DIGITAL SYSTEM DESIGN LAB

LAB #10

Spring 2021

CSE308L DSD LAB

Submitted by: **Shah Raza**

Registration No. : **18PWCSE1658**

Class Section: **B**

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

Student Signature: ______________

Submitted to:

Engr. Madiha Sher

Thursday, July 15, 2021

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Task :

Implement a Traffic Light controller using FSM.

Introduction

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.

The Traffic Light Controller

This week you will design a traffic light controller for the highway and farm road intersection shown in Figure 1. For simplicity, we will assume that the traffic lights on opposite ends of the intersection are the same; in other words, if the light for the North bound traffic is green, then the light for the South bound traffic is also green. The traffic light controller outputs two 2-bit signals, highwaySignal and farmSignal, for the highway road and the farm road, respectively.

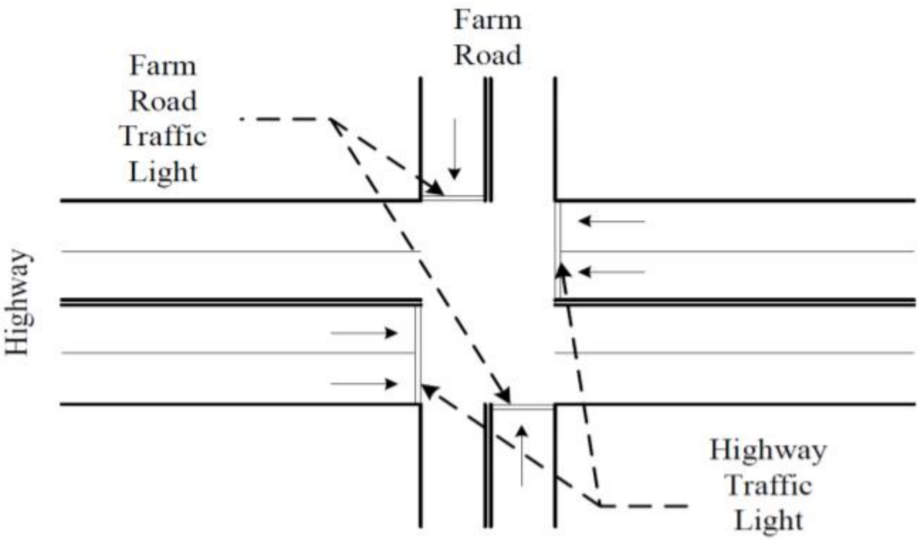

Figure 1: State Transition Diagram

The following encoding is used for both signals:

001 : Green

010 : Yellow

100 : Red

Obviously, to avoid accidents, the highway lights and farm road lights must not be green at the same time. Furthermore, the transitions from green on each road must be followed by 3 seconds of yellow. Because the traffic on the highway is much greater than that of the farm road, the traffic light must remain green on the highway longer than on the farm road. For this design, the highway light must remain green unless there is a vehicle on the farm road, while the farm road light must remain green for only 10 seconds.

The brain of the traffic light controller is the FSM. For the initial design, Vehicle, Clock and Rst constitute the only input to the FSM. The output of the FSM includes the highwaySignal and farmSignal. The operation of the FSM came be described by using a state diagram as shown in figure 2.

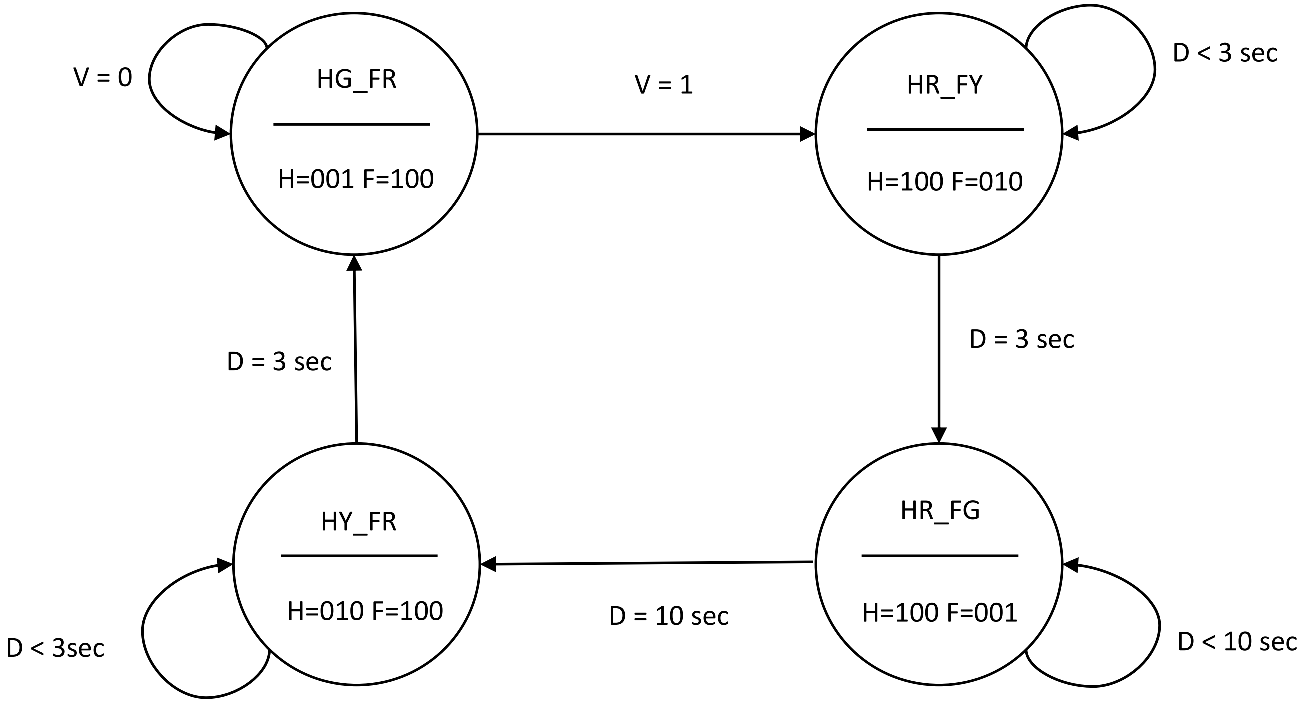

Figure 2: State Transition Diagram

Code:

TrafficLight:

```
module TrafficLight(clk,rst,v,farm,highway);
    input clk,rst,v;
    output reg [2:0] farm, highway;
    reg [1:0] PS,NS;
    parameter [1:0] HG_FR = 0, HR_FY = 1, HR_FG = 2, HY_FR = 3;
    wire oclk;

    Divider d1(clk,oclk,rst);

    always @(posedge oclk)
        if(rst==0)
        begin
            PS = HG_FR;
        end
        else
            PS = NS;
    always @(PS or v or rst)
        case(PS)
            HG_FR:
            begin
                NS = v?HR_FY: HG_FR;
                highway = v?3'b100:3'b001;
                farm = v?3'b010:3'b100;
            end
            HR_FY:
            begin
                NS = HR_FG;
                highway = 3'b100;
                farm = 3'b001;
            end
            HR_FG:
            begin
                NS = HY_FR;
                highway = 3'b010;
                farm = 3'b100;
            end
            HY_FR:
            begin
                NS = HG_FR;
                highway = 3'b001;
                farm = 3'b100;
            end
        endcase
endmodule
```

Divider:

```
module Divider(input iclk,output reg oclk, input rst);
    reg [100:0] count;
    always @(posedge iclk)
        if(rst==0)
        begin
            oclk = 0;
            count = 0;
        end
        else
        begin
            count = count+1;
            if(count==3*100000000)
            begin
                    oclk = ~oclk;
                    count = 0;
            end
        end
endmodule
```

UCF File:

```
net "rst" LOC =m18 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST | PULLUP;
net "v" LOC =f17 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST | PULLUP;
net "clk" LOC =v10 | IOSTANDARD = LVCMOS33 | period = 100MHz;

net "highway[0]" LOC =P15 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
net "highway[1]" LOC =P16 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
net "highway[2]" LOC =N15 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
net "farm[0]" LOC =U18 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
net "farm[1]" LOC =T17 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
net "farm[2]" LOC =T18 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
```