LOG BOOK [Week 8]
Name: TABINA KAMAL
Matric No.: 208651
Assigned Date: 16 December 2021

LAB 6: MULTIPLEXERS AND STATE MACHINE

1.0 Objectives

- Designing, implementing and testing a three-bit wide 5-to-1 multiplexer and a finite state machine for a sequence counter using Verilog entry.

2.0 Results and Simulation

A. Multiplexer

```
module fivemux (S, U, V, W, X, Y, M);

input [2:0] S;
input U, V, W, X, Y;

reg M;
output M;

//output [2:0] M;

//reg [2:0] M;

   always@(U or V or W or X or Y or S or M)

      if (S == 3'b000)
         M = U;

      else if (S == 3'b001)
         M = V;

      else if (S == 3'b010)
         M = W;

      else if (S == 3'b011)
         M = X;

      else if (S == 3'b100)
         M = Y;

      else M = 0;

endmodule
```

Figure 1: 5-to-1 Multiplexer Code

Table 1: Truth table obtained from 5-to-1 multiplexer

SELECT LINES			INPUTS					OUTPUT
S2 [SW2]	S1 [SW1]	S0 [SW0]	U [SW4]	V [SW5]	W [SW6]	X [SW7]	Y [SW8]	M [LED0]
0	0	0	1	0	0	0	0	1
0	0	1	0	1	0	0	0	1
0	1	0	0	0	1	0	0	1
0	1	1	0	0	0	1	0	1
1	0	0	0	0	0	0	1	1

	tatu	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
1	✔		in S[1]	Location	PIN_AC12	Yes			
2	✔		in S[2]	Location	PIN_AF9	Yes			
3	✔		out M	Location	PIN_V16	Yes			
4	✔		in U	Location	PIN_AD11	Yes			
5	✔		in V	Location	PIN_AD12	Yes			
6	✔		in W	Location	PIN_AE11	Yes			
7	✔		in X	Location	PIN_AC9	Yes			
8	✔		in Y	Location	PIN_AD10	Yes			
9	✔		in S[0]	Location	PIN_AB12	Yes			

Figure 2: 5-to-1 Multiplexer Pin Assignments

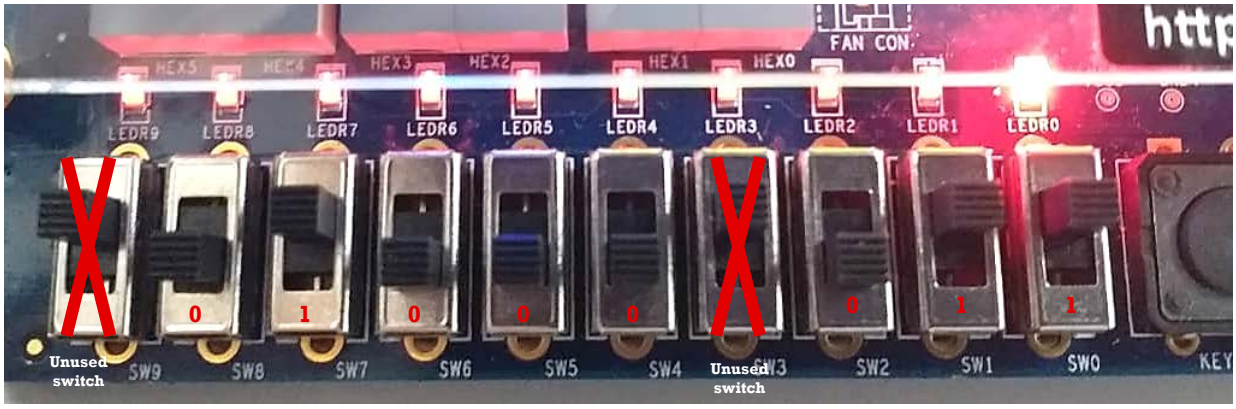

Figure 3: 5-to-1 Multiplexer in Operation on Board

B. Finite State Machine

```
module fsm_counter (clk, reset, outp);

   input clk, reset;
   output [2:0] outp; //reg [2:0] outp;

   wire [2:0] outp;

   reg [2:0] c_state;

   //define all possible state
   parameter s1 = 3'b000; parameter s2 = 3'b110;
   parameter s3 = 3'b111; parameter s4 = 3'b011;
   parameter s5 = 3'b010; parameter s6 = 3'b101;

   //your code here

   initial
      begin
         c_state = s1;
      end

   always@(posedge clk, posedge reset)

      begin
          if(reset)    //Set Counter to Zero
            c_state <= s1;
          else if(c_state == 3'b000)
            c_state = s2;
          else if(c_state == 3'b110)
            c_state <= s3;
          else if(c_state == 3'b111)
            c_state <= s4;
          else if(c_state == 3'b011)
            c_state <= s5;
          else if(c_state == 3'b010)
            c_state <= s6;
          else
            c_state <= 3'b000;
        end

 assign outp = c_state;


endmodule
```

Figure 4: Finite State Machine Counter Code

	tatu	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
1	✔		in S[1]	Location	PIN_AC12	Yes			
2	✔		in S[2]	Location	PIN_AF9	Yes			
3	✔		out M	Location	PIN_V16	Yes			
4	✔		in U	Location	PIN_AD11	Yes			
5	✔		in V	Location	PIN_AD12	Yes			
6	✔		in W	Location	PIN_AE11	Yes			
7	✔		in X	Location	PIN_AC9	Yes			
8	✔		in Y	Location	PIN_AD10	Yes			
9	✔		in S[0]	Location	PIN_AB12	Yes			

Figure 5: Finite State Machine Counter Pin Assignment

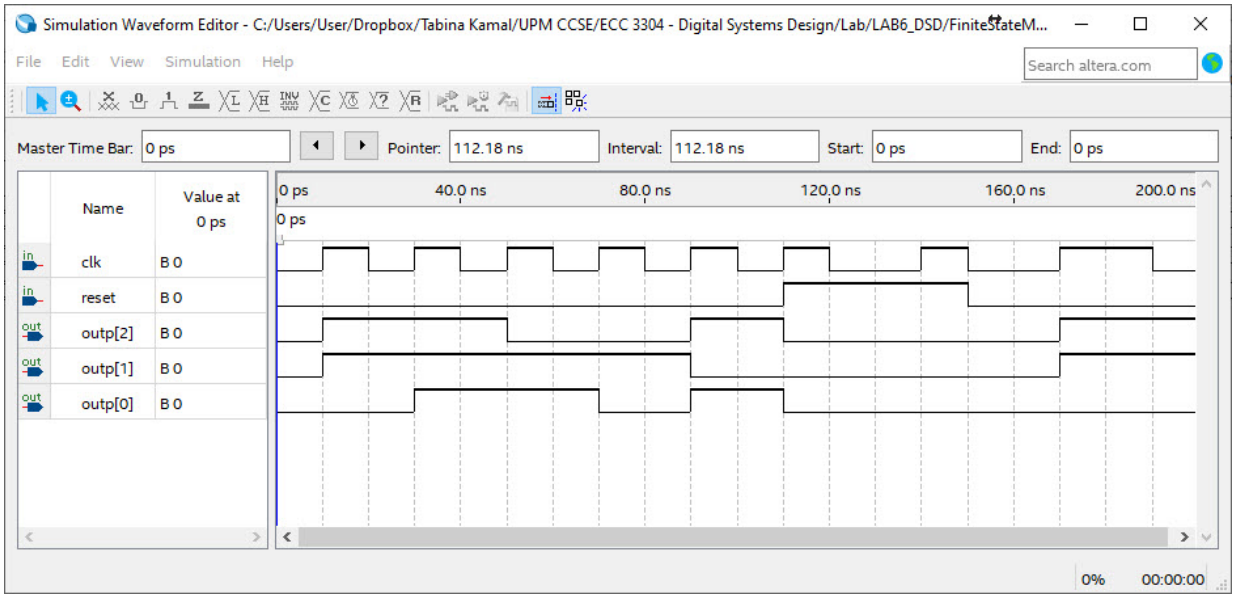

Figure 6: Finite State Machine Counter Functional Simulation

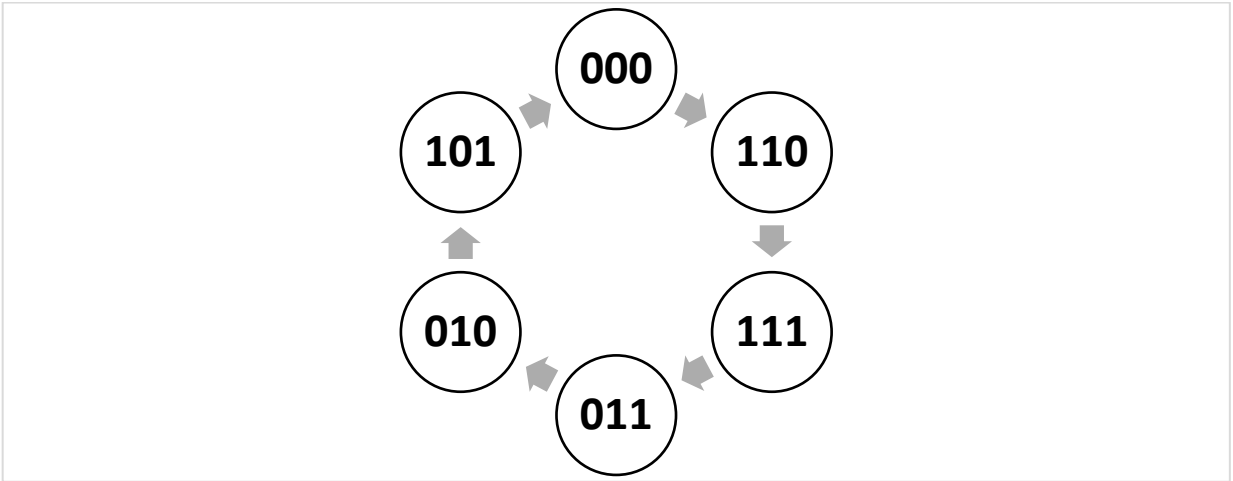

Figure 7: State Diagram

3.0 Conclusion

A. Multiplexer

Successfully compiled and demonstrated.

B. Finite State Machine

Successfully compiled and simulated.
Not yet implemented on the board.