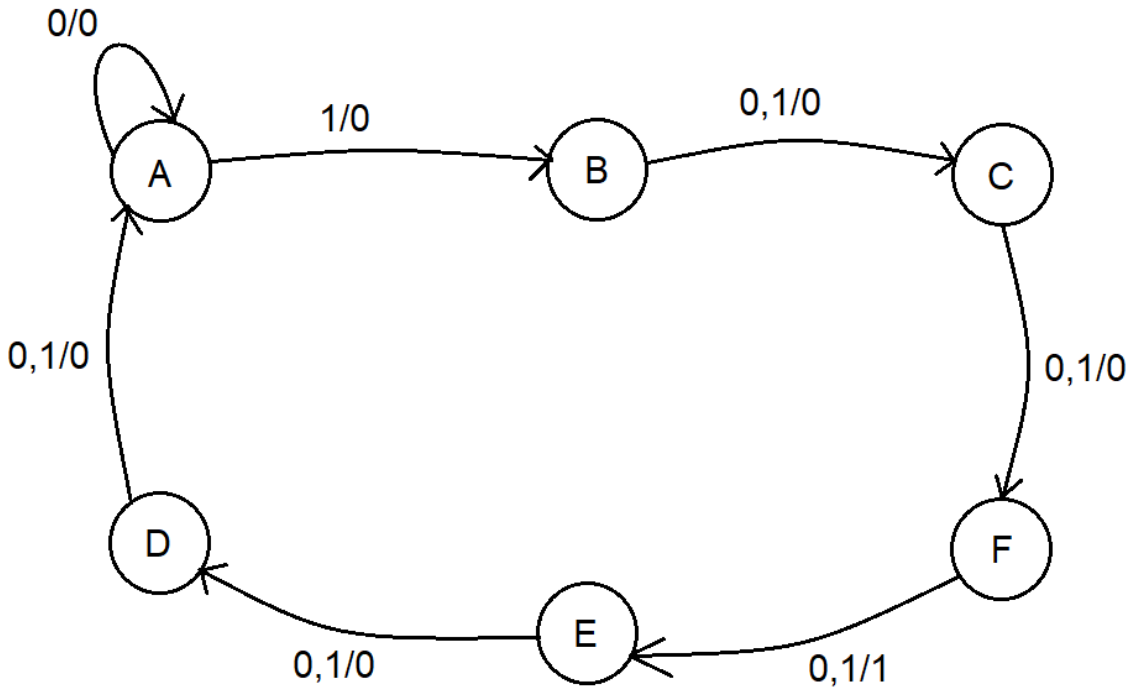
Assignment-3 (ELL201)
Aditya Rahi (2020MT10784)

X3 = 8 % 8 = 0
X4 = 4 % 8 = 4

Therefore, the sequence to be generated is : {0,0,0,1,0,0}

State Diagram for FSM:



Here, **Mealy FSM** is implemented.

Number of flip flops required:

6 states are used to make the FSM. Hence,
No. of flip-flops required ≥ $\log_2$(no. of states)= $\log_2 6$
⇒ **No. of flip-flops required = 3**

The values assigned to states are:

A: 000 B: 001
C: 010 D: 011
E: 100 F: 101

State Table:

MSB is Q2, LSB is Q0 and the state in the order is Q2 Q1 Q0.

Current State			**Input**	**Output**	**Next State**			**D Flipflops**		
Q2	Q1	Q0	X	Y	Q2	Q1	Q0	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0	1
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	1	0	0	1	0
0	1	0	0	0	0	1	1	0	1	1
0	1	0	1	0	0	1	1	0	1	1
0	1	1	0	1	1	0	0	1	0	0
0	1	1	1	1	1	0	0	1	0	0
1	0	0	0	0	1	0	1	1	0	1
1	0	0	1	0	1	0	1	1	0	1

1	0	1	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

The K-Maps are:

D_0 is given by:

D0 = Q2Q1'Q0' + Q2'Q1Q0' + Q2'Q0'X

Q2Q1\Q0X	00	01	11	10
00	0	1	0	0
01	1	1	0	0
11	0	0	0	0
10	1	1	0	0

D_1 is given by:

D1 = Q2'Q1'Q0 + Q2'Q1Q0'

Q2Q1\Q0X	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	0	0	0	0
10	0	0	0	0

D_2 is given by:

D2 = Q2’Q1Q0 + Q2Q1’Q0’

Q2Q1\Q0X	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	0	0	0	0
10	1	1	0	0

Y (Output) is given by:
Y = Q2’Q1Q0

Q2Q1\Q0X	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	0	0	0	0
10	0	0	0	0

Verilog Code:

Positive Edge triggered D Flip-Flop:

Get Started flip_flop.v

C: > Users > DELL > flip_flop.v

```
1  module flip_flop(clk,next_state,changed_state);
2      input clk, next_state;
3      output changed_state;
4      reg changed_state;
5      always @(posedge clk)
6      begin
7          changed_state = next_state;
8      end
9  endmodule
```

Mealy FSM:

Get Started | assn3.v

C: > Users > DELL > assn3.v

```
module assn3;
    reg clk;
    reg xReg;
    wire [2:0] q;
    wire yWire;
    reg Dinput0, Dinput1, Dinput2;
    assign yWire = (~q[2] & q[1] & q[0]);

    flip_flop FF2(clk,Dinput2,q[2]);
    flip_flop FF1(clk,Dinput1,q[1]);
    flip_flop FF0(clk,Dinput0,q[0]);

    always @(negedge clk)
    begin
        Dinput2 = (~q[2] & q[1] & q[0])|(q[2] & ~q[1] & ~q[0]);
        Dinput1 = (~q[2] & ~q[1] & q[0])|(~q[2] & q[1] & ~q[0]);
        Dinput0 = (q[2] & ~q[1] & ~q[0])|(~q[2] & q[1] & ~q[0])|(~q[2] & ~q[0] & xReg);
    end

    initial begin
        $dumpfile("assn3.vcd");
        $dumpvars(0,assn3);
        $monitor($time," Input : %b, States : %b %b %b,  Output : %b;",xReg,q[2],q[1],q[0],yWire);
        Dinput0 = 0;
        Dinput1 = 0;
        Dinput2 = 0;
        xReg = 0;
        clk = 1;
        #7 xReg = 1;
        #8 xReg = 0;
        #8 xReg = 1;
        #8 xReg = 0;
        #8 xReg = 1;
        #8 xReg = 0;
        #8 xReg = 1;
        #8 xReg = 0;
        #8
        $finish;
    end
    always #2 clk = ~clk;
endmodule
```

Output:

```
  0 Input : 0, States : 0 0 0,  Output : 0;
  7 Input : 1, States : 0 0 0,  Output : 0;
 12 Input : 1, States : 0 0 1,  Output : 0;
 15 Input : 0, States : 0 0 1,  Output : 0;
 16 Input : 0, States : 0 1 0,  Output : 0;
 20 Input : 0, States : 0 1 1,  Output : 1;
 23 Input : 1, States : 0 1 1,  Output : 1;
 24 Input : 1, States : 1 0 0,  Output : 0;
 28 Input : 1, States : 1 0 1,  Output : 0;
 31 Input : 0, States : 1 0 1,  Output : 0;
 32 Input : 0, States : 0 0 0,  Output : 0;
 39 Input : 1, States : 0 0 0,  Output : 0;
 44 Input : 1, States : 0 0 1,  Output : 0;
 47 Input : 0, States : 0 0 1,  Output : 0;
 48 Input : 0, States : 0 1 0,  Output : 0;
 52 Input : 0, States : 0 1 1,  Output : 1;
 55 Input : 1, States : 0 1 1,  Output : 1;
 56 Input : 1, States : 1 0 0,  Output : 0;
 60 Input : 1, States : 1 0 1,  Output : 0;
 63 Input : 0, States : 1 0 1,  Output : 0;
 64 Input : 0, States : 0 0 0,  Output : 0;
```

GTK Wave Output: