Harshit Goyal
2021MT10143

Sequence Generator

States:

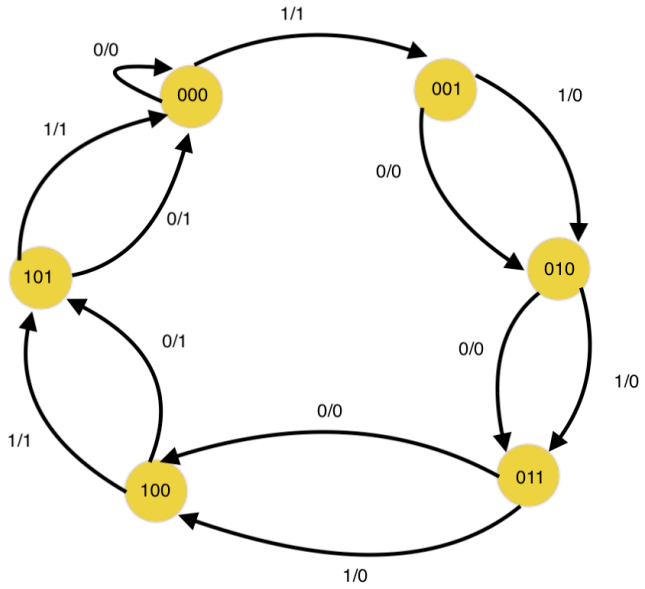
$(4)_{10} = (100)_{(2)}$

$(3)_{10} = (011)_{(2)}$

The required sequence is (for the correct input)

*** idle → 1 → 0 → 0 → 0 → 0 → 1 → 1 → idle**

Mealy Finite State Machine: State Diagram



- Here (000) is the 'idle' state.
- 'idle' transitions to 'idle' giving Output 0 if input is 0 else it transitions to (001) giving Output (1) {First bit of sequence 100011}
- In from states (001) to (101), the output is 'independent' of the 'input' as the machine is 'working' and it keeps generating the sequence until it finishes as specified in the problem.
- We have 6 states and need $[\log_2 6] + 1 = 3$ D flip flops.
- Basically all states 000, 001, 010, 011, 100, 101 can be represented as 3-bit numbers

Excitation Table for D-Flip Flop

Q^n	Q^{n+1}	D
0	**0**	**0**
0	**1**	**1**
1	**0**	**0**
1	**1**	**1**

So for D-Flip Flop, D = Q^{n+1}

- State Transition Table for FSM

Q_2^n	Q_1^n	Q_0^n	Input (I)	$Q_2^{n+1} = D_2$	$Q_1^{n+1} = D_1$	$Q_0^{n+1} = D_0$	Output (Q)
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	1
0	0	1	0	0	1	0	0
0	0	1	1	0	1	0	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	0
0	1	1	0	1	0	0	0
0	1	1	1	1	0	0	0
1	0	0	0	1	0	1	1
1	0	0	1	1	0	1	1
1	0	1	0	0	0	0	1
1	0	1	1	0	0	0	1
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

- K-maps

* K-map for D_2

$Q_2Q_1\backslash Q_0$	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	x	x	x	x
10	1	1	0	0

* K-map for D_1

$Q_2Q_1\backslash Q_0$	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	x	x	x	x
10	0	0	0	0

* K-map for D_0

$Q_2Q_1\backslash Q_0$	00	01	11	10
00	0	1	0	0
01	1	1	0	0
11	x	x	x	x
10	1	1	0	0

* K-map for Output (O)

$Q_2Q_1\backslash Q_0$	00	01	11	10
00	0	1	0	0
01	0	0	0	0
11	x	x	x	x
10	1	1	1	1

- Reduced Expressions:

$$D_2 = Q_1\ Q_0 + Q_2\ Q_0'$$

$$D_1 = Q_1\ Q_0' + Q_2'Q_1'Q_0$$

$$D_0 = Q_0'I + Q_1Q_0' + Q_2Q_0'$$

$$D_0 = Q_2 + Q_1'Q_0'I$$

Verilog Code

* **Here I = Input received**
* **O = Output (bits of generated sequence)**
* **clk = Clock**

* Positive Edge Triggered D-Flip Flop

```
module D_flip_flop(D, Q, clk, reset);
    input D, clk, reset ;
    output reg Q;
    always @(posedge clk)
        begin
            if (reset)
                Q <= 1'b0;
            else
                Q <= D;
        end
endmodule
```

* Sequence Generator

```
   module sequence_gen(O,I, clk, reset);
      input I, clk, reset;
      output O;
      wire q2, q1, q0;
      wire d2, d1, d0;

      assign d2 = (q1 && q0) || (q2 && ~q0);
      assign d1 = (q1 && ~q0) || (~q2 && ~q1 && q0);
      assign d0 = (~q0 && I) || (q1 && ~q0) || (q2 && ~q0);
      assign O  = q2 || (~q1 && ~q0 && I);

      D_flip_flop dff2(d2, q2, clk, reset);
      D_flip_flop dff1(d1, q1, clk, reset);
      D_flip_flop dff0(d0, q0, clk, reset);
   endmodule
```

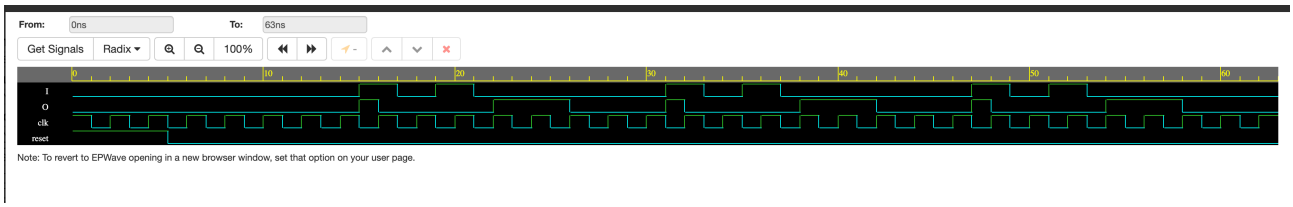
* Test Bench

```
module test_bench();
  reg clk, reset, I;
  wire O;
  sequence_gen inst(O, I, clk, reset);
  always #1 clk = ~clk;
  always @(posedge clk)
      $display("Input I: %b Output O %b ", I,O);
  initial begin
      $dumpfile("graph.vcd");
      $dumpvars;
      clk = 1;
      reset = 1;
      I = 0;
      #5 reset = 0;

      repeat (3) begin
          #10 I = 1;
          #2 I = 0;
          #2 I = 1;
          #2 I = 0;
      end
      #10 $finish;
   end
endmodule
```

*** In the TestBench, I have set I high for 1 cycle duration then set it to low. The machine starts giving the output (as seen on the waveform). Then during it is working, I is again set high for 1 clock duration and machine keeps producing the desired output without getting disturbed.**

* Waveforms



Terminal

```
VCD info: dumpfile graph.vcd opened for output.
Input I: 0 Output 0 x
Input I: 0 Output 0 0
Input I: 0 Output 0 0
Input I: 0 Output 0 0
Input I: 0 Output 0 0
Input I: 0 Output 0 0
Input I: 0 Output 0 0
Input I: 0 Output 0 0
Input I: 1 Output 0 1
Input I: 0 Output 0 0
Input I: 1 Output 0 0
Input I: 0 Output 0 0
Input I: 0 Output 0 1
Input I: 0 Output 0 1
Input I: 0 Output 0 0
Input I: 0 Output 0 0
Input I: 1 Output 0 1
Input I: 0 Output 0 0
Input I: 1 Output 0 0
Input I: 0 Output 0 0
Input I: 0 Output 0 1
Input I: 0 Output 0 1
Input I: 0 Output 0 0
Input I: 0 Output 0 0
Input I: 1 Output 0 1
Input I: 0 Output 0 0
Input I: 1 Output 0 0
Input I: 0 Output 0 0
Input I: 0 Output 0 1
Input I: 0 Output 0 1
Input I: 0 Output 0 0
Input I: 0 Output 0 0
main.v:55: $finish called at 63 (1s)
```

***Here the Ouput is shown against the Input at each positive edge of the clock. We can see 3 repetitions of the sequence 100011.**