

ELP201 Lab Report 3

Adit Malhotra

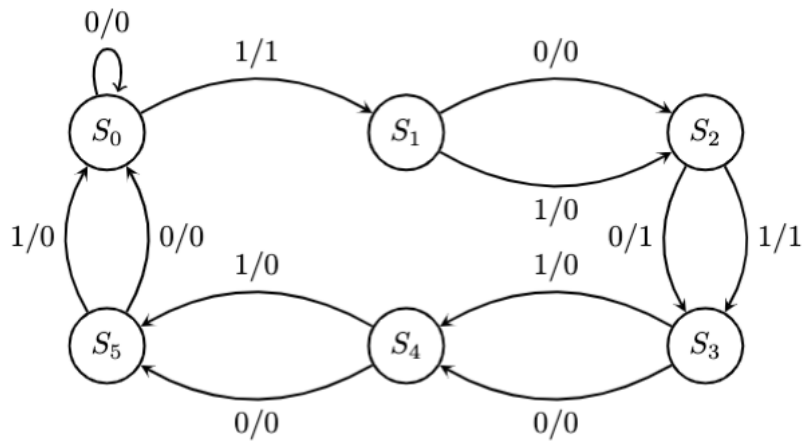
2020EE10458

1 Introduction

According to my entry number (2020EE10458), the sequence to be generated at output side is 1, 0, 1, 0, 0, 0.

2 State diagram

In total there would be 6 states, let them S_i , where $i \in \{0, 1, 2, 3, 4, 5\}$. Let S_0 be the idle state. Then according to the statement of question the state diagram would be:



Since total number of states are 6 and $6 \leq 2^3$, hence total of 3 flipflops would be needed. Let's assign each state S_i the binary value of i .

3 State table

The outputs $Q_2Q_1Q_0$, of the D-flipflops would represent the state at any time t . And since the characteristic equation for the D-flipflop is $Q(t+1) = D$, then the state table would be:

$Q_2(t)$	$Q_1(t)$	$Q_0(t)$	$X(in)$	$Q_2(t+1)$	$Q_1(t+1)$	$Q_0(t+1)$	$Y(out)$	D_2	D_1	D_0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	1	0	0	1
0	0	1	0	0	1	0	0	0	1	0
0	0	1	1	0	1	0	0	0	1	0
0	1	0	0	0	1	1	1	0	1	1
0	1	0	1	0	1	1	1	0	1	1
0	1	1	0	1	0	0	0	1	0	0
0	1	1	1	1	0	0	0	1	0	0
1	0	0	0	1	0	1	0	1	0	1
1	0	0	1	1	0	1	0	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

4 Simplification using K-maps

From the state table, the k-maps for *Output* (Y), D_2 , D_1 , D_0 are as follows:

A 4x4 Karnaugh map for Output (Y). The vertical axis is labeled Q_2Q_1 with values 00, 01, 11, 10. The horizontal axis is labeled Q_0X with values 00, 01, 11, 10. The map contains the following values: (00,00)=0, (00,01)=1, (00,11)=0, (00,10)=0; (01,00)=1, (01,01)=1, (01,11)=X, (01,10)=0; (11,00)=X, (11,01)=X, (11,11)=X, (11,10)=X; (10,00)=0, (10,01)=0, (10,11)=0, (10,10)=0. The cell (00,01) is highlighted in pink, and the cells (01,00) and (01,01) are highlighted in blue.

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

Figure 1: K-Map for *Output* (Y)

A 4x4 Karnaugh map for D_2 . The vertical axis is labeled Q_2Q_1 with values 00, 01, 11, 10. The horizontal axis is labeled Q_0X with values 00, 01, 11, 10. The map contains the following values: (00,00)=0, (00,01)=0, (00,11)=0, (00,10)=0; (01,00)=0, (01,01)=0, (01,11)=1, (01,10)=1; (11,00)=X, (11,01)=X, (11,11)=X, (11,10)=X; (10,00)=1, (10,01)=1, (10,11)=0, (10,10)=0. The cells (01,11) and (01,10) are highlighted in pink, and the cells (10,00) and (10,01) are highlighted in yellow.

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

Figure 2: K-Map for D_2

A 4x4 Karnaugh map for D_1 . The vertical axis is labeled Q_2Q_1 with values 00, 01, 11, 10. The horizontal axis is labeled Q_0X with values 00, 01, 11, 10. The map contains the following values: (00,00)=0, (00,01)=0, (00,11)=1, (00,10)=1; (01,00)=1, (01,01)=1, (01,11)=0, (01,10)=0; (11,00)=X, (11,01)=X, (11,11)=X, (11,10)=X; (10,00)=0, (10,01)=0, (10,11)=0, (10,10)=0. The cells (00,11) and (00,10) are highlighted in green, and the cells (01,00) and (01,01) are highlighted in orange.

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

Figure 3: K-Map for D_1

A 4x4 Karnaugh map for D_0 . The vertical axis is labeled Q_2Q_1 with values 00, 01, 11, 10. The horizontal axis is labeled Q_0X with values 00, 01, 11, 10. The map contains the following values: (00,00)=0, (00,01)=1, (00,11)=0, (00,10)=0; (01,00)=1, (01,01)=1, (01,11)=0, (01,10)=0; (11,00)=X, (11,01)=X, (11,11)=X, (11,10)=X; (10,00)=1, (10,01)=1, (10,11)=0, (10,10)=0. The cells (00,01) and (01,00) are highlighted in pink, and the cells (10,00) and (10,01) are highlighted in blue.

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

Figure 4: K-Map for D_0

And from the k-maps the simplified expressions are as follows:

$$Y = \bar{Q}_2\bar{Q}_0X + Q_1\bar{Q}_0$$

$$D_2 = Q_2\bar{Q}_0 + Q_0Q_1$$

$$D_1 = \bar{Q}_2\bar{Q}_1Q_0 + Q_1\bar{Q}_0$$

$$D_0 = \bar{Q}_0X + Q_1\bar{Q}_0 + Q_2\bar{Q}_0$$

5 Verilog Simulation

5.1 Code

The code file (design.v) for the implementation of the above finite state machine using D-flipflops is: The D-flipflop designed is a postive edge triggered one and has a reset pin which is active low.

```
1 module DFlipFlop(D,clock,preset,Q,Qbar);
2 input D,clock,preset;
3 output reg Q,Qbar;
4
5 initial
6 begin
7     if(~preset)
8     begin
9         Q=1;Qbar=0;
10    end
11    else
12    begin
13        Q=0;Qbar=1;
14    end
15 end
16
17 always@(posedge clock)
18 begin
19     Q <= D;
20     Qbar <= ~D;
21 end
22 endmodule
23
24 module FSM(in,out,clock,q2,q1,q0,q2bar,q1bar,q0bar);
25 input in,clock;
26 output wire q2,q1,q0,q2bar,q1bar,q0bar;
27 output reg out=0;
28
29 DFlipFlop ff1((q0bar & in)|(q1 & q0bar)|(q2 & q0bar),clock,1'b1,q0,q0bar);
30 DFlipFlop ff2((q2bar & q1bar & q0)|(q1 & q0bar),clock,1'b1,q1,q1bar);
31 DFlipFlop ff3((q2 & q0bar)|(q1 & q0),clock,1'b1,q2,q2bar);
32 always @(posedge clock )
33 begin
34     assign out=(q1 & q0bar)|(q2bar & q0bar & in);
35 end
36 endmodule
```

The code file for the testbench (main.v) is: Intitally the input (X) is kept 1 for atleast 6 clock cycles and when it reaches the idle state again, it is then made 0 for again atleast 6 clock cycles, and finally it is again made 1 for one clock cycle so that machine again becomes active then immediately made 0 for again atleast 6 clock cycles. The D-flipflop designed is a postive edge triggered one and has a reset pin which is active low.

```
1 `timescale 1s/100ms
2 `include "design.v"
3
4 module main();
5
6 reg clk,in;
7 wire q0,q1,q2,q2bar,q1bar,q0bar,out;
```

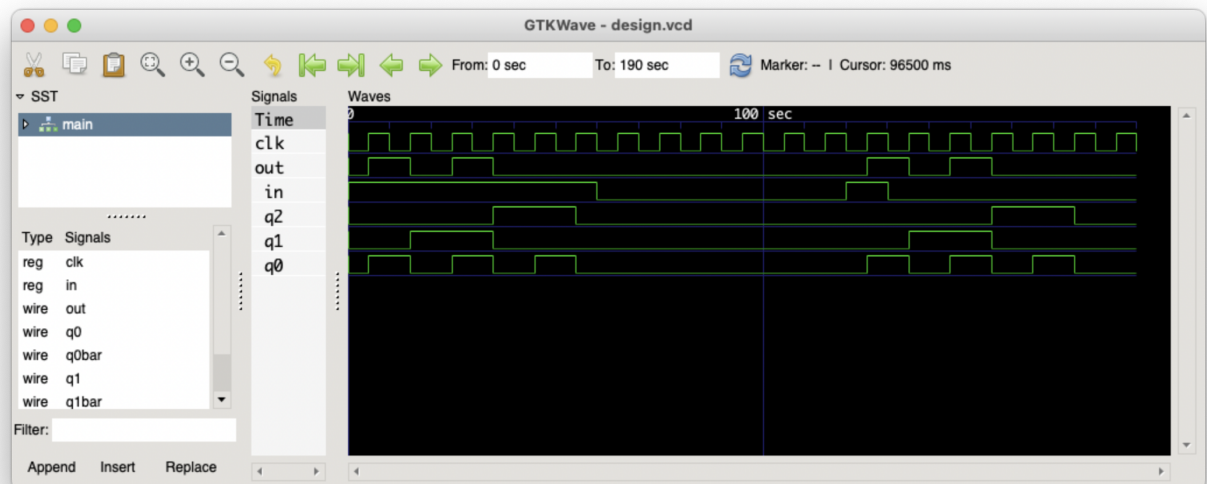
```

8
9 //Instantating the finite state machine
10 FSM func(in,out,clk,q2,q1,q0,q2bar,q1bar,q0bar);
11
12 initial
13 begin
14 $monitor("CLK=%b,out=%b,in=%b,q2=%b,q1=%b,q0=%b",clk,out,in,q2,q1,q0);
15 $dumpfile("design.vcd");
16 $dumpvars(0,main);
17   clk=0; //initially clock=0;
18   in=1;#60 //initially input=1;
19   in=0;#60; //after running the simulation for 60s, input is made 0;
20   in=1;#10;in=0;
21   #60;
22   $finish;
23 end
24 always
25   #5 clk = ~clk; //changing clock after every 5s
26 endmodule

```

5.2 Waveform

The gtkwave plot obtained is:



Hence it can be observed from the plot, that the FSM once starts generating the sequence, keep generating it no matter whatever is the value of X is and also it remains in the idle state as long as X is kept 0. And at last the sequence generated is 1,0,1,0,0,0 at the output side. Hence, the FSM is working perfectly as required.

Download link for the code files

The code files can be downloaded from [here](#).