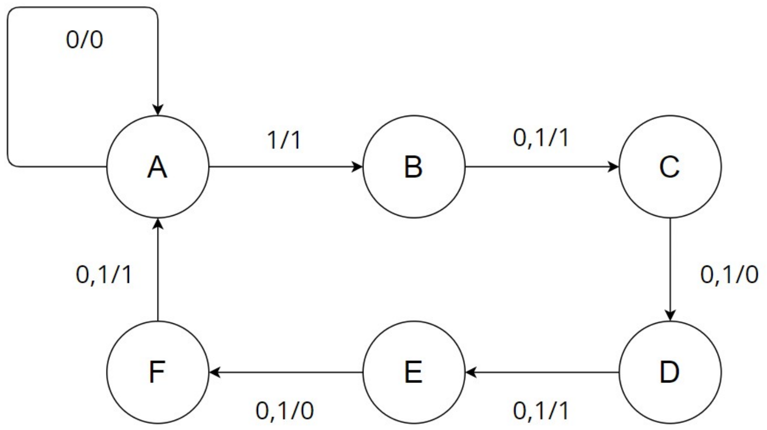
ELL201 Lab Project

Saket Kandoi

2021MT60265

Sequence to be generated: {1,1,0,1,0,1}

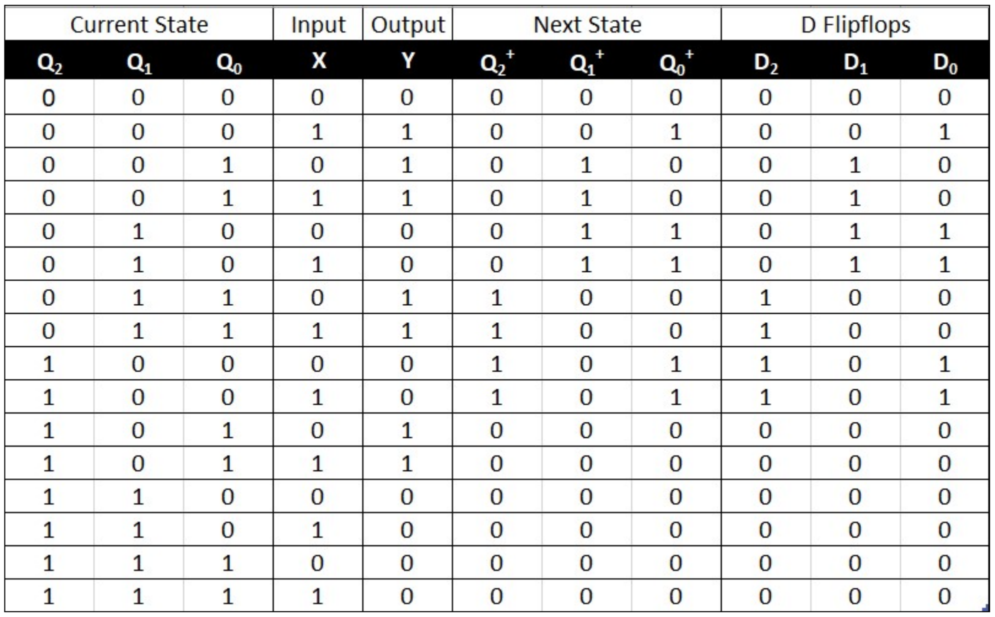
Mealy FSM



Idle state: A (000)
Unused state: A (000)
Minimum number of flip-flops needed $= \lceil \log_2 6 \rceil = 3$

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

Truth table

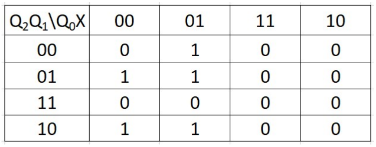


Current State			Input	Output	Next State			D Flipflops		
Q_2	Q_1	Q_0	X	Y	Q_2^+	Q_1^+	Q_0^+	D_2	D_1	D_0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	1	0	0	1
0	0	1	0	1	0	1	0	0	1	0
0	0	1	1	1	0	1	0	0	1	0
0	1	0	0	0	0	1	1	0	1	1
0	1	0	1	0	0	1	1	0	1	1
0	1	1	0	1	1	0	0	1	0	0
0	1	1	1	1	1	0	0	1	0	0
1	0	0	0	0	1	0	1	1	0	1
1	0	0	1	0	1	0	1	1	0	1
1	0	1	0	1	0	0	0	0	0	0
1	0	1	1	1	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

Note: Any stray states are redirected to idle state.
States are represented by $Q_2Q_1Q_0.D_0$ is the LSB and D_2 is the MSB.

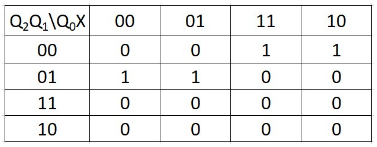
Karnaugh Maps

$$\begin{aligned}
D_0 &= Q_2Q_1'Q_0' + Q_2'Q_1Q_0' + XQ_2'Q_0' \\
D_1 &= Q_2'Q_1'Q_0 + Q_2'Q_1Q_0' \\
D_2 &= Q_2'Q_1Q_0 + Q_2Q_1'Q_0' \\
Y &= Q_2'Q_0 + Q_2'Q_1'X + Q_2Q_1'Q_0
\end{aligned}$$



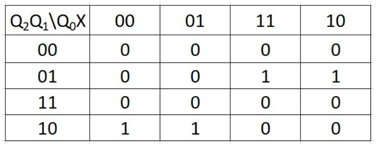
$Q_2Q_1\backslash Q_0X$	00	01	11	10
00	0	1	0	0
01	1	1	0	0
11	0	0	0	0
10	1	1	0	0

Figure 1: K-map for D_0



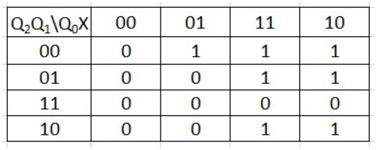
$Q_2Q_1\backslash Q_0X$	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	0	0	0	0
10	0	0	0	0

Figure 2: K-map for D_1



$Q_2Q_1\backslash Q_0X$	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	0	0	0	0
10	1	1	0	0

Figure 3: K-map for D_2



$Q_2Q_1\backslash Q_0X$	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	1

Figure 4: K-map for Y

D Flipflop

Positive edge triggered

```
module D_FF(
    input clk,
    input D,
    output Q,
    input reset);

    reg O;
    assign Q = O;
    always @(posedge clk)
    begin
    if (reset == 1)
    begin
      O = 0;
    end
    else
    begin
        O = D;
    end
    end
endmodule
```

FSM Implementation

Using 3 D Flipflops (code for simulation)

```
module FSM(
   input x,
   input clk,
   input reset,
   output y,
   output clk_out);
   wire D2,D1,D0;
   wire [2:0] Q;

   assign y = (~Q[2] & Q[0])|(~Q[2] & ~Q[1] & x)|(Q[2] & ~Q[1] & Q[0]);
   assign D2 = (~Q[2] & Q[1] & Q[0])|(Q[2] & ~Q[1] & ~Q[0]);
   assign D1 = (~Q[2] & ~Q[1] & Q[0])|(~Q[2] & Q[1] & ~Q[0]);
   assign D0 = (Q[2] & ~Q[1] & ~Q[0])|(~Q[2] & Q[1] & ~Q[0])|(~Q[2] &
       ~Q[0] & x);
   assign clk_out = clk;

   D_FF D_FF2(clk,D2,Q[2],reset);
   D_FF D_FF1(clk,D1,Q[1],reset);
   D_FF D_FF0(clk,D0,Q[0],reset);
endmodule
```

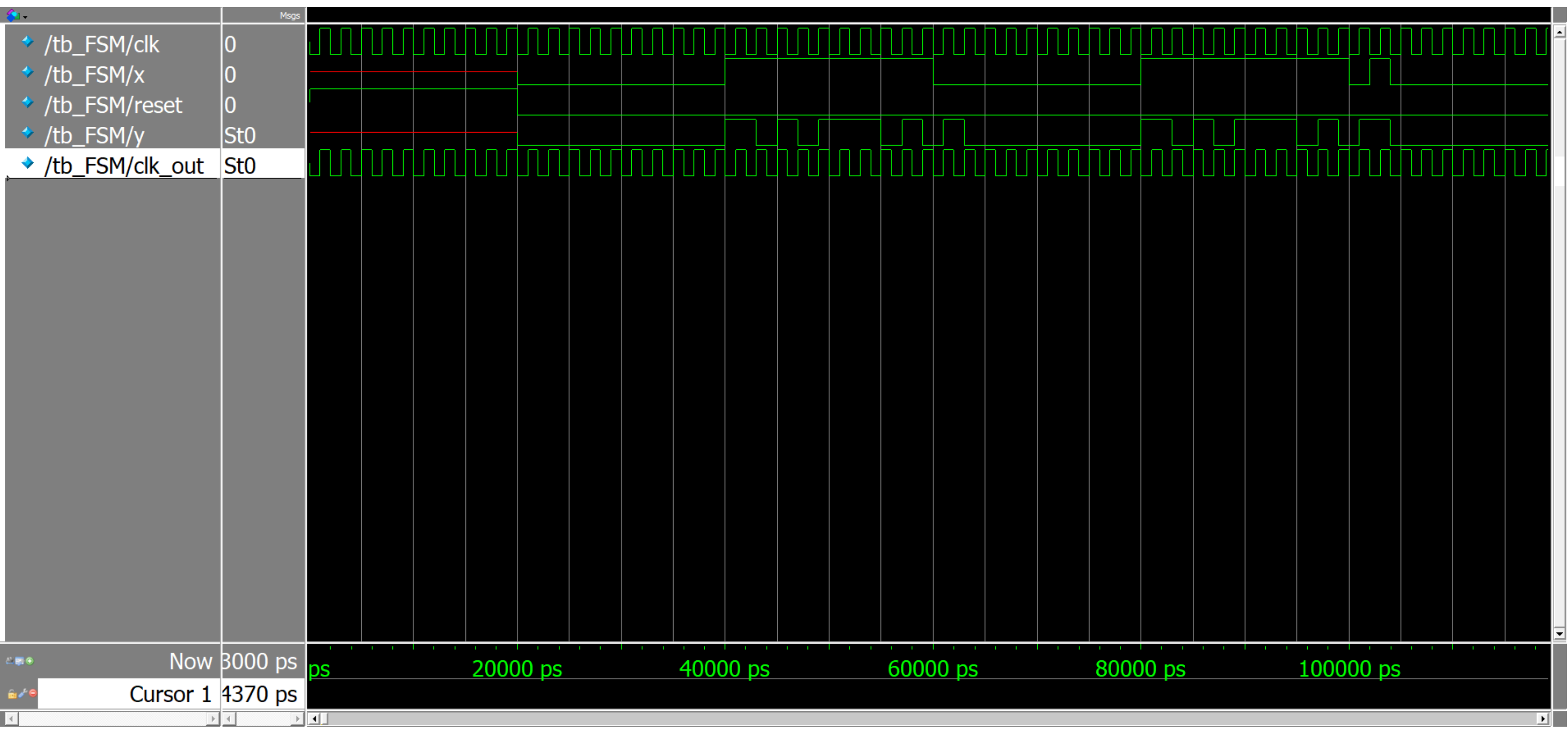
Testbench Code

```
 `timescale 1ns / 1ps
 module tb_FSM();
   reg clk;
   reg x;
   reg reset;
   wire y;
   wire clk_out;

   FSM f(x,clk,reset,y,clk_out);

   initial begin
      clk = 1'b0;
      forever #1 clk = ~clk;
   end

   initial begin
      reset = 1'b1;
      #20
      reset = 1'b0;
      x = 1'b0;
```

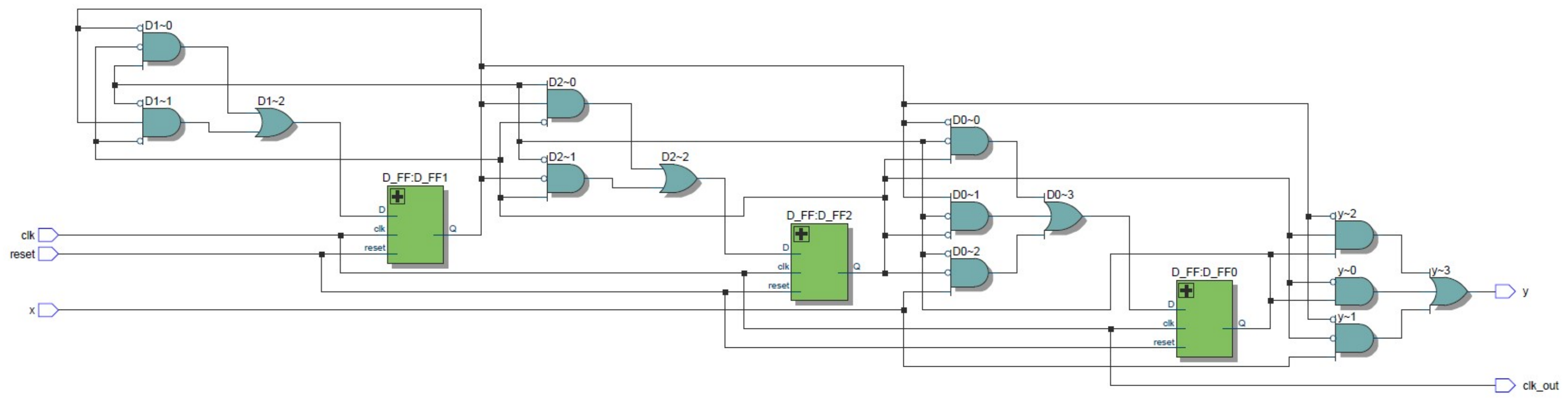
```
        #20
        x = 1'b1;
        #20
        x = 1'b0;
        #20
        x = 1'b1;
        #20
        x = 1'b0;
        #2
        x = 1'b1;
        #2
        x = 1'b0;
        #2
        x = 1'b0;
    end
endmodule
```

Output Waveform

Reset is set to 1 at the beginning to initialize all the variables correctly. Once Reset is set to 0, the sequence 1,1,0,1,0,1 prints as soon it gets serial input as 1 at falling edge of clock. It prints a complete cycle. The next cycle is also printed since x remains 1, and is fully printed despite setting x = 0 mid-sequence. Similarly x is toggled high/low to show that the wave follows necessary requirements.



Circuit Diagram


D1~0
D1~1
D1~2
D2~0
D2~1
D2~2
D_FF:D_FF1
D_FF:D_FF2
D_FF:D_FF0
D0~0
D0~1
D0~2
D0~3
y~2
y~0
y~1
y~3
D
clk
reset
Q
clk
reset
x
y
clk_out