Mini Project report

On

TRAFFIC LIGHT CONTROLLER USING VERILOG

Submitted in partial fulfillment of the degree of

Bachelor of Engineering

In

Electronics and Communication Engineering

Submitted by
A Devipriya

ABSTRACT

Traffic control is a challenging problem in many cities. This is due to the large number of vehicles and the high dynamics of the traffic system. Poor traffic systems are the big reason for accidents, time losses. In this method of approach, it will reduce the waiting time of the vehicles at traffic signals. The hardware design has been developed using Verilog Hardware Description Language (HDL) programming.

Verilog designing is hardware descriptive language, the name itself suggest that it deals with the hardware designing and simulation. Basically, it becomes very difficult to mount the various electronic components on breadboard or PCB circuit. It also takes too much time for the simulation and sometimes many errors occur because of improper connection of components onto the circuit.
And thus, to overcome this factor hardware descriptive language comes into conclusion. we can code the process using Verilog and we can mount it on a circuit or just upload it to the circuit accordingly so that particular circuit will work as according to the code we have written.

HDL language is often used for sequential circuits like shift register, combinational logic circuit like adder, subtractor etc. Basically it describes the digital systems like microprocessor or a memory. Whatever design that is described in HDL are independent, it has its unique state of work, very much easy to simulate, designing and debugging, and very useful than schematics, especially for large circuits thus, to overcome difficulties or problems to design the circuits manually with breadboard and PCB, use of Verilog designing in this complex world is increasing a way better.

This project deals with a basic design of a T - Shaped road for traffic light control. The output of system has been tested using Xilinx 14.5.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1 : INTRODUCTION

Traffic light signal controlling is most important and essential thing for any country to protect the people from heavy load of traffic. Before this kind of invention there was much difficult for traffic police to handle the heavy traffic (particular direction given manually). Thus, traffic signal controlling technology made much easier to handle the heavy loads of traffic. Safe movement of vehicles without any type of collision, accidents.

Apart from the traffic it is very necessary for the people to cross the roads at particular time interval. And this is only possible by controlling the traffic by giving some kind of signal. Analyzing the traffic, estimating the delays to the areas is crucial part. Population can be predicted using GPS trekking and thus we can easily estimate the amount of time to be taken for delay. A perfect aligning of cars, bikes, cycles, trucks with orderly flow by giving right of way, this makes the process very systematic and even in the presence of heavy traffic accident rate goes down which is one of the biggest advantage.

Timing and the delays of particular signal plays a vital role because it is very necessary for us to keep information about the amount of traffic which present in the local area. This gives us an idea about timing and delay requirement of every signal in the local area. As we know the timing depends on traffic volume and its not necessary for us to have same traffic volume at each day so for that we can estimate average volume of traffic around the local area. Average can be made with consideration of 20 days for example we will analyse the traffic of 20 days then we will take an average of it and estimate delays and timings of it. It is very necessary factor for us to have adaptive mechanism.

Apart from that we can estimate the traffic volume using GPS, which can give you volume prediction every day. For this we no need to take an average of particular days. GPS give us more correct prediction of volume of traffic than calculating an average of traffic of particular days. Thus, coordinating signal timing minimizes stating and stopping of vehicles in the traffic to avoid the traffic jam.

Chapter 2 : LITERATURE SURVEY

Traffic Light Systems :

A traffic light system is an electronic device that assigns right of way at an intersection or crossing or street crossing by means of displaying the standard red, yellow and green colored indications. In addition, it also works in conjunction with pedestrian displays to assign pedestrian crossing right of way.

A traffic light, also known as traffic signal, stop light, stop-and-go lights, is a signaling device positioned at a road intersection, pedestrian crossing, or other location in order to indicate when it is safe to drive, ride, or walk using a universal colour code.
Nowadays,
A red light meant traffic in all directions had to stop.
A yellow light meant cross-town traffic would have to slow and,
A green light would to go or proceed.

The difficulty in understanding this confusing color sequence was compounded by neighboring towns using another system. The development of an intelligent control structure ensures an optimal solution for all participants in the transportation and road traffic system. There are different ways controlling road intersections. In the simplest cases the right-hand rule or, if the traffic is higher, a round about or the signal of a policeman can help steer the traffic.

However, especially in big cities, in the complicated cases when the roads in the intersection have several lanes, the use of traffic lights cannot be avoided. An additional issue arises when in the intersection not only roads but also railroad tracks take part, what often occurs in suburban traffic situations. The most common way to handle this type of intersection is the conventional cyclic lights control. In more enhanced control, the traffic in different directions is monitored by sensors and the signals thus obtained control the traffic lights. In this method the control is adapting to the traffic. The general problem is the huge number of variables and the need for large computing efforts.

Benefit of Traffic Light Controller :

When properly used, traffic control signals are important devices for the control of vehicles in road. They assign the right-of-way to a choice of traffic movements and thereby deeply influence traffic flow.
Traffic control signals that are properly designed, located, operated, and maintained will have one or more of the following advantages :

- Provide orderly movement of traffic.
- Minimize completing movement.
- Coordinated for continuous movement.
- Provide driver confidence by assigning right way traffic control signals are often considered, a cure for all traffic problems at intersections.

This belief has led to traffic control signals being installed at many locations where they are not needed, adversely affecting the safety and efficiency of vehicular, bicycle, and pedestrian traffic. Traffic control signals, even when justified by traffic and roadway conditions, can be ill-designed, ineffectively placed, improperly operated, or poorly maintained. While traffic signals can help in locations where they are justified and installed properly, they also have disadvantages.

There will always be some disadvantages even if the signal is justified. Most traffic signals will have the following components or part:

- Main display with red, yellow and green lights.
- Traffic signal cabinet containing the traffic signal controller and Vehicle Detection systems.
- Inductive loops or sensors.

Traffic jamming is a critical predicament in many of the cities and towns all over the world. Traffic congestion has been causing many setbacks and challenges in the major and most occupied cities all over the globe. This traffic jam directly impacts the productivity of the workers, traders, suppliers and in all affecting the market and raising the prices of the commodities in a way light.

The problem of heavy jam is happened because of never configure the level of jam in each way and set the delay time. Another problem represents when there is no jam, but the waiting still continues. The solution for these problems is to determine the level of jam and set the delay time. This problem need of evaluation of the traffic policeman, and then there is need for manual control of the traffic.

The target of this paper is to propose system provide solution for all above problems with least possible cost. Traffic light controller (TLC) can be implemented using microcontroller, FPGA, and ASIC design. FPGA has many advantages over microcontroller, some of these advantages are; the speed, number of input/output ports and performance which are all very important in TLC design, at the same time ASIC design is more expensive than FPGA.

Nowadays, FPGA becomes one of the most successful of today's technologies for developing the systems which require a real time operation. FPGA is a re-configurable integrated circuit that consists of two dimensional arrays of logic blocks and flip-flops with an electrically programmable interconnection between logic blocks.

The reconfiguration property enables fast prototyping and updates for hardware devices even after market launch. Most of the TLCs implemented on FPGA are simple ones that have been implemented as examples of Finite State Machine (FSM).

The Verilog language has been selected for programming the FPGA to fill two important needs in the design process.

- Firstly, it gives full description of the structure of a design that is how it is decomposed into sub-designs, and how those sub-designs are interconnected.
- Secondly, it allows simulating the design before starting the manufacturing.

Accordingly, the designers can quickly compare alternatives and test for correctness without the delay and expense of hardware prototyping.

Benefits of Using Verilog HDL (Hardware Description Language) :

Verilog is a widely used Hardware Description Language (HDL) for designing digital circuits. It can also be used for modeling analog circuits. Verilog is a descriptive language that describes a relationship between signals in a circuit.
A Verilog model describes a unit of digital hardware in terms of :

- Interconnections of other hardware unit whose models prescribe their behavior in a simulation.
- Behavioral / procedural algorithms that abstractly describe input/output behavior that could be personified in a hardware unit.

Hardware description language (HDL) is divided by two types, Verilog and VHDL (VHSIC – Very High Speed Integrated Circuit Hardware Description Language). Both have its advantages and its disadvantage.
In this project, Verilog HDL was chosen because it's used for synthesis of logic circuits (synthesizable code), used for verification purposes of a circuit (can be analog or digital or mixed signal), can be used by combining synthesis & verification (synthesizable & behavioral code) and it used for netlist representation of a synthesizable circuit (structural code).

The advantages using Verilog HDL are shown below :

- Easy to write.
- Easy to understand as it similar to C program.
- Easier to learn compared with VHDL.

Chapter 3 : METHODOLOGY

Verilog is a hardware descriptive language (HDL) which is generally used to model electronic system, that is, we can design a circuit and the function of circuit can be control by Verilog coding. Thus, design and verification can be done using Verilog designing. Similarly, traffic light signal controlling can be done using Verilog (hardware descriptive language). so now we have design the traffic light signal system with Verilog using sequence detector method.
We have considered a T - shaped road and heavy load of traffic is present over this road where we have to control it using traffic signal.
Six cases have been considered here and analyzed using state diagram and state table.

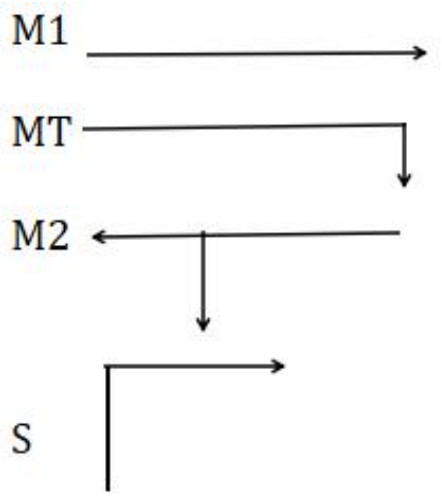

Figure 3.1 : Directions chart

The directions, M1, MT, M2, S, that is been considered for analysis of our problem is shown in the figure 4.1. And, the problem statement is explained in the figure 4.2.
Six states, S1, S2, S3, S4, S5, S6 are taken into consideration and state diagram(Figure 4.3), state table(Figure 4.4) is made using the following logic explained in the figure 4.2.

- Green light indicates that there is no traffic and there is easy flow of vehicles in that route/direction.
- Red light indicates that there is a traffic jam and that route is blocked for the vehicles to move and,
- Yellow light indicates that the route has medium flow of vehicles.

Time delays for changing from one state to another(shown in Figure 4.2) is considered as, TMG(from S1 to S2), TY(from S2 to S3), TTG(from S3 to S4), TY(from S4 to S5), TSG(from S5 to S6) and TY(from S6 to S1) and the cycle continues.

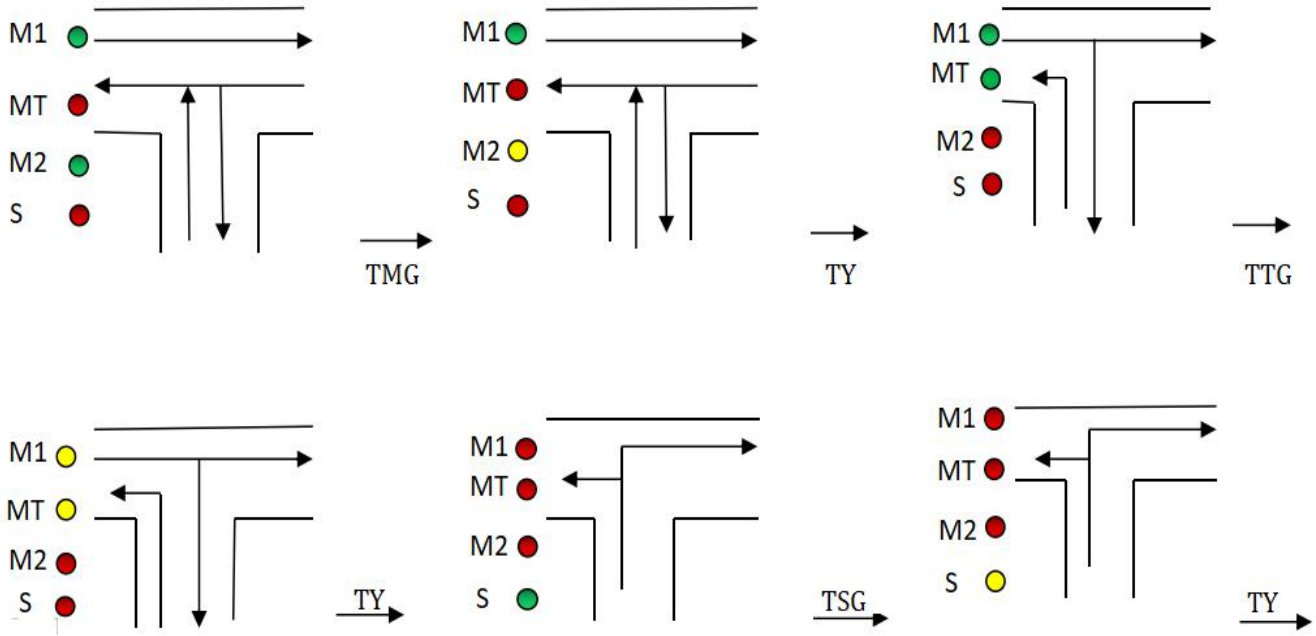

Figure 3.2 : Problem statement (Logic)

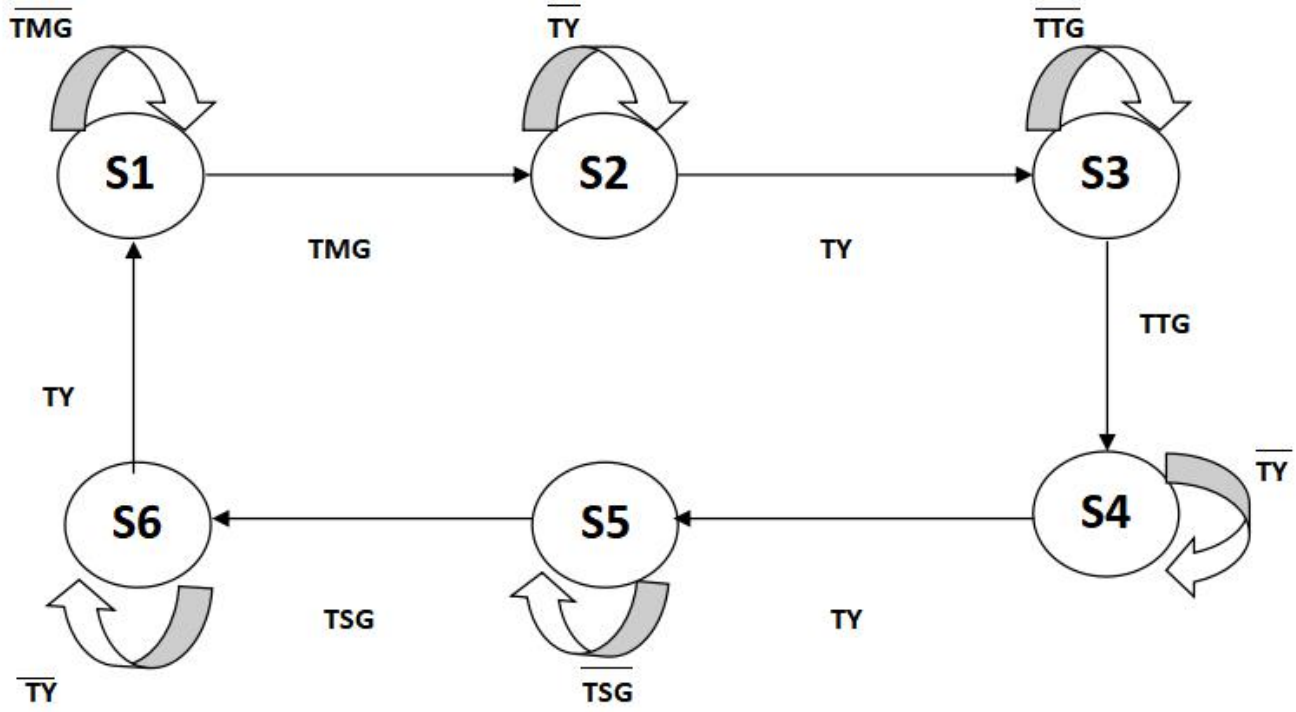

Figure 3.3 : State Diagram

In Figure 4.3, The time delays are considered as follows :

- TMG = 7 seconds
- TY = 2 seconds
- TTG = 5 seconds
- TSG = 3 seconds

Until TMG seconds, the signal will remain in S1 state, and after TMG seconds, it will move to S2 state. Until TY seconds it will remain in S2 state and after TY seconds, it will move to S3 state, and so on. After TY seconds, in state S6, it will go back to S1 state and the cycle continues.
The state table for the problem statement is shown below in Figure 4.4.

PRESENT STATE ABC	INPUT	NEXT STATE A*B*C*	S T	M1 RYG	M2 RYG	MT RYG	S RYG
000	-	001	1	000	000	000	000
001	$\overline{TMG}$ TMG	001 010	0 1	001	001	100	100
010	$\overline{TY}$ TY	010 011	0 1	001	010	100	100
011	$\overline{TTG}$ TTG	011 100	0 1	001	100	001	100
100	$\overline{TY}$ TY	100 101	0 1	010	100	010	100
101	$\overline{TSG}$ TSG	101 110	0 1	100	100	100	001
110	$\overline{TY}$ TY	110 001	0 1	100	100	100	010
111	-	000	0	000	000	000	000

Figure 3.4 : State Table

In Figure 4.4,

- R = RED,
- Y = YELLOW and,
- G = GREEN.

ST = State Transition; A, B and C are considered as the present state.
The state table is made, considering the Logic diagram/problem statement given in Figure 4.2.
From the Figure 4.2, Figure 4.3 and Figure 4.4,

In state S1(001); M1 = GREEN, implies, RYG value = 001,
MT = RED, implies, RYG value = 100,
M2 = GREEN, implies, RYG value = 001 and,
S = RED, implies, RYG value = 100.

After TMG seconds,

In state S2(010); M1 = GREEN, implies, RYG value = 001,
MT = RED, implies, RYG value = 100,
M2 = YELLOW, implies, RYG value = 010 and,
S = RED, implies, RYG value = 100.

After TY seconds,

In state S3(011); M1 = GREEN, implies, RYG value = 001,
MT = GREEN, implies, RYG value = 001,
M2 = RED, implies, RYG value = 100 and,
S = RED, implies, RYG value = 100.

After TTG seconds,

In state S4(100); M1 = YELLOW, implies, RYG value = 010,
MT = YELLOW, implies, RYG value = 010,
M2 = RED, implies, RYG value = 100 and,
S = RED, implies, RYG value = 100.

After TY seconds,

In state S5(101); M1 = RED, implies, RYG value = 100,
MT = RED, implies, RYG value = 100,
M2 = RED, implies, RYG value = 100 and,
S = GREEN, implies, RYG value = 001.

After TSG seconds,

In state S6(110); M1 = RED, implies, RYG value = 100,
MT = RED, implies, RYG value = 100,
M2 = RED, implies, RYG value = 100 and,
S = YELLOW, implies, RYG value = 010.

And after S6 state, the cycle repeats and goes to S1 state.

Chapter 4 : RTL Code

```
`timescale 1ns / 1ps
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
```

Below shown figure is the RTL Schematic View obtained in Xilinx :

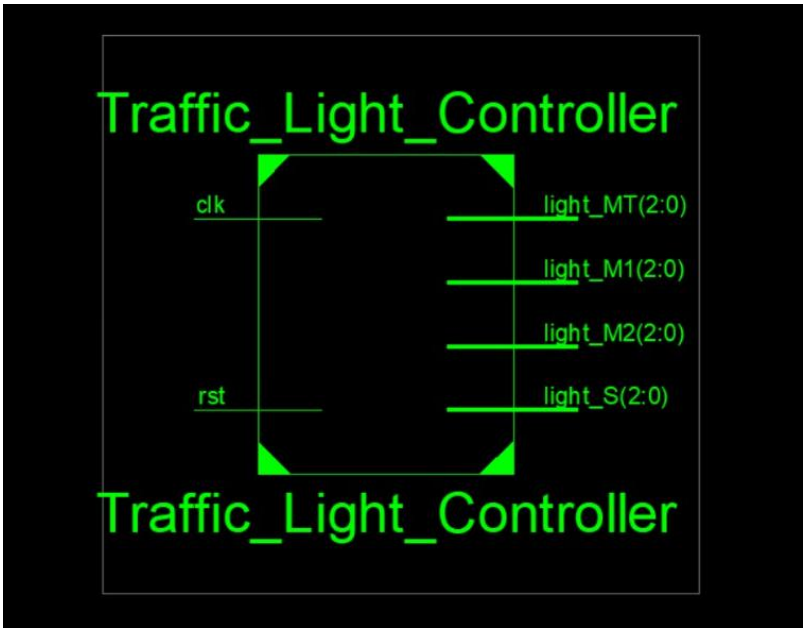

Figure 4 : RTL Schematic View

Chapter 5 : TESTBENCH

```
`timescale 1ns / 1ps
module Traffic_Light_Controller_TB;
reg clk,rst;
wire [2:0]light_M1;
wire [2:0]light_S;
wire [2:0]light_MT;
wire [2:0]light_M2;
wire [3:0] count;
wire [2:0] ps;
Traffic_Light_Controller
dut(.clk(clk), .rst(rst), .light_M1(light_M1), .light_S(light_S), .light_M2(light_M2),
.light_MT(light_MT) );
initial
begin
	clk=1'b0;
	forever #(1000000000/2) clk=~clk;
end
initial
begin
	rst=0;
	#1000000000;
	rst=1;
	#1000000000;
	rst=0;
	#(1000000000*200);
	$finish;
	end
endmodule
```

Chapter 6 : RESULTS

Initial Outlet of the waveform after the simulation :

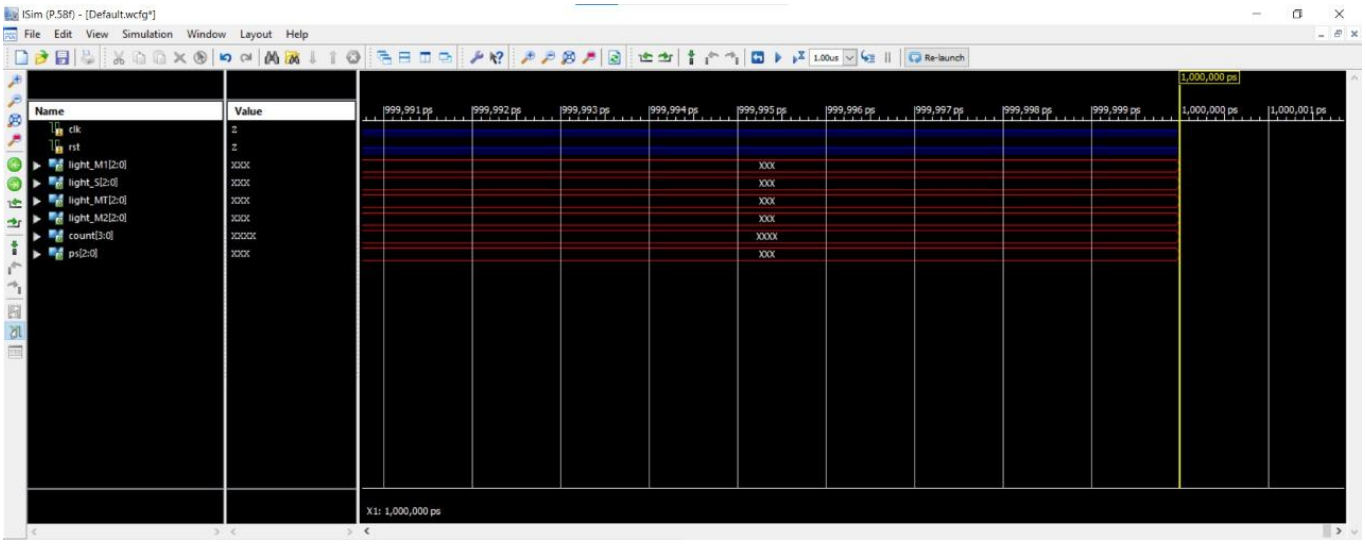

Figure 6.1 : Output Waveform 1

After giving the clock :

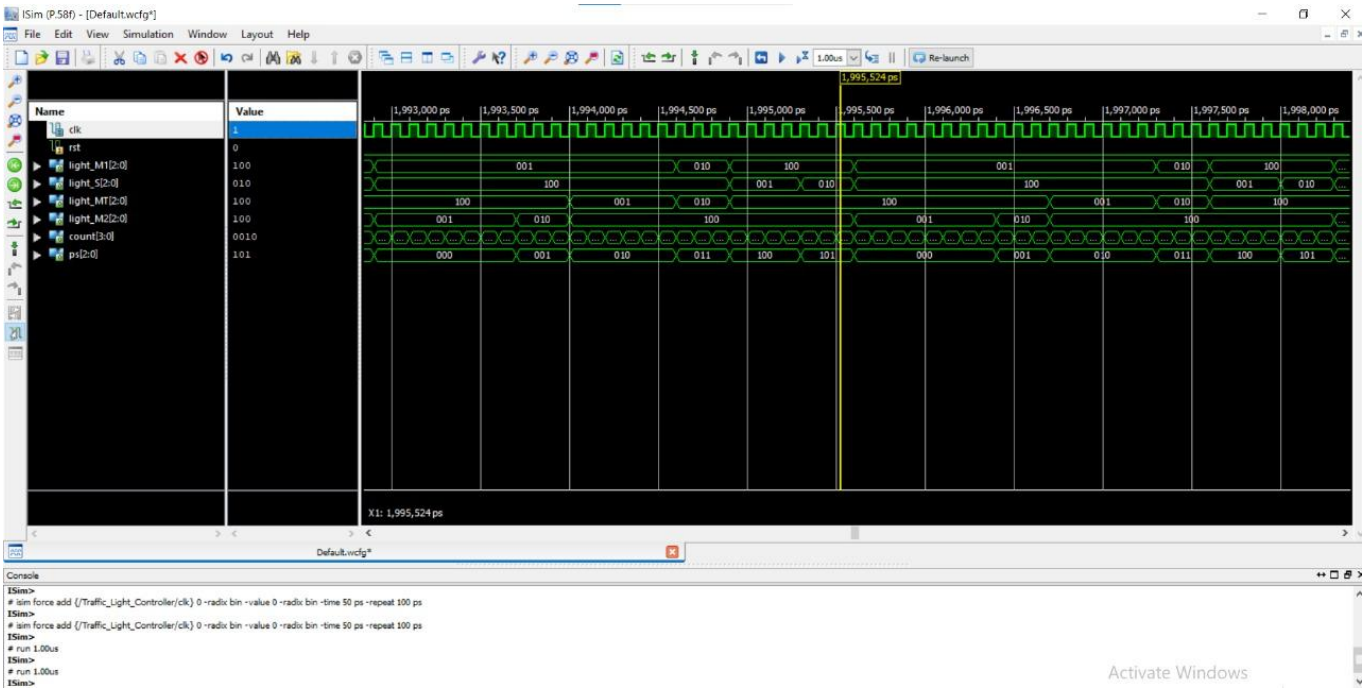

Figure 6.2 : Output Waveform 2

Closer Look :

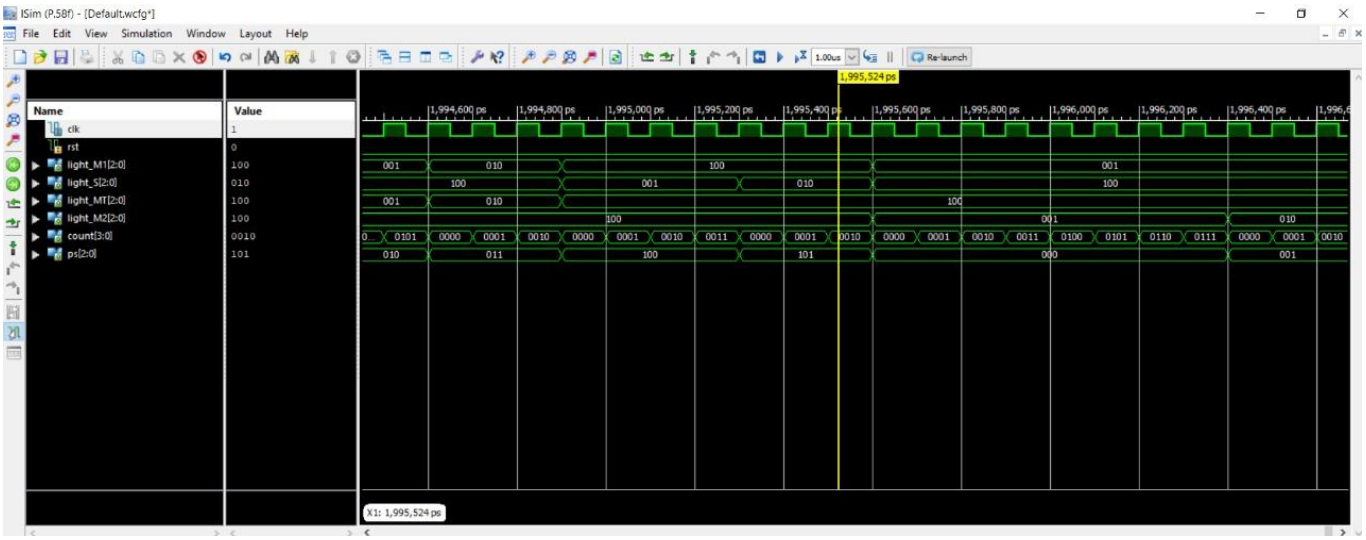

Figure 6.3 : Output Waveform 3

Chapter 8 : CONCLUSION AND FUTURE WORK

In this model we have observed various stages which describes about every signals. For example, Consider that at first stage (north-south end) signals gives some indication. Then, the signal is red that means signal at east-west side gives a green indication and traffic moves to their respective direction. Then after some delay yellow signal is obtain at east-west side and after the red signal arrives at the same time at the north-south end red signal goes off and green signal gets on and traffic moves to their particular direction. In this way process continues in the loop every day.

The modern ways of multi-way traffic management improves the traffic condition up to a large extent. Traffic intensity is sensed and accordingly time is allotted for traffic to pass. Verilog HDL is used to circuit description, code is generated which is simulated using xilinx14.5.

This traffic light control system works on the concept of fixed time allocation at each side of the junction which cannot be changed as per varying traffic density. Timings allotted at every junction are fixed. Sometimes higher traffic density at one side of the junction demands longer time duration for green signal compared to the standard allotted time.

Thus, traffic light control system helps to conduct orderly flow of vehicles. There are lot many issues of obstacles, high level accidents which occurs every day. So, traffic signal controller prevents such occurrences. Still many areas or small towns don't have the traffic light control facilities. And thus, many accident problems occur at those areas. Therefore, it is a primary purpose to have such facility in order to control and maintain the area.

This project can be enhanced in such away as to control automatically the signals depending on the traffic density on the roads using sensors like IR detector/receiver module extended with automatic turn off when no vehicles are running on any side of the road which helps in power consumption saving.

A lot of development ideas for work in future can be implemented, such as using solar energy (independent power supply, i.e. saving the power). Using the GPRS map as an additional step for development and choosing the best road for the emergency and police vehicles. For national highways we can also design the 8 road traffic light controllers.

Chapter 9 : REFERENCES

1) Laboratory Exercise #12, The Traffic Light Controller Lab. “ECEN 248: Introduction to Digital Design”. Department of Electrical and Computer Engineering Texas A&M University.
2) Rajeev Madhavan, Verilog HDL Reference Guide, Automata Publishing Company, CA, 1993. ISBN 0-9627488-4-6
3) E. Sternheim, Rajvir Singh, Rajeev Madhavan, Yatin Trivedi, Digital Design and Synthesis with Verilog HDL, Automata Publishing Company, CA, 1993. ISBN 0-9627488-2-X
4) https://intentswell.blogspot.com/2021/03/traffic-light-controller-project-report.html
5) https://www.semanticscholar.org/paper/An-Advanced-Traffic-Light-Controller-using-Verilog-B.-ala/1ca8a8e50c2c8273a3ee1247a6057f8eb4f402ee
6) https://www.academia.edu/21200096/DESIGN_AND_IMPLEMENTATION_OF_TRAFFIC_LIGHTS_CONTROLLER_USING_FPGA_A_Project_Based_Laboratory_Report_in_partial_fulfilment_for_the_award_of_III_IV_B_Tech_I_Semester_Submitted_by_Lab_Instructor
7) https://vlsicoding.blogspot.com/2013/11/verilog-code-for-traffic-light-control.html
8) https://www.fpga4student.com/2016/11/verilog-code-for-traffic-light-system.html
9) https://www.slideshare.net/UtkarshDe/four-way-traffic-light-conrol-using-verilog