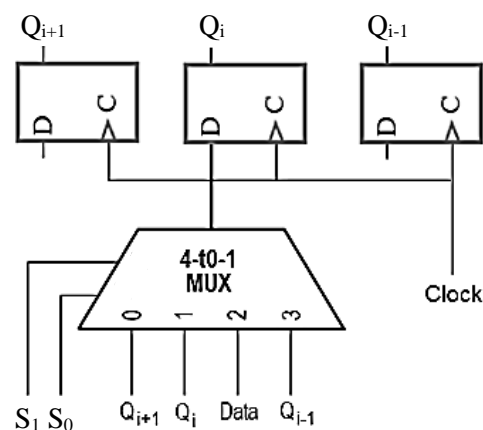
1. A high value to the two R and S inputs of the NAND-gate latch will No Change (Change/No Change) the state/output of the latch. **(1 point)**
2. Asynchronous reset to a flip flop doesn't depend on the clock input True (True/False). **(1 point)**
3. Given a synchronous sequential circuit with 9 states, the minimum number of flip flops required to implement the circuit is 4 flip flops and the number of unused states is 7 states. **(2 points)**

4. The following circuit shows a parallel load binary counter, the range of the counter is 3 to 5. **(2 points)**

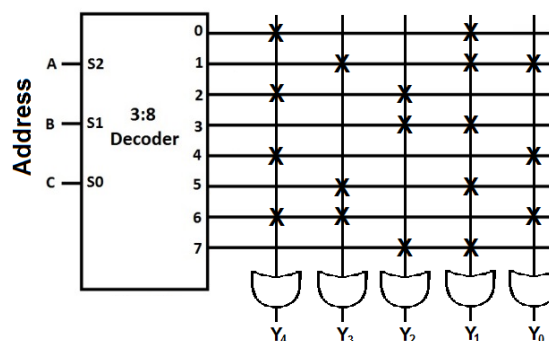


5. The below figure shows connections to the **D** input of stage **i** in a multi-function register of D-type flip flops. Study the circuit and fill in the missing information in the table below (empty slots) only for supported register functions. **(2 points)**

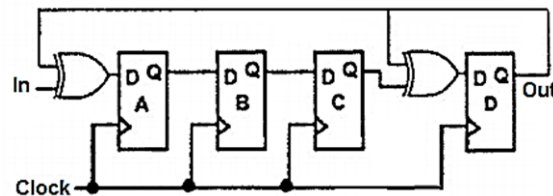
$S_1 S_0$	Register Function (where applicable)
<u>00</u>	Shift right
<u>11</u>	Clear register
<u>01</u>	No change in output
<u>10</u>	Load Data input



6. In the ROM circuit shown, **X** indicates a connection. At address $ABC = 110$, the ROM stores the data $Y_4 Y_3 Y_2 Y_1 Y_0 =$ 11001. **(1 point)**



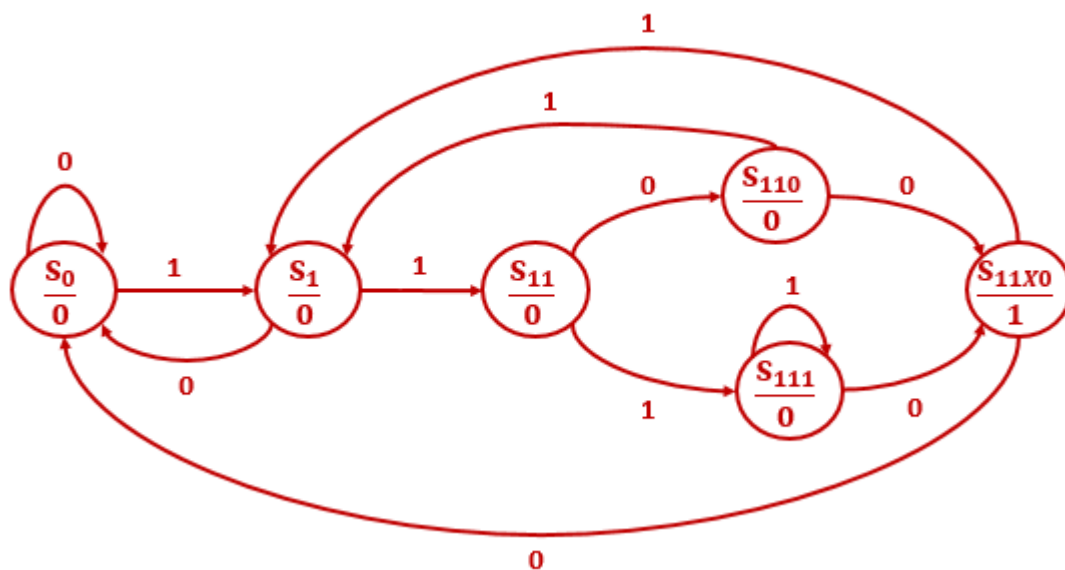
7. For a 4-bit synchronous binary counter (outputs Q_3 , Q_2 , Q_1 and Q_0), with input clock frequency of 48 MHz, the frequency of Q_1 is 12 MHz and the frequency of Q_3 is 3 MHz. (2 points)
8. Two RS level sensitive latches and one inverter are used to make an edge triggered flip flop which can be used to store up to two bits False (True/False). (1 point)
9. Consider the below 4-bit register. If the initial register contents (Q outputs) $ABCD$ are 0111 and the serial input is kept at 1, show the contents $ABCD =$ 1001 of the register after two clock pulses. (2 points)



10. A state machine with two inputs and three outputs will have 4 (how many) arcs going out from each state to any other states. (1 point)
11. For any given problem/requirement, the number of states required by a Mealy machine or a Moore machine is always the same False (True/False). (1 point)

Question 2:**(7 points)**

Derive the state diagram of a synchronous Moore sequential circuit that receives a serial input Y and produces a serial output F that is set to **1** when the circuit detects the sequence **11X0**, where **X** represents don't care.



Question 3:**(6 points)**

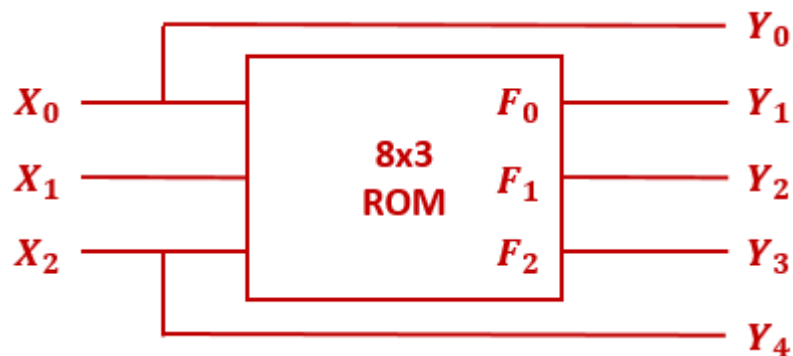
Design a combinational circuit using a ROM. The circuit accepts a 3-bit number $X = X_2X_1X_0$ and generates an output binary number Y equal to $3X + 4$. The ROM should contain a minimum number of columns. Fill the truth table and ROM table below and draw the block diagram.

Truth Table

<i>Inputs</i>			<i>Outputs</i>				
X_2	X_1	X_0	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1	0	0
0	0	1	0	0	1	1	1
0	1	0	0	1	0	1	0
0	1	1	0	1	1	0	1
1	0	0	1	0	0	0	0
1	0	1	1	0	0	1	1
1	1	0	1	0	1	1	0
1	1	1	1	1	0	0	1

ROM Table

<i>Inputs</i>			<i>Outputs</i>		
X_2	X_1	X_0	F_2	F_1	F_0
0	0	0	0	1	0
0	0	1	0	1	1
0	1	0	1	0	1
0	1	1	1	1	0
1	0	0	0	0	0
1	0	1	0	0	1
1	1	0	0	1	1
1	1	1	1	0	0

Block Diagram

Question 4:**(15 points)**

Given the following state table:

a) Obtain minimal sum-of-products equations for the next state and output. **(12 points)**b) Draw the circuit diagram. **(3 points)**

Present State	Next State		Output
A B C	x = 0	x = 1	z
0 0 0	0 0 1	0 0 0	0
0 0 1	0 1 0	0 0 0	1
0 1 0	0 1 1	0 0 0	1
0 1 1	1 0 0	0 0 0	1
1 0 0	1 0 1	0 0 1	0
1 0 1	1 0 1	0 1 1	1

Part a)**Solution1: Using Don't Care for the unused states**

	K-Map for D _A				K-Map for D _B				K-Map for D _C				K-Map for z	
	C x				C x				C x				C	
A B	0 0	0 1	1 1	1 0	0 0	0 1	1 1	1 0	0 0	0 1	1 1	1 0	C=0	C=1
0 0								1	1					1
0 1				1	1				1				1	1
1 1	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1 0	1			1			1		1	1	1	1		1

$$D_A = A x' + B C x' \quad D_B = B C' x' + A C x + A' B' C x'$$

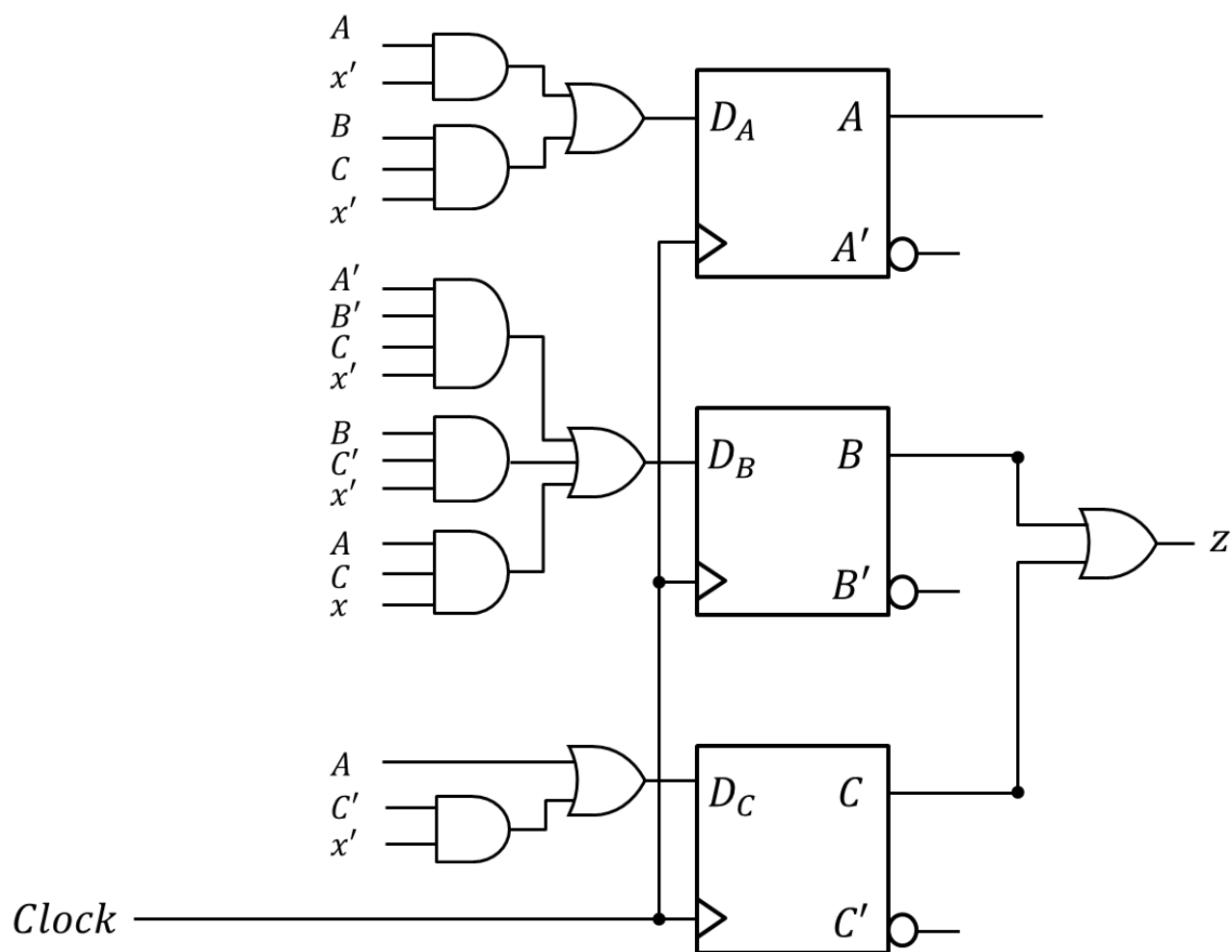
$$D_C = A + C' x' \quad z = B + C$$

Solution2: Forcing transition from the unused states to state 000

	K-Map for D _A				K-Map for D _B				K-Map for D _C				K-Map for z	
	C x				C x				C x				C	
A B	0 0	0 1	1 1	1 0	0 0	0 1	1 1	1 0	0 0	0 1	1 1	1 0	C=0	C=1
0 0								1	1					1
0 1				1	1				1				1	1
1 1													X	X
1 0	1			1			1		1	1	1	1		1

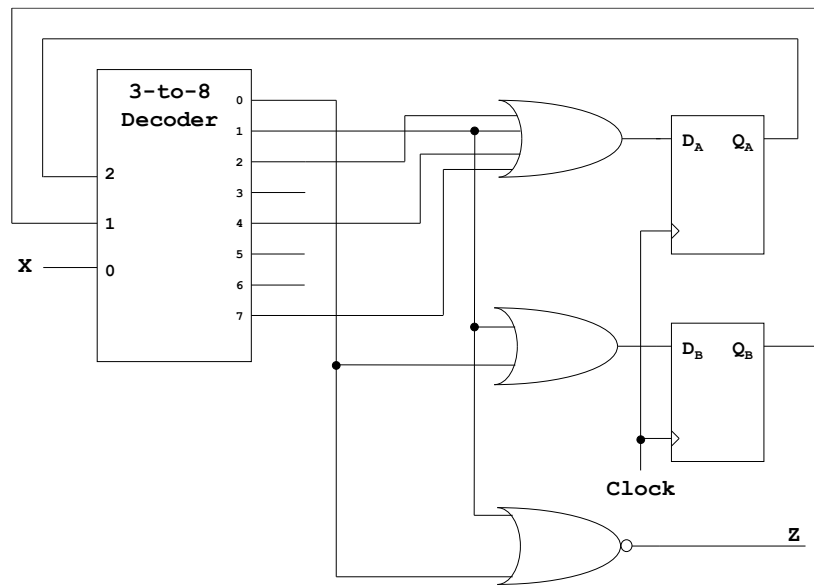
$$D_A = A B' x' + A' B C x' \quad D_B = A' B C' x' + A B' C x + A' B' C x'$$

$$D_C = A B' + A' C' x' \quad z = B + C$$

Part b) Circuit Diagram for Solution 1

Question 5: Consider the following sequential circuit:

(9 points)



- a) Provide a state table for the given circuit showing the Present State, the input **X**, the Next State, and the output **Z**. (8 points)

$$D_A = \sum m(1,2,4,7), \quad D_B = \sum m(0,1), \quad Z = \prod M(0,1)$$

Q_A	Q_B	X	Q_A^+	Q_B^+	Z
0	0	0	0	1	0
0	0	1	1	1	0
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	0	0	1
1	1	1	1	0	1

- b)** Is the circuit type *Mealy* or *Moore*? Justify your answer.

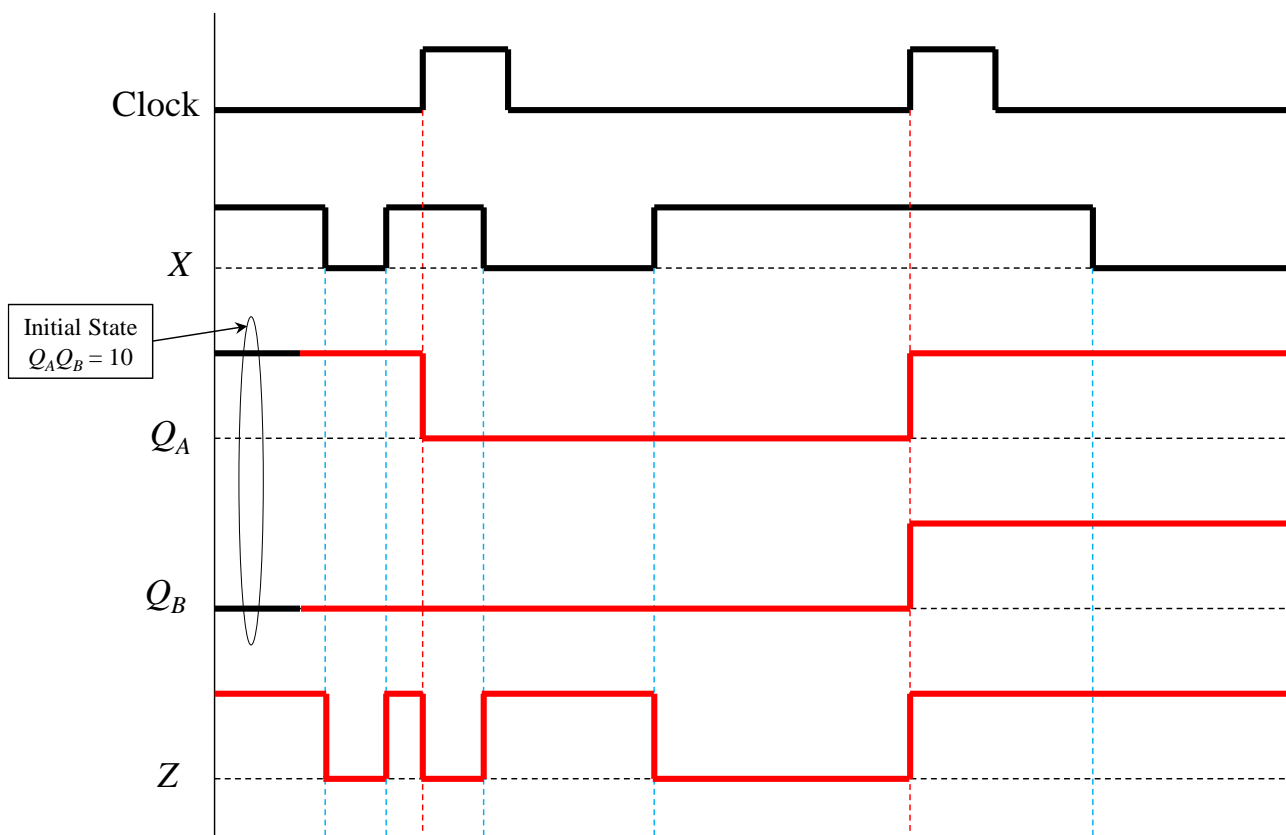
(1 point)

Moore since $Z = \Pi M(0,1) = (Q_A + Q_B + X)(Q_A + Q_B + X') = Q_A + Q_B$ which is a function of the present states only. This is also clear from the state table obtained in part (a).

Question 6:**(6 points)**

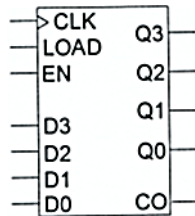
Consider the following state table. Assume that the initial state of the circuit implementation of the given state table is ($Q_A Q_B = 10$). Draw the waveforms of Q_A , Q_B , and Z for the given 2 clock cycles in response to the shown applied input X . *Ignore propagation delays, setup times, and hold times. Assume that the circuit uses rising edge-triggered D-FF(s).*

Present State		X	Next State		Z
Q_A	Q_B		D_A	D_B	
0	0	0	0	1	1
0	0	1	1	1	0
0	1	0	1	0	0
0	1	1	0	0	0
1	0	0	1	0	0
1	0	1	0	0	1
1	1	0	0	0	1
1	1	1	1	0	1

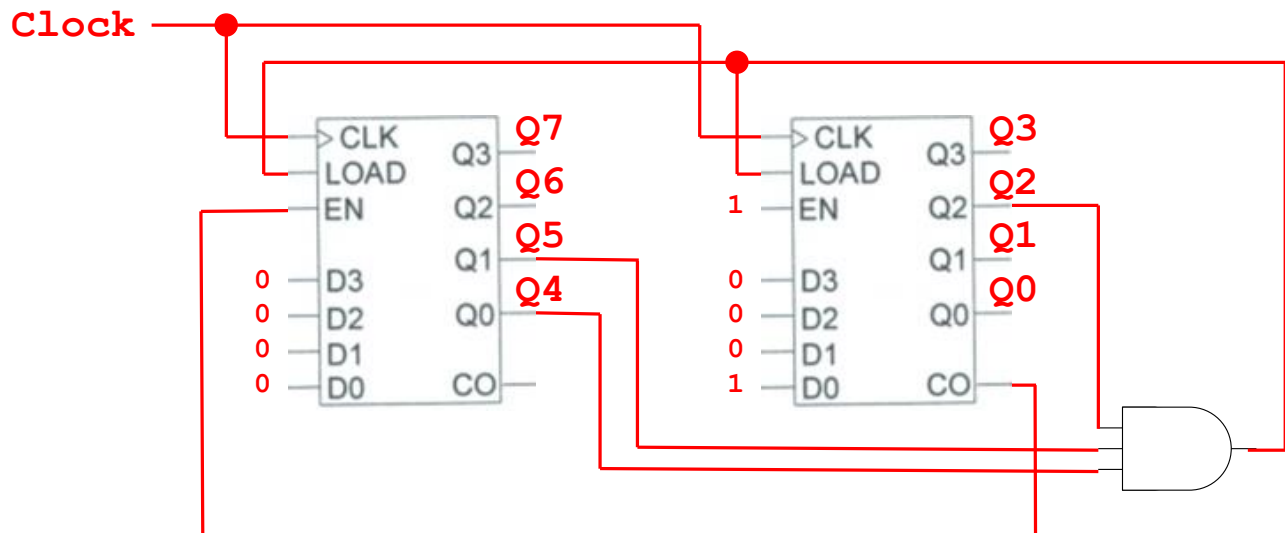


Question 7:**(6 points)**

Use minimal external gates and as many of the following counter as necessary to design a counter that counts from **1** to **52** and then repeats. Note that the operation of the provided counter is according to the table to the right of the counter. Note also that CO stands for Carry-output which gets set to 1 when the maximum count of 15 is reached. Assume that the counter is initially loaded with the value 1. Properly label (Q0 – Q7) and (D0 – D7) of the designed counter.



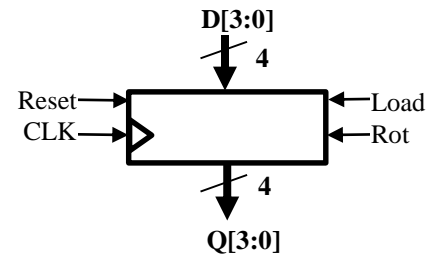
<i>LOAD</i>	<i>EN</i>	<i>Operation</i>
0	0	Hold count
0	1	Increment count
1	X	Parallel load



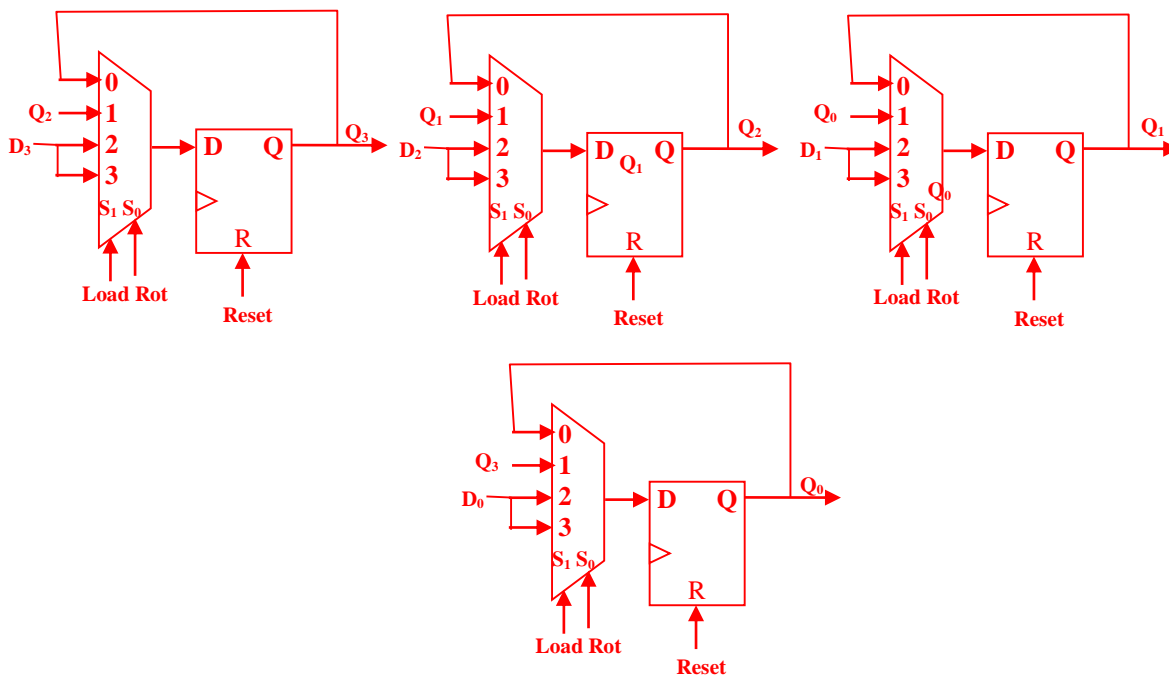
Question 8:**(15 points)**

Using D-FFs and any other components, design a 4-bit rotator register with direct asynchronous reset that has two control inputs; Load and Rot with the functionality shown in the table below:

Reset	Load	Rot	Function
0	0	0	No Change (Q stay as is)
1	X	X	asynchronous reset: Q = 0
0	1	X	Load: Q = D
0	0	1	Rotate Left: Q3=Q2, Q2=Q1, Q1=Q0, Q0=Q3



a) Draw the circuit diagram

(5 points)

b) Write a **behavioral** Verilog description of the above rotator register

(6 points)

```
module rotator4 (input [3:0] D, reset, clk, load, rot, output reg [3:0] Q);
```

```
always @ (posedge clk, posedge reset)
if (reset) Q<= 0;
elseif (load) Q<=D;
else if (rot) Q <= {Q[2:0],Q[3]};
endmodule
```

c) Write a test bench to test the above rotator register. Let the clock cycle = 20 time units. First, reset the register, then load the register with **1010**, then do nothing for two clock cycles, then rotate the register **3** times.

(4 points)

```
module tb_rot ();
wire [3:0] Q;
reg [3:0] D;
reg clk, reset, load, rot;
```

```
rotator ml (D, reset, clk, load, rot, Q) ;
```

```
initial begin                /reset and clock sequence
```

```
reset = 1; clk=0;
```

```
#5 reset =0 ; forever #10 clk=~clk;
```

```
end
```

```
initial begin
```

```
load=0 ;      rot=0 ;      D = 4'b1010 ;      //or D=10 .. initialize inputs
```

```
@ (negedge clk) load= 1 ;      //load the register with 1010
```

```
@ (negedge clk) load= 1 ;      //deactivate the load signal – one clock cycle after load – do nothing
```

```
@ (negedge clk)      ;      //another clock cycle after load– do nothing
```

```
@ (negedge clk) rot=1 ;      //1st rotation
```

```
@ (negedge clk)      ;      //2nd rotation, rot signal remains 1
```

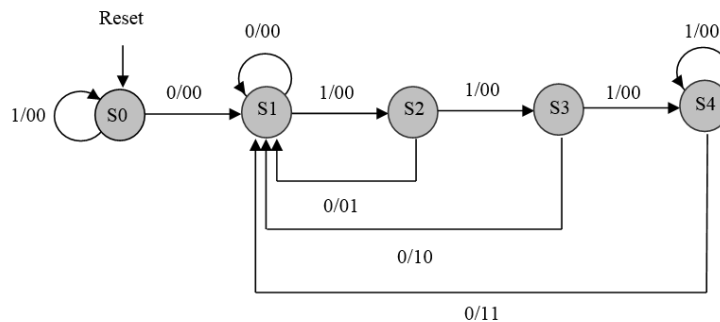
```
@ (negedge clk)      ;      //3rd rotation, rot signal remains 1
```

```
end
```

```
endmodule
```

Question 9:**(10 points)**

- a) Write a **behavioral** Verilog description of a sequential circuit with the state diagram below. The circuit has an asynchronous **Reset** input, one input **X**, and two outputs: **Y** and **Z**. Use the following state encodings: **S0=000**, **S1=001**, **S2=010**, **S3=100**, **S4=111**. If the circuit ever gets into any of the unused states, it will go to state **S0** no matter what the input value is. **(6 points)**



```

module Mealy_fsm (output wire y,z, input x, clk, reset );

localparam S0 = 2'b000, S1=2'b001, S2=2'b010, S3=2'b011, S4=100 ;      //symbolic names for state values

reg [2:0] state ; //state register

assign y = ((state== s3) | (state== s4)) & ~x ; //y=1 only if we are in state S3 or state S4, and x is 0

assign z = ((state== s2) | (state== s4)) & ~x ; //y=1 only if we are in state S2 or state S4, and x is 0

always @(posedge clk, posedge reset) //This is only for the state transition
  if (reset) state <= S0;

  else case (state)
    S0: if (!x) state <= S1 ;      //if x=1, state remain at S0
    S1: if (x) state <= S2 ;      //if x=0, state remain at S0
    S2: if (x) state <= S3 ; else state <= S1 ;
    S3: if (x) state <= S4 ; else state <= S1 ;
    S4: if (!x) state <= S1 ; else state <= S1 ;
    default: state <= S0 ;      //all non-used states go to S0
  endcase

endmodule

```

- b) Write a test bench to test the circuit. Let the clock cycle = 20 time units. First, reset the circuit, then apply the following input sequence to X: **0, 1, 0, 1, 1, 1, 0**. **(4 points)**

```

module tb_fsm ();
wire y,z ;      reg      clk, reset,x ;
Mealy_fsm m1 (y, z, x, clk, reset );

initial begin          /reset and clock sequence
reset = 1; clk=0;
#5 reset =0 ; forever #10 clk=~clk;
end

initial begin
x=0 ;                //1st input bit x=0
@ (negedge clk) x=1 ;                //2nd input bit x=1
@ (negedge clk) x=0 ;                //3rd input bit x=0
@ (negedge clk) x=1 ;                //4th input bit x=1
@ (negedge clk) ;                    //5th input bit, x remain 1
@ (negedge clk) ;                    //6th input bit, x remain 1
@ (negedge clk) x=0 ;                //7th input bit x=0
end
endmodule

```