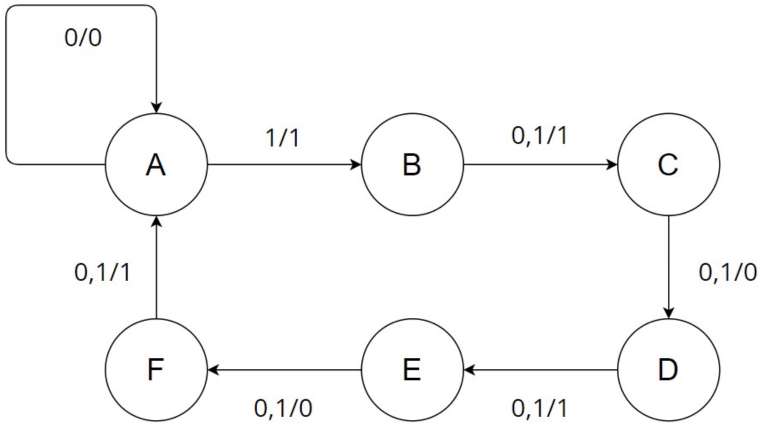
ELL201 Lab Project

Saket Kandoi

2021MT60265

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

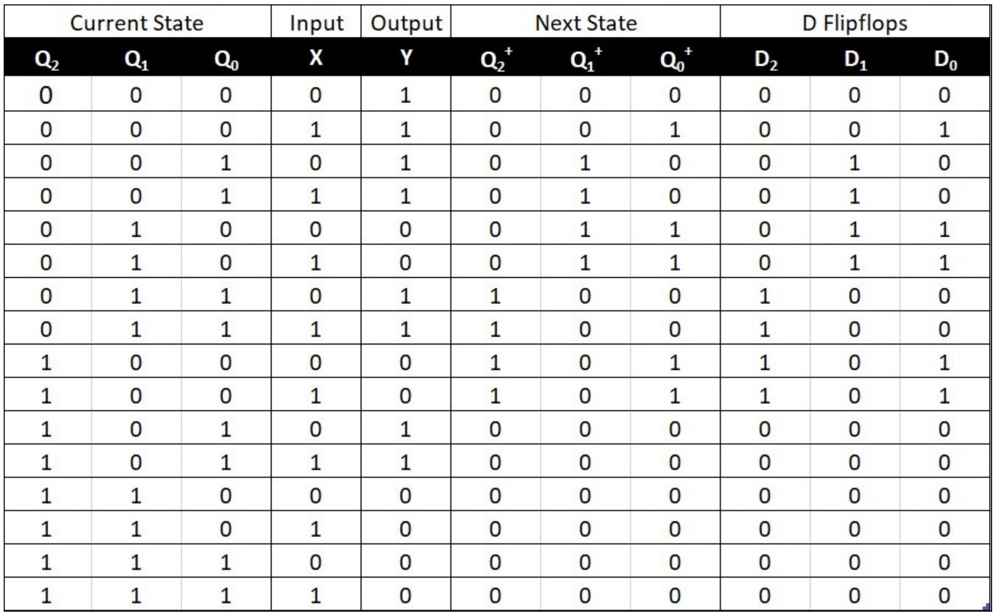
Mealy FSM



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

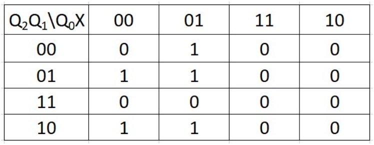


Current State			Input	Output	Next State			D Flipflops		
Q_2	Q_1	Q_0	X	Y	Q_2^+	Q_1^+	Q_0^+	D_2	D_1	D_0
0	0	0	0	1	0	0	0	0	0	0
0	0	0	1	1	0	0	1	0	0	1
0	0	1	0	1	0	1	0	0	1	0
0	0	1	1	1	0	1	0	0	1	0
0	1	0	0	0	0	1	1	0	1	1
0	1	0	1	0	0	1	1	0	1	1
0	1	1	0	1	1	0	0	1	0	0
0	1	1	1	1	1	0	0	1	0	0
1	0	0	0	0	1	0	1	1	0	1
1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

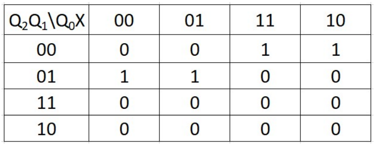
Karnaugh Maps

$$\begin{aligned}
D_0 &= Q_2Q_1'Q_0' + Q_2'Q_1Q_0' + XQ_2'Q_0' \\
D_1 &= Q_2'Q_1'Q_0 + Q_2'Q_1Q_0' \\
D_2 &= Q_2'Q_1Q_0 + Q_2Q_1'Q_0' \\
Y &= Q_2'Q_0 + Q_2'Q_1' + Q_2Q_1Q_0'
\end{aligned}$$



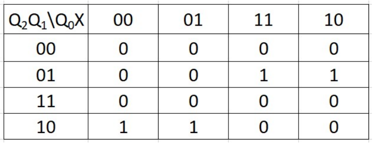
$Q_2Q_1\backslash Q_0X$	00	01	11	10
00	0	1	0	0
01	1	1	0	0
11	0	0	0	0
10	1	1	0	0

Figure 1: K-map for D_0



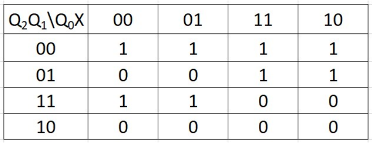
$Q_2Q_1\backslash Q_0X$	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	0	0	0	0
10	0	0	0	0

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	1	1	1	1
01	0	0	1	1
11	1	1	0	0
10	0	0	0	0

Figure 4: K-map for Y

D Flipflop

Positive edge triggered

```
module D_FF(
    input clk,
    input D,
    output Q);
    reg Q;
    always @(posedge clk)
    begin
        Q = D;
    end
endmodule
```

FSM Implementation

Using 3 D Flipflops (code for simulation)

```
module fsm();
    reg clk;
    reg D2,D1,D0,x;
    wire [2:0] Q;
    wire y;

    assign y = (~Q[2] & Q[0])|(~Q[2] & ~Q[1])|(Q[2] & Q[1] & ~Q[0]);
    // Y = Q_2’Q_0 + Q_2’Q_1’ + Q_2Q_1Q_0’

    D_FF D_FF2(clk,D2,Q[2]);
    D_FF D_FF1(clk,D1,Q[1]);
    D_FF D_fF0(clk,D0,Q[0]);
    always @(negedge clk)
    begin
        D2 = (~Q[2] & Q[1] & Q[0])|(Q[2] & ~Q[1] & ~Q[0]);
        D1 = (~Q[2] & ~Q[1] & Q[0])|(~Q[2] & Q[1] & ~Q[0]);
        D0 = (Q[2] & ~Q[1] & ~Q[0])|(~Q[2] & Q[1] & ~Q[0])|(~Q[2] &
             ~Q[0] & x);
    end

    initial begin
        $dumpfile("fsm.vcd");
        $dumpvars(0,fsm);
        $monitor($time," %b Input = %b Output = %b State = %b %b
             %b",clk,x,y,Q[2],Q[1],Q[0]);
        x = 0;
        D2 = 0;
        D1 = 0;
        D0 = 0;
        clk = 1;

        #7
        x = 1;

        #8
        x = 0;

        #20
        x = 1;

        #8
        x = 0;
        #20

        $finish;
    end

    always #2 clk = ~clk;
endmodule
```

Output on Terminal

Output Waveform