# Lab 12 Implement a Traffic Light controller using FSM.



Spring 2025

Submitted by: Mohsin Sajjad

Registration No: 22pwsce2149

Class Section: A

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

Mohan Sayad

Student Signature:

Submitted to:

Engr. Faheem Jan

Month Day, Year (28 05, 2025)

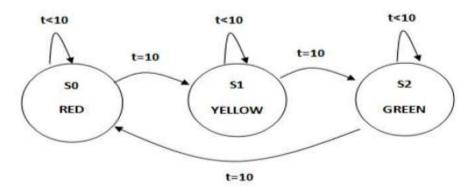
Department of Computer Systems Engineering
University of Engineering and Technology, Peshawar

## Implement a Traffic Light controller using FSM.

### **Objective:**

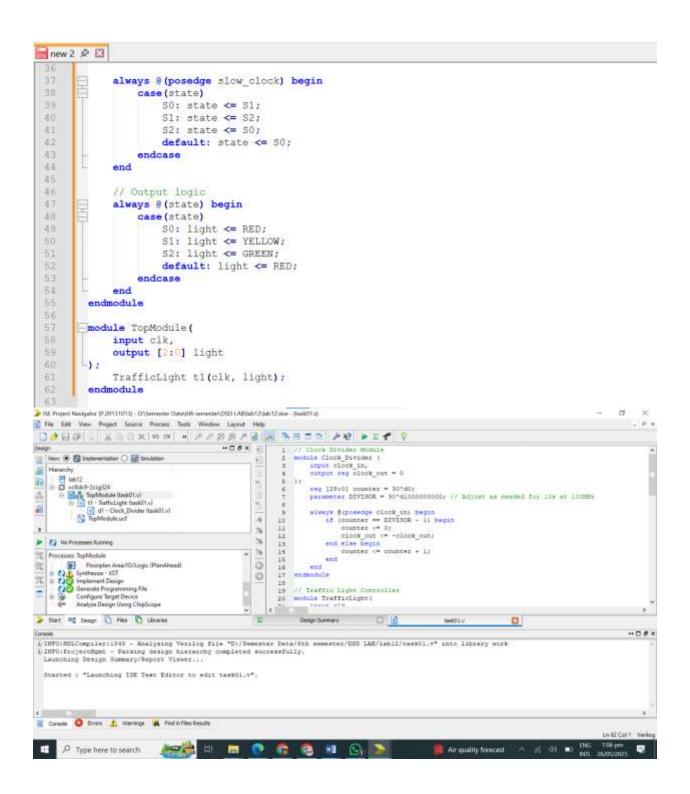
The lab this week will continue the introduction to FSMs with a simple Traffic Light Controller design project. In the lab, you are required to create a state diagram of a Mealy machine, which implements a traffic light controller, based on provided guidelines. You will then use this state diagram to write the behavioral Verilog description of the traffic light controller. The testing of your traffic light controller will take place during the lab session using the ISE development tools and the Spartan 6 board. As with the previous lab, we will make use of the character LEDs to display the outputs of our digital circuit.

Figure 1. Task1:

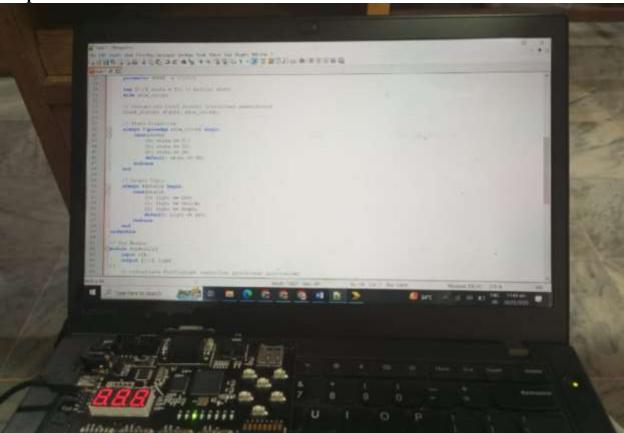


#### **CODE:**

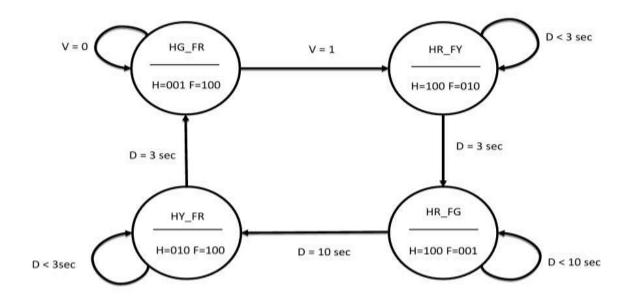
```
new 2 A E3
        module Clock Divider (
             output reg clock out = U
       137
             reg [29:0] counter = 307/00:
             parameter DIVISOR = 30'd100000000B; // Adjust as needed for 10s at 100MHz
             always 8 (posedge clock in) begin
                  if (counter == DIVISOR - 1) begin
counter <= 0;
                      clock out <= -clock out;
日日 日日 日日 日日日日
                  end else begin
                      counter <= counter + 1:
         endmodule
       module TrafficLight (
             output reg [2:0] light
             // State encoding
             parameter S0 = 2'b00;
parameter S1 = 2'b00;
             // Light signals
             parameter RED = 3'H100;
parameter YELLOW = 3'H005;
             parameter GREEN = 1'h010;
             reg [1:0] state = S0: // Initial state
             Clock_Divider dl (clk, slow_clock);
```



**Output:** 



**Task 02:** 



#### **CODE:**

```
new 2 / 🖸
        | module Clock Divider (
                input clock_in,
                output reg clock out = 0
         13
                reg [20:0] counter = 50'd0;
parameter DIVISOR = 30'd100000000; // 1 second at 100 MHz
                always 0 (posedge clock in) begin
                     if (counter == DIVISOR - 1) begin
                           counter <= 0;
 拉克拉拉拉拉拉
                           clock out <= -clock out;
                      end else begin
                           counter <- counter + 1;
                     end
                end
          endmodule
 10
         module Traffic Controller (
 19
20
21
22
23
24
                input clk,
                input reset,
                input V,
                output reg [2:0] Hout,
                output reg [2:0] Fout
25
26
27
28
                // State encoding
                parameter HG_FR = 2'b00, // Highway Green, Farm Red
HR_FY = 2'b01, // Highway Red, Farm Yellow
HR_FG = 2'b10, // Highway Red, Farm Green
BY_FR = 2'b11; // Highway Yellow, Farm Red
 29
                reg [1:0] state = HG FR;
reg [3:0] counter = \overline{0}; // 4-bit counter is enough for 10 sec
                always 8(posedge clk or posedge reset) begin
if (reset) begin
 34
```

```
mew 2 ≯ 🖾
                      state <= HG FR;
                      counter <= 0;
                  end else begin
                      case (state)
 400
                          HG_FR: begin
                               Hout <= 3'b001; // Green
Fout <= 3'b100; // Red
 41
 42
 43
                               if (V) begin
                                   state <= HR FY;
 22
                                   counter <= 0;
 45
                               end
 44
 47
                          and
 48
 49
                           HR_FY: begin
                               Hout <= 3*b100; // Red
Fout <= 3*b010; // Yellow
                               if (counter < 3)
                                   counter <= counter + 1;
 54
                               else begin
                                   state <= HR FG;
 56
                                   counter <= 0;
                               end
                          end
 60
                           HR FG: begin
 61
                               Hout <= 3°b100; // Red
 62
                               Fout <= 3'h001; // Green
                               if (counter < 10)
                                   counter <= counter + 1;
 64
                               else begin
                                   state <= HY FR;
 67
                                   counter <= 0;
 68
 69
```

```
mew 2 🖈 🖸
                                                                                                                                                         HY FR: begin
                                                                                                                                                                                 Hout <= 3'h010; // Yellow
                                                                                                                                                                                 Fout <= 3'b100; // Red
           74
                                                                                                                                                                                 if (counter < 3)
                                                                                                                                                                                                         counter <= counter + 1;
           76
                                                                                                                                                                                 else begin
                                                                                                                                                                                                         state <= HG FR;
                                                                                                                                                                                                          counter <= 0;
           79
                                                                                                                                                                                 end
                                                                                                                                                         end
                                                                                                                                endcase
                                                                                                       end
          83
          84
                                                       endmodule
         85
         86
                                           module TopModule (
         87
                                                                              input clk,
         88
                                                                              input reset,
          89
                                                                              input V,
          90
                                                                              output [2:0] Hout,
          91
                                                                              output [2:0] Fout
                                                -);
          92
          93
                                                                              wire slow clk;
                                                                              Clock_Divider divider(clk, slow_clk);
          94
                                                                              Traffic Controller controller (slow clk, reset, V, Hout, Fout);
          96
                                                       endmodule
          93
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 CX
 15E Project Navigator (F20111013) - O/Mercenter Data/diff nerrente/D93 LAB/lab12 test02/lab12 test02 size - (bask02/lab
The Edit View Project Source Process Tools Window Layout Help
   Date of the state 
                                                                                                                                                                                                                                BEES PR PER V
                                                                                                                                                                         +08× 4
                                                                                                                                                                                                                                                                                      HR_FG: state ow HZ_FR;
HY_FR: state ow HG_FR;
Hefault: state ow HG_FR;
  West (B) (S) Implementation (C) ((E) Smulation
                                                                                                                                                                                                                              41
                                                                                                                                                                                                                                                        endosse
and
 Jij Herarchy
            施
                                                                                                                                                                                                                              43 44 45 47 48 49 50 51
                                                                                                                                                                                                                                                        // Combinational Output Lagin
slways 8(*) begin
case (state)

By Fk: begin Bout = 3'b001; Fout = 3'b100; end // B-Green, F-Ded
HE FY: begin Bout = 3'b100; Fout = 3'b010; end // B-Ged, F-Tellow
HE FO: begin Bout = 3'b100; Fout = 3'b010; end // B-Ged, F-Tellow
BY Fk: begin Bout = 3'b100; Fout = 3'b001; end // B-Yellow, F-Med
default: begin Bout = 3'b000; Fout = 3'b000; end
                                                                                                                                                                                                         A
                                                                                                                                                                                                          2
                                                                                                                                                                                                         %
  ▶ 🖸 No Processes Running
  Processes: TopModule
Processes Replination

| 10 Fin Planning (Planshead) - Pier Synthesis
| 10 Fin Planning (Planshead) - Pier Synthesis
| 10 Fin Planning (Planshead) - Pier Synthesis
| 10 Fin Planning (Planshead) |
| 10 Fin Planning (Planshead) |
| 10 Fin Planning (Planshead) |
| 10 Fin Planshead |
|
                                                                                                                                                                                                         0
                                                                                                                                                                                                                                           endmodule
                                                                                                                                                                                                         0
                                                                                                                                                                                                                                            // Top-Level Module to Connect Divides and FERS module TopModule (
                                                                                                                                                                                                                                                                                                                                               (7.100) 0004 45144
                                                                                                                                                                                                                                                 Deagn Summery
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             + D # ×
  UTHFO:HHLCompiler:1845 - Analysing Verslog file "D)/Bemester Data/4th semester/DSD LAB/labil_tack02/tack02.v" into library work 
UTHFO:ProjectNgmt - Farming design hierarchy completed successfully. 
Launching Design Summary/Report Viewer...
         Started : "Launching ISE Text Editor to edit tank02.v".
    Consile O Errors 🔝 Warrengs 🚜 Find in Files Results
```

En SE Col 1 Verlag

#### **Output:**

