SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
Faculty of Engineering and Technology
Department of Electronics and Communication Engineering

18ECC206J - VLSI Design
VI Semester, **2022-2023 (EVEN** Semester)

Title of Mini Project : TRAFFIC LIGHT CONTROLLER USING VERILOG

Date of Submission : 28/04/2023

Particulars	**Max. Marks**	Marks Obtained	Marks obtained
		Name: R.Koushik	**Name: k.Lokesh**
		Register No: RA2011004010231	**Register No:RA2011004010242**
Design Code	25		
Demo verification &viva	10		
Project Report	05		
Total	**40**		

Staff Name :

Signature:

TRAFFIC LIGHT CONTROLLER USING VERILOG

OBJECTIVE :

To design a Traffic light Controller using Verilog HDL in XILINX and simulate the output in ModelSim to verify the output.

SOFTWARE USED :

Xilinx ISE and ModelSim.

ABSTRACT :

Traffic light control systems are widely used to monitor and control the flow of automobiles through the junction of many roads. They aim to realize smooth motion of cars in the transportation routes. However, the synchronization of multiple traffic light systems at adjacent intersections is a complicated problem given the various parameters involved. The code is programmed in Xilinx, using Verilog HDL.

INTRODUCTION :

The main objective of this project is to design a Verilog module to control the traffic . Through using VHDL language towards the traffic controller that is light design, the traffic light control circuit utilizes signal that is electronic automatic control to understand two sets of lights which are red, Green and yellowish. Those lights command automobiles and pedestrians moving properly at the crossroad, which bases regarding the information of traffic state transition. Most of control systems are designed by advanced PLC technology, that also can effortlessly imitating the traffic that has experience.

The FPGA gets current signals of vehicles moving crossroad and base on those signals send next thing order. Also, into the specific road the traffic light should specifically be set. In addition, the FPGA need to consider the best time, meaning that Isolating the traffic situation by the proper time. The codes must certanly be packed within the FPGA development base on different models, that might increase the program flexibly.

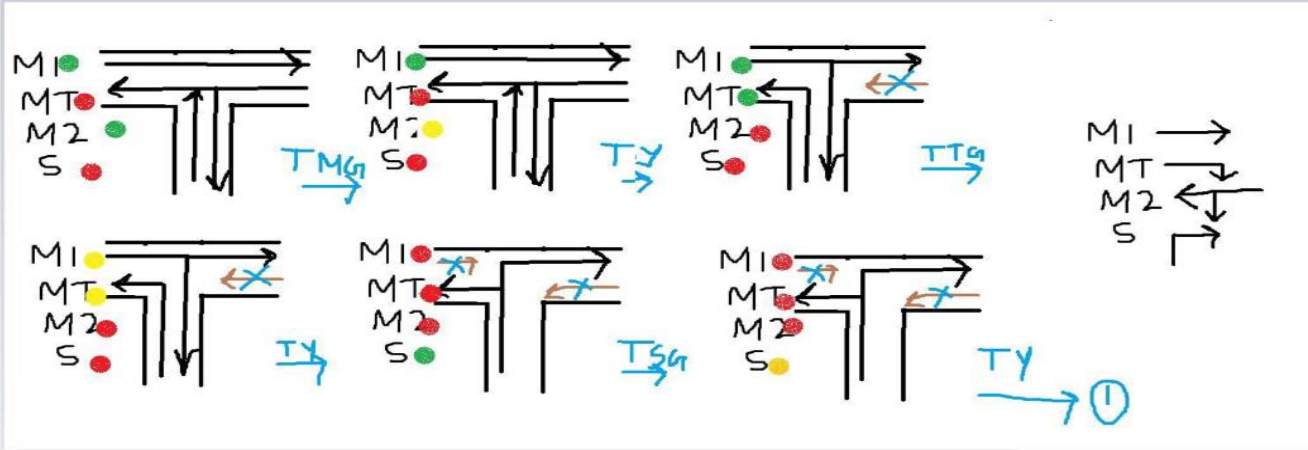
REALISTIC CONSTRAINS :

When we consider real life scenario, there will be a delay in the process of traffic light.

Problem Statement:

The aim of the project is to design a traffic controller for a T-intersection.

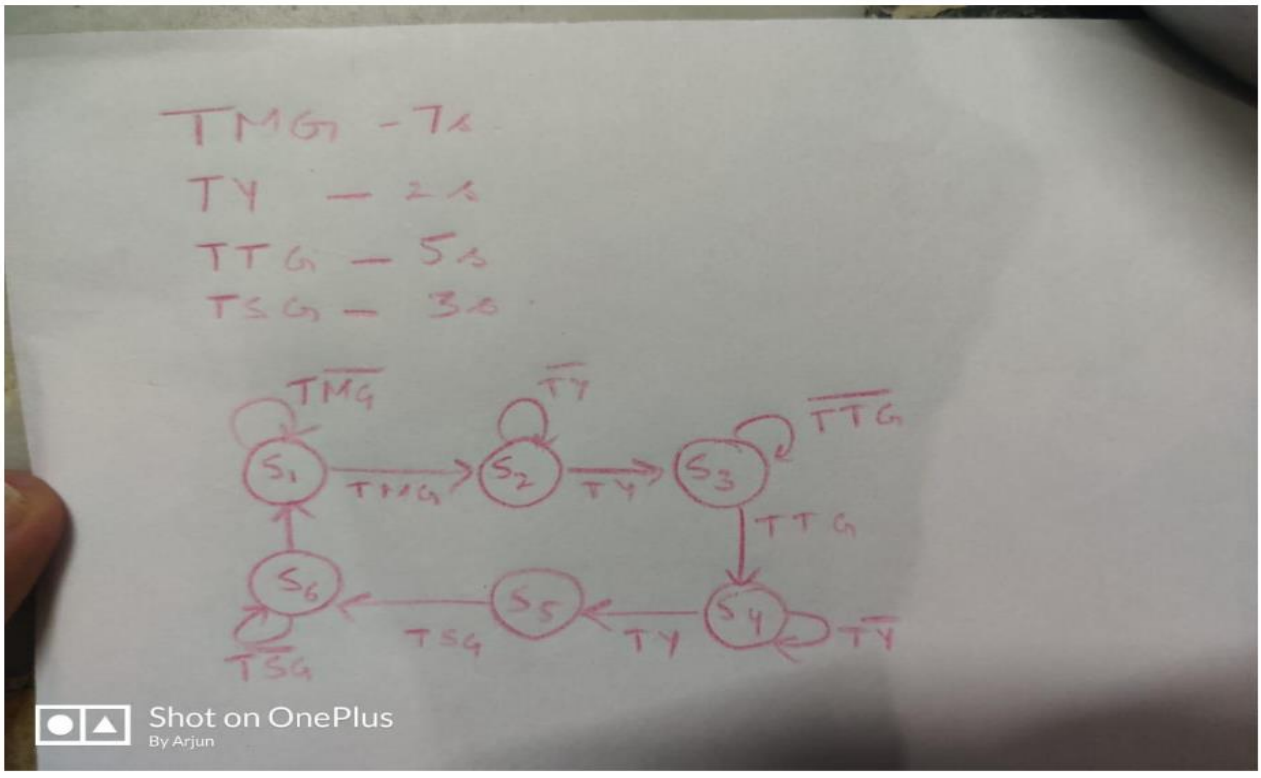
Let's understand the problem statement through the image given below.



Architechture:

The six cases present here eventually turn to the six states .

This is the state diagram:



SOURCE CODE :

```
module Traffic_Light_Controller(


   input clk,rst,     output reg
[2:0]light_M1,     output reg
[2:0]light_S,     output reg
[2:0]light_MT,
   output reg [2:0]light_M2
   );

   parameter  S1=0, S2=1, S3 =2, S4=3, S5=4,S6=5;
   reg [3:0]count;
   reg[2:0] ps;
   parameter  sec7=7,sec5=5,sec2=2,sec3=3;


   always@(posedge clk or posedge rst)
      begin
if(rst==1)
begin
ps<=S1;
count<=0;
end        else



                    case(ps)
S1: if(count<sec7)
begin                      ps<=S1;
count<=count+1;
                   end
else                      begin
ps<=S2;
count<=0;
end
             S2: if(count<sec2)
begin                      ps<=S2;
count<=count+1;
                   end

                else
                   begin
ps<=S3;
```

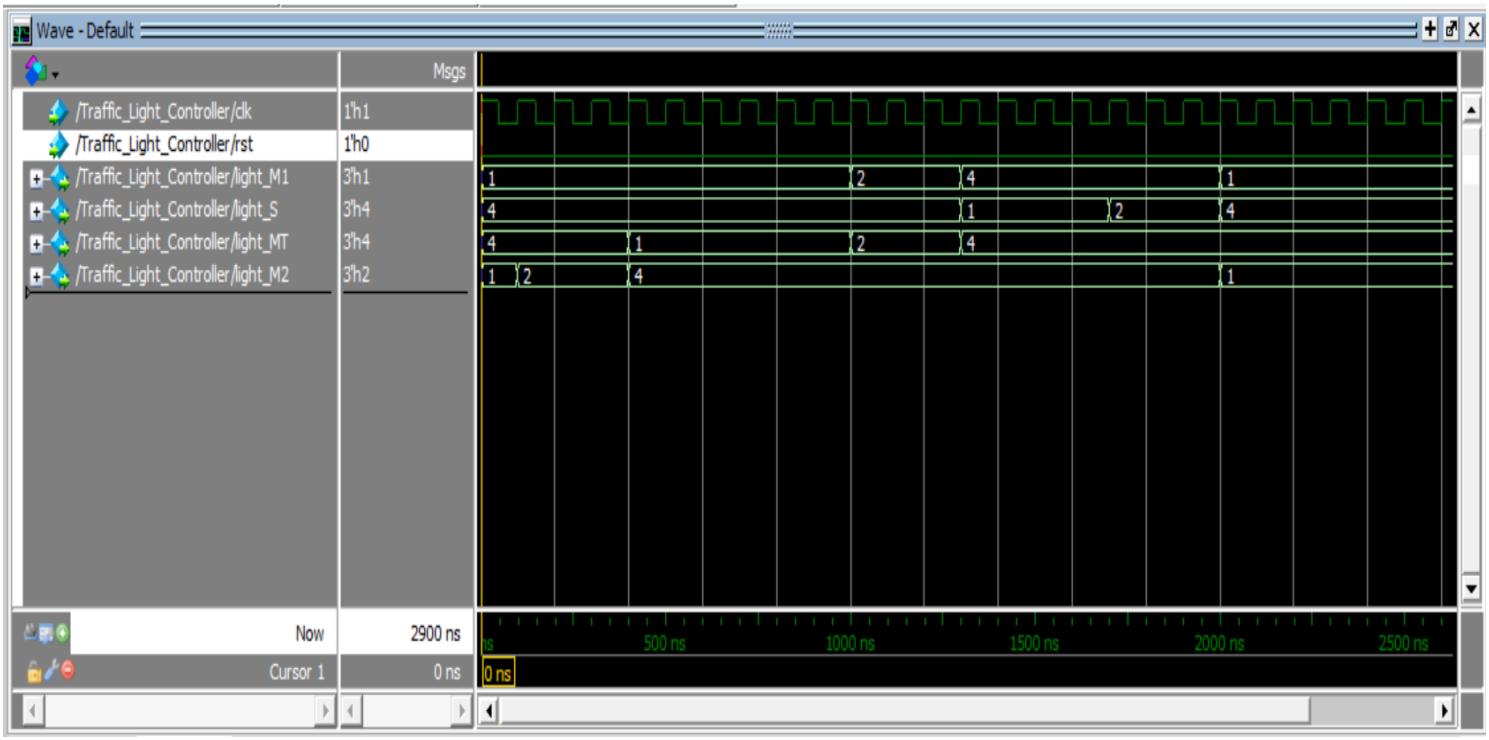
```
count<=0;
end
            S3: if(count<sec5)
begin                   ps<=S3;
count<=count+1;
end
                else
begin
ps<=S4;
count<=0;
end
            S4:if(count<sec2)
begin                   ps<=S4;
count<=count+1;
end
                else
begin
ps<=S5;
count<=0;
                  end
            S5:if(count<sec3)
begin                   ps<=S5;
count<=count+1;
end
                else
begin
ps<=S6;
count<=0;
end

            S6:if(count<sec2)
begin                   ps<=S6;
count<=count+1;
end
                else
begin
ps<=S1;
count<=0;
end             default:
ps<=S1;
endcase           end

        always@(ps)
        begin
case(ps)
                            S1:
begin
```

```
light_M1<=3'b001;
light_M2<=3'b001;
light_MT<=3'b100;
light_S<=3'b100;
end                 S2:
begin
light_M1<=3'b001;
light_M2<=3'b010;
light_MT<=3'b100;
light_S<=3'b100;
end                 S3:
begin
light_M1<=3'b001;
light_M2<=3'b100;
light_MT<=3'b001;
light_S<=3'b100;
end                 S4:
begin
light_M1<=3'b010;
light_M2<=3'b100;
light_MT<=3'b010;
light_S<=3'b100;
end                 S5:
begin
light_M1<=3'b100;
light_M2<=3'b100;
light_MT<=3'b100;
light_S<=3'b001;
end                 S6:
begin
light_M1<=3'b100;
light_M2<=3'b100;
light_MT<=3'b100;
light_S<=3'b010;
                end
default:
begin
                  light_M1<=3'b000;
light_M2<=3'b000;
light_MT<=3'b000;
light_S<=3'b000;
                end
endcase
          end
endmodule
```

SIMULATION OUTPUT :



In summary, the output waveform of the traffic light controller implemented by this Verilog code would show the various states of the traffic lights (represented by the "light_M1", "light_M2", "light_MT", and "light_S" signals) changing over time according to the state transitions and timer defined in the Verilog code. However, the specific details of the output waveform would depend on the specific timing parameters and initial conditions of the module.

CONCLUSION:

Thus the Above circuit of Traffic Light Controller was successfully implemented and simulated too on the Xilinx Software as well as successfully hardware tested on Xilinx FPGA kit.