FSM Lock Counter

Lab Report # 07

Spring 2020

CSE 308-L—Digital Systems Design--Lab

Submitted by: **Muhammad Asif Ayub**

Registration No: **17PWCSE1508**

Class Section: **A**

"On my honor, as student of Engineering and Technology, I have neither given nor received unauthorized assistance of this academic work".

Submitted to:

Engr. Madiha Sher

Thursday, July 9, 2020th

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

FSM Lock Counter

Question 1:-

Build an electronic combination lock with reset button, two number buttons
(0 and 1), and an unlock output. The combination should be "01011".

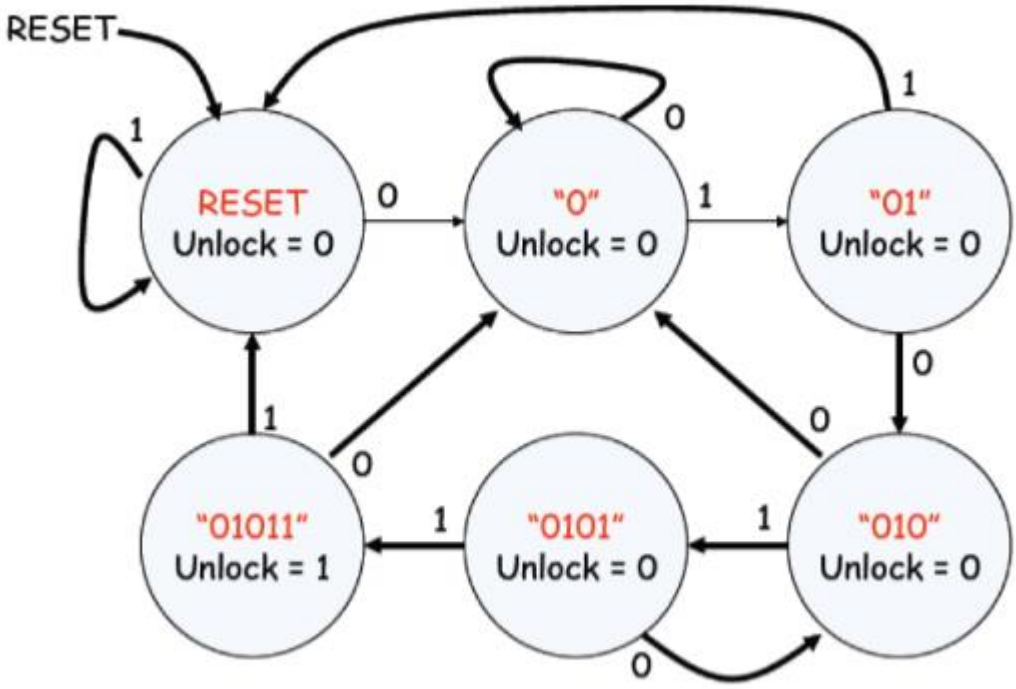

Fig2: State Transition Diagram:

Main Module:-

```
 1 module FSM_UNLOCK_COUNTER (clk,reset,data0,data1,dataout);
 2
 3 input clk;
 4 input reset,data0,data1;
 5 reg [2:0]state;
 6 reg [2:0]next_state;
 7 output dataout;
 8
 9 parameter RESET = 0, state_0 = 1,
10           state_01 = 2, state_010 = 3,
11           state_0101 = 4, state_01011 = 5;
12
13 initial
14 begin
15         state <= RESET;
16         next_state <= RESET;
17 end
18
19 always @(posedge clk)
20 begin
21         state <= next_state;
22 end
23
24 always @(posedge clk or state or reset or data0 or data1)
25 begin
26         if(reset)
27                 next_state <= RESET;
28         else
29                 case(state)

30
31                  RESET:      next_state <= data0 ? state_0 : data1 ? RESET : next_state;
32                  state_0:    next_state <= data0 ? state_0 : data1 ? state_01 : next_state;
33                  state_01:   next_state <= data0 ? state_010 : data1 ? RESET : next_state;
34                  state_010:  next_state <= data0 ? state_0 : data1 ? state_0101 : next_state;
35                  state_0101: next_state <= data0 ? state_010 : data1 ? state_01011 : next_state;
36                  state_01011: next_state <= data0 ? state_0 : data1 ? RESET : next_state;
37                  default:
38                         next_state <= RESET;
39             endcase
40 end
41 assign dataout = (next_state == state_01011);
42 endmodule
```

Top Module:-

```
 1 module FSM_TOP;
 2
 3 reg clk, reset,data0,data1;
 4 wire dataout;
 5
 6 FSM_UNLOCK_COUNTER DUT(clk,reset,data0,data1,dataout);
 7
 8 initial
 9 begin
10            clk=0;
11            reset=1;
12            data0 = 1;
13            data1 = 0;
14         #5 reset=0;
15         #10 data0 = 0;
16             data1 = 1;
17         #20 data0 = 1;
18             data1 = 0;
19         #30 data0 = 0;
20             data1 = 1;
21         #200 $finish;
22 end
23 always
24 #10 clk = ~clk;
25
26 initial
27 $monitor($time,":\t data0 = %d \t data1 = %d \t dataout=%d", data0, data1, dataout);
28
29 endmodule
```

```
# Compile of mainmodule.v was successful.
# Compile of topmodule.v was successful.
# 2 compiles, 0 failed with no errors.
vsim work.FSM_TOP
# vsim work.FSM_TOP
# Loading work.FSM_TOP
# Loading work.FSM_UNLOCK_COUNTER
run
#                0:  data0 = 1   data1 = 0   dataout=0
#               15:  data0 = 0   data1 = 1   dataout=0
#               35:  data0 = 1   data1 = 0   dataout=0
#               65:  data0 = 0   data1 = 1   dataout=0
#               70:  data0 = 0   data1 = 1   dataout=1
#               90:  data0 = 0   data1 = 1   dataout=0
# Compile of mainmodule.v was successful.
# Compile of topmodule.v was successful.
# 2 compiles, 0 failed with no errors.
quit -sim
```