Four Way Traffic Light Controller Design Using Schematic and HDL

Farah Naz[1], Harshit Kapil[2], Ruhool Khan[3], Shrayansh Gupta[4]
[1,2,3,4]*Raj Kumar Goel Institute of Technology (RKGIT), Ghaziabad, Uttar Pradesh (201003)*
Department of Electronics and Communication Engineering, NBA Accredited

***Abstract* - Traffic Light Control System is designed specifically to control and manage the movement of vehicles. Traffic Light Control System is basically based on specific switching of traffic lights i.e. red, yellow, and green. This Traffic Light sequence is generated by using a specific switching technique or mechanism which will be very helpful in control a traffic on a roads in a specified sequence. It is a sequential machine which we design using basic logic gates. The focus of project to minimize the traffic congestion using different waiting time. The simulation is done through the Verilog language and implemented using Xilinx 14.7. In this project Xilinx software is used to write code, schematic edit and synthesis.**

***Index Terms* - Traffic Light Control System, Xilinx ISE, Schematics, Verilog, FPGA, EDA Tool.**

I.INTRODUCTION

Traffic jams are major problem faced in many of the metropolitan cities and towns all over the earth. Traffic congestion causes many problems and challenges in cities. To travel within the cities has become a big problem. Due to these problems people lose their time, money and most importantly the energy resources will be get exhausted very rapidly due to the continual use in the automobiles. This traffic jam decreases the productivity of the workers, traders, suppliers and in all effecting the market. To solve these traffic related problems, we have to build new devices & infrastructure to make ease of transport but at the same time we have to make them smart. The only disadvantage in construction of the more roads on good facilities is that it makes the surroundings more obstructed and congested, but then this TLC system would be able to provide much relief from the traffic and it will provides new ways to ease the traffic. All the countries are working to solve the problem of traffic flow and advance transportation and reduce the demand of vehicle use.

Lights of system have been used to manage and control the congestion of traffic at each road of intersection using light cycle schedules. The traffic light works on the specific manner switching of red, yellow and green lights in a particular way with unique time form. They provide safe management of traffic to share the road intersection. The constant delay at each road intersection decreases the traffic flow and then these results into decrement of the traffic efficiency all over the road network.

Traditional traffic control system had a drawback due to fixed timing of traffic signals, the traffic had to hold for long time on the lane with less vehicle while the lane having more vehicle cannot pass in short time. So, we need to develop a reliable, fast and smart traffic control system capable to control the traffic in rush hours without a need of traffic man.

Our project is basically first be digitally designed using logic gates. It is a sequential machine which can be model it in Verilog by different methods just like Gate Level Modeling, Data Flow Modeling & Finite State Machine. We use Finite State Machine (FSM) method to model our TLC. Our whole project (i.e. Schematic, Verilog Code, Synthesis) done using XILINX 14.7 design suit. TLC is further implemented on Field Programmable Gate Array using Xilinx software

II.OBJECTIVE

Traffic Light Controllers are used to manipulate traffics at areas that are shared among multiple lanes called intersections by managing the access of traffics to the particular lane in intersections and create effective interval of time between various junctions. The primary objective of this project is to control the traffic movement of crossing lanes and acquire the finest use of the traffic light. Generally, traffic light control system of all main roads are managed with a rigid timing system but on the other hand the smaller roads are controlled by the sensors autonomously.

Our main objective is to design a specific four-way traffic system that have flexible waiting time with respect to density of vehicles as such it does not create any congestion at the intersection and save time of the people.

III.DESIGN OF TRAFFIC LIGHT CONTROL

3.1 ROAD STRUCTURE:

Our objective to design traffic control system, so we have taken a four-way simple intersection as shown in Figure.1.

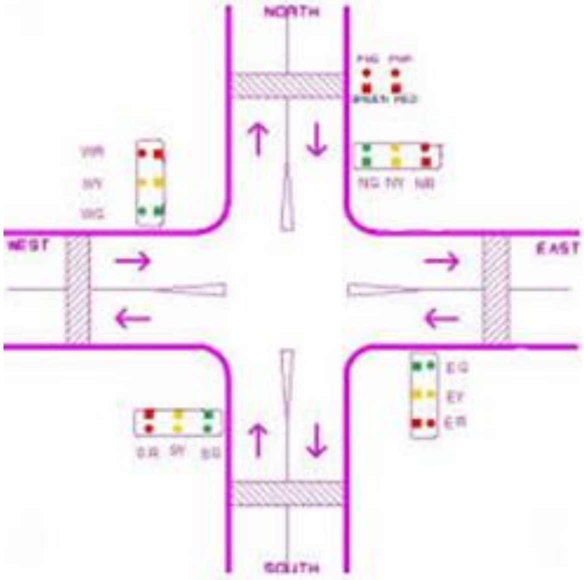


FIGURE 1: MECHANISM

We have taken 4 direction and each direction has there 3 specific lights (RED, GREEN, YELLO W). We have provided a fixed time difference as for default case. As timing plays very important role in the TLC that must be set very wisely and precisely, so that any dangerous situation i.e. accident of cars can be avoided. The TLC timing setting are as follow: -

•Green

The green light timer is 16sec for rush lane and 8sec for not rush lane.

•Yellow

The timing of yellow to be set for 4 sec for rush and not rush lane.

•Red

The timing of red to be fixed for 2 sec for rush and not rush lane

3.2 FUNCTIONAL BLOCK:

Here we have proposed the system for the above-described problem as shown in figure2. In this figure we have shown the Logic Block diagram

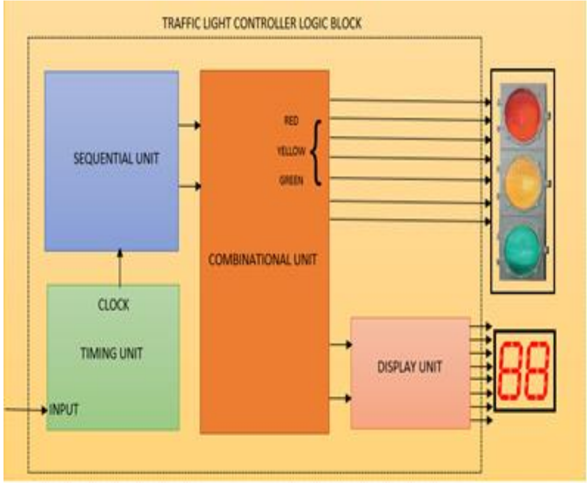
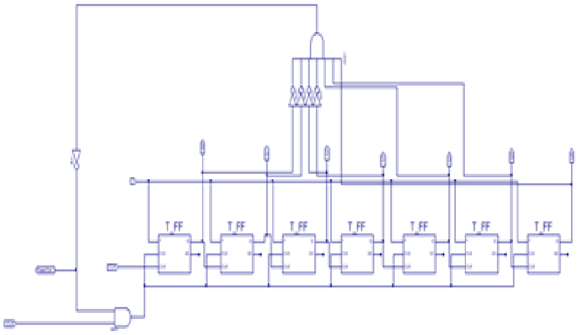


FIGURE 2: BLOCK DIAGARM OF SYSTEM

of our traffic light controller it consists of four main units are as follows combinational, sequential, timing and display unit. Each unit has its own functionality to perform in the controller.

3.2.1 SEQUENTIAL UNIT:

Sequential unit responsible for generating state transition state for the combinational unit to perform respective task. This unit consists of the counters here is the example of asynchronous counter. Asynchronous Up counter is a sequential circuit that counts from 0 to some defined state. It is an asynchronous counter built using T flip flops. Here we are using 7 bit counter that counts 128 states i.e. from 0 to 127. Following is the representation of schematic and RTL view of the same.



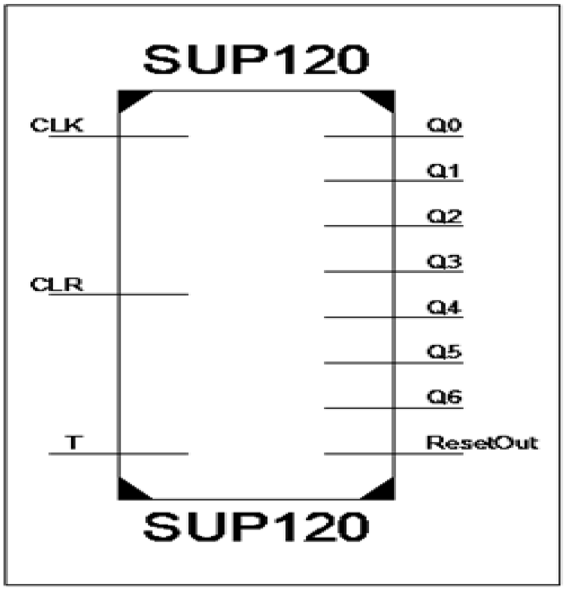


FIGURE 3: RTL AND SCHEMATIC VIEW OF COUNTER

3.2.2 COMBINATIONAL UNIT:

Combinational unit main task to perform the comparison of generated state transition state to a fixed state stated. So to do so we use a comparator to compare bits and perform desired task which to provide display unit and traffic indicator a specific value.

The comparator is combinational logic circuit that compare 2 binary numbers and produce result on comparison of those two numbers. Here we are using 8-bit comparator that takes 8-bit inputs of input A & 8 inputs of input B & have 3 outputs : AGB(input1 is greater than input2); AEB(input1 & input2 are equal) BGA(input2 is greater than input1). This comparator has an additional feature of the reset pin that is when reset pin is active low(0) then all three outputs are zero.

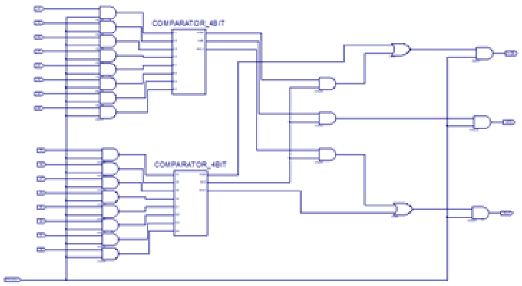
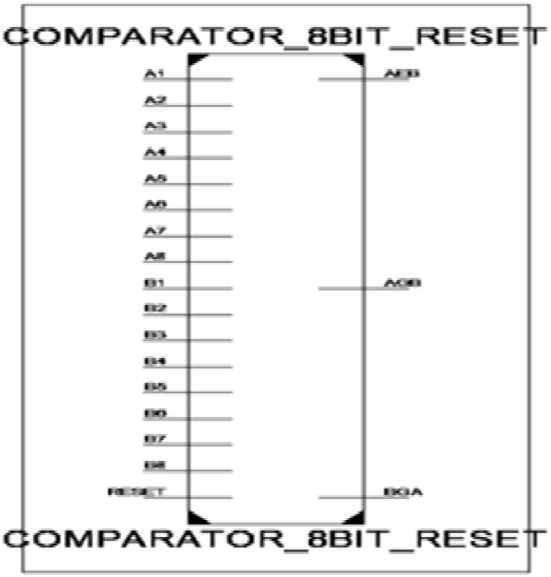


FIGURE 4: RTL AND SCHEMATIC VIEW OF COMPARATOR



3.2.3 TIMING UNIT:

Timing unit provide the clock signal to the whole design which is further responsible for the proper generation of the clock pulse with desired frequency to provide desired output.

This asynchronous clock is a clock whose time period is 1 second or frequency is 1 hertz. This clock is built using connecting number of T flip flops in series.

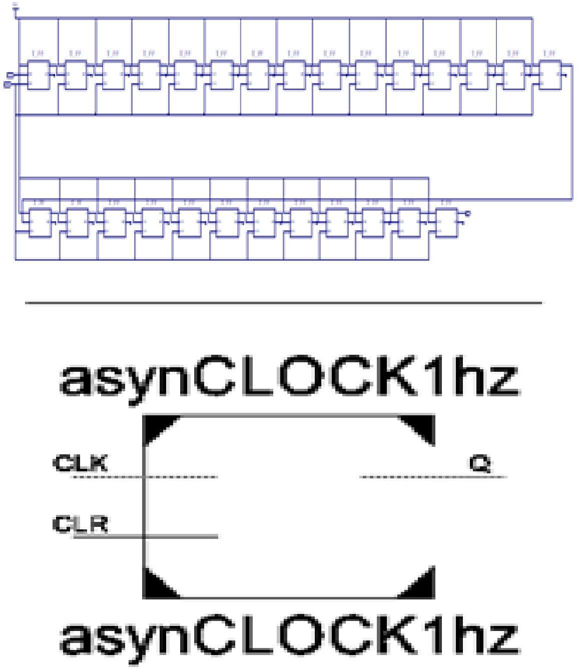


FIGURE 5: RTL AND SCHEMATIC VIEW OF CLOCK

3.2.4 DISPLAY UNIT:

Display unit is responsible for the generation of the signal required for the seven segment of the board.

It generates the combinational output to a specific bits to generate desired output at the seven segment.

Here we are using 3X8 decoder that have 3 inputs, 8 outputs & 1 enable line.

The decoder will works only when the enable pin is in active high state. When the enable is low, the output of decoder is zero.

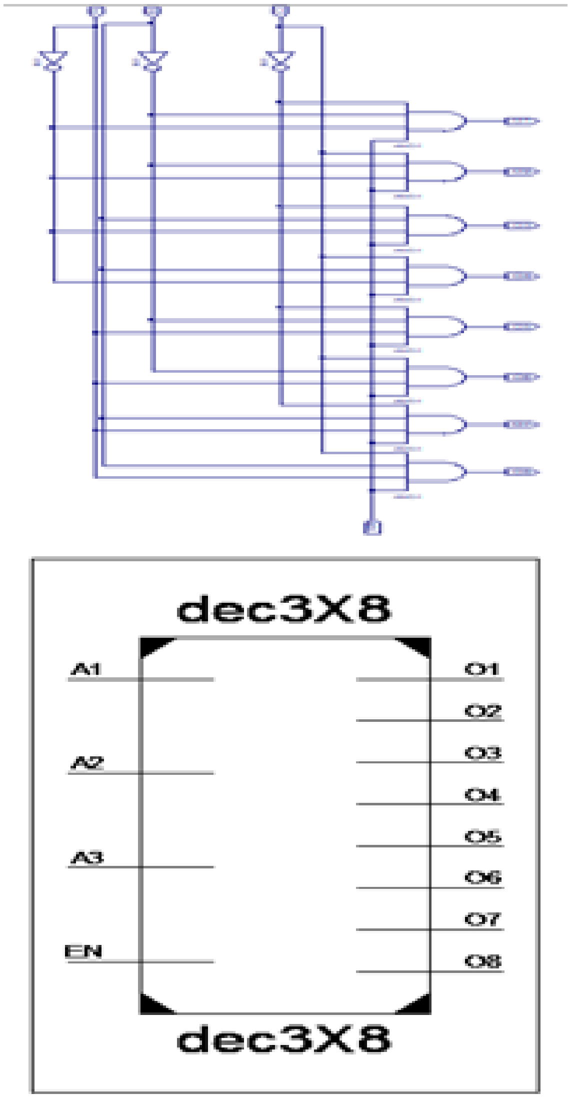


FIGURE 6: RTL AND SCHEMATIC VIEW OF DECODER

3.3 DIGITAL DESIGNING:

For better learning and understanding of TLC we have to design the schematic of the controller using basic logic gates. We have to design this sequential machine using basic component of sequential circuit. We first design flip flop, latches and further counters to complete our project. Our project also requires some combinational circuitry to provide better timing logics to design we use bottom up approach to construct the project. We design schematic on Xilinx tool .

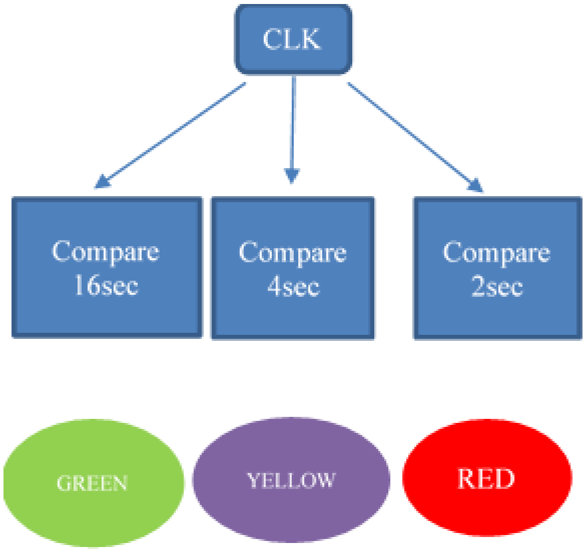


FIGURE 7: LOGIC OF TIMING BLOCK

Here in the figure.2 shows basic working of each lane traffic lights. It shows the switching mechanism of lights during the rush time as it for low density time the green will automatically changes to 8sec provide the less time for lane as it not has any huge traffic. As it has specified for each lane and timing between each lane is set very precisely and accurately.

3.4 STATE TRANSITION TABLE

STATE	INPUT	TIME DURATION	L 1	L 2	L 3	L 4	L 5	L 6	L 7	L 8
S0	rst=1	Infinite	R	R	R	R	R	R	R	R
S1	rst=0	4secs	Y	R	R	R	Y	R	R	R
S2	rst=0	16secs	G	R	R	R	G	R	R	R
S3	rst=0	4secs	Y	R	R	R	Y	R	R	R
S4	rst=0	8secs	R	G	R	R	R	G	R	R
S5	rst=0	4secs	R	R	Y	R	R	R	Y	R
S6	rst=0	16secs	R	R	G	R	R	R	G	R
S7	rst=0	4secs	R	R	Y	R	R	R	Y	R
S8	rst=0	8secs	R	R	R	G	R	R	R	G

TABLE 1: STATE TRANSITION

IV.IMPLEMENTATION OF SEVEN SEGMENT LED:

In our project we have focused to display the Timer on the seven segment for which our output signals are switching. As our proposed system is generating binary outputs at the end so we require a seven bit decoder to generate specific bits signal to generate the desired output which is to be applied as an input to the seven segment. So, we have a maximum 30 sec timer to display it we require two seven segments LEDs to represent it. Here is the diagram of seven segment given below as figure 4

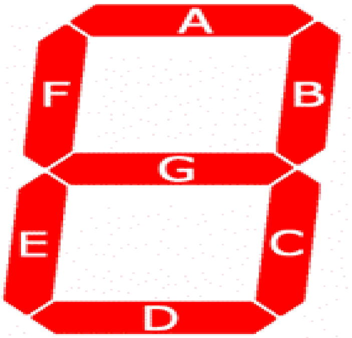


FIGURE 8: SEVEN SEGMENT LED

Here we can see seven segment denoted as A ,B ,C, D, E, F and G each segment is set of LED's and anode of these are set to be High always to make a particular segment active we have to make its cathode to be low. For example we want to display 3 number so we have to make cathode of the F and E to be HIGH and kept others at low voltage for displaying the number. We have denoted each segment cathode as CA, CB, CC, CD, CE, CF and CG respectively. The detailed description is given in the following table.

No	CA	CB	CC	CD	CE	CF	CG	Cathode
0	L	L	L	L	L	L	H	7'b0000001
1	H	L	L	H	H	H	H	7'b1001111
2	L	L	H	L	L	H	L	7'b0010010
3	L	L	L	L	H	H	L	7'b0000110
4	H	L	L	H	H	L	L	7'b1001100
5	L	H	L	L	H	L	L	7'b0100100
6	L	H	L	L	L	L	L	7'b0100000
7	L	L	L	H	H	H	H	7'b0001111
8	L	L	L	L	L	L	L	7'b0000000
9	L	L	L	L	H	L	L	7'b0000100

TABLE 2: INPUT TABLE FOR SEVEN SEGMENT

In the above table the L indicate Low and H indicates The High values.

Suitable connection of our Traffic light controller to seven segment LED's are given below

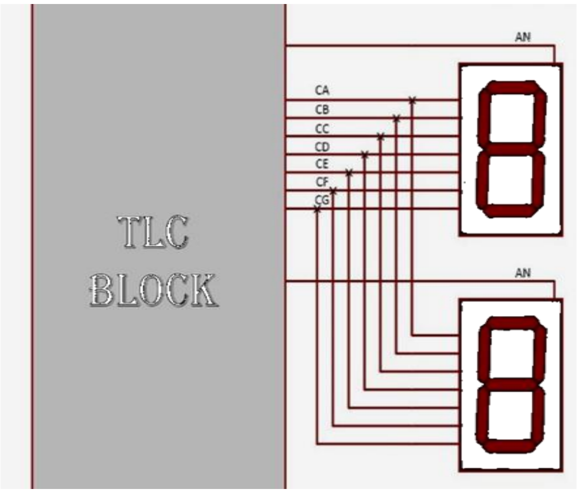


FIGURE 9: INTERFACING OF SEVEN SEGMENT LED

This diagram shows the output from TLC block which comes out of decoder to enable the segment to display the timing of the circuit. Here the CA- CG are the cathodes input to the seven segment and AN signal is anode input to the seven segment.

V.MAIN COMPONENTS:

4.1 FPGA:

Field Programmable Gate Arrays (FPGAs) are the semiconductor devices that are made up of table of the configurable logic blocks (CLBs) which are banded together through the programmable interconnects. The FPGA contains an array of programmable logic blocks, and the re-configurable interconnections that will allow the blocks and logic gates to get them connected through wire altogether, to accomplish the complex combinational functions, or only some simple logic gates functions. In most of the FPGAs, the logic blocks can also include memory element, which perhaps be simple flip-flops or can be additional complete blocks of memory. FPGAs are reprogrammable devices that can be reprogrammed to a suitable and needed application or functionality prerequisite after fabrication.

Field Programmable Gate Arrays (FPGAs) are widely used in quick prototyping and verification of theoretical design and as well as useful in electronic systems. The primary use of the FPGA's is to avoid the high expenses for the custom VLSI projects such as ASIC for a small quantity. A design flow is given in the following block diagram

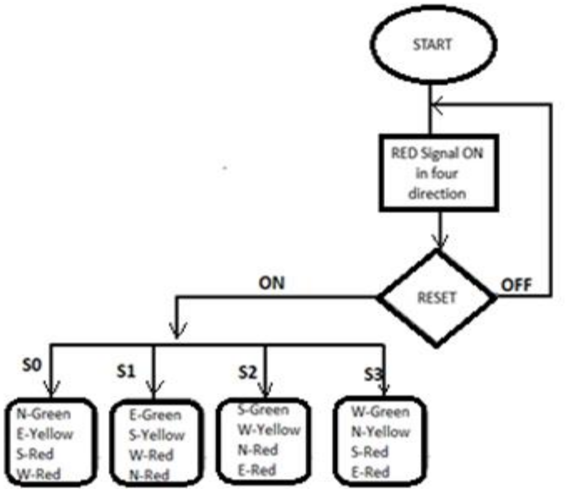


FIGURE 10: FPGA FLOW

4.2 VERILOG (VHDL):

TLC is sequential circuit and coding method we choose is FSM (Finite State Machine). As specific we decide different states for the machine and provide respecting timing in between states to get the desired output.

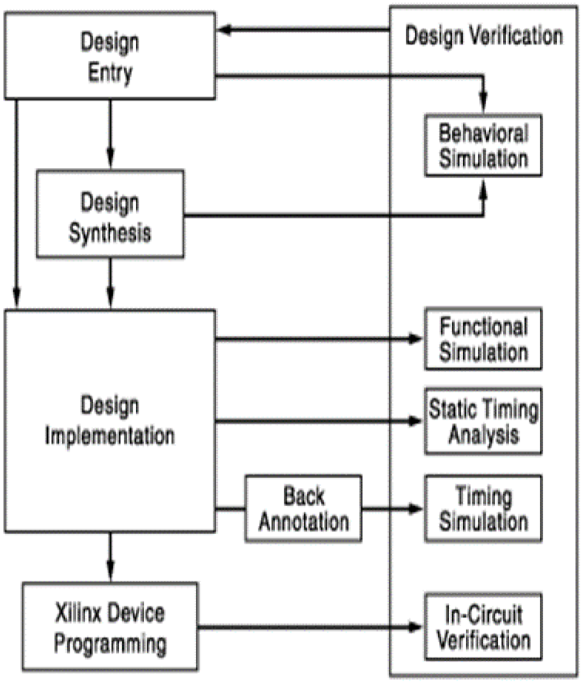
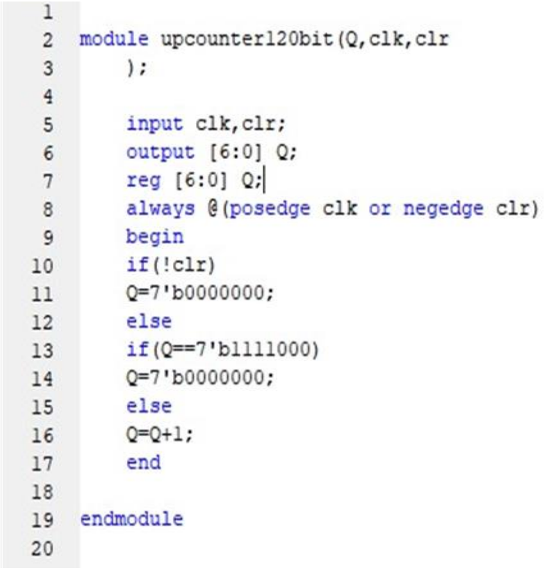


FIGURE 11: CODE LOGIC

4.2.1 UP-COUNTER 120



```
module upcounter120bit(Q,clk,clr
    );

    input clk,clr;
    output [6:0] Q;
    reg [6:0] Q;
    always @(posedge clk or negedge clr)
    begin
    if(!clr)
    Q=7'b0000000;
    else
    if(Q==7'b1111000)
    Q=7'b0000000;
    else
    Q=Q+1;
    end

endmodule
```

FIGURE 11: VHDL CODE UPCOUNTER 120

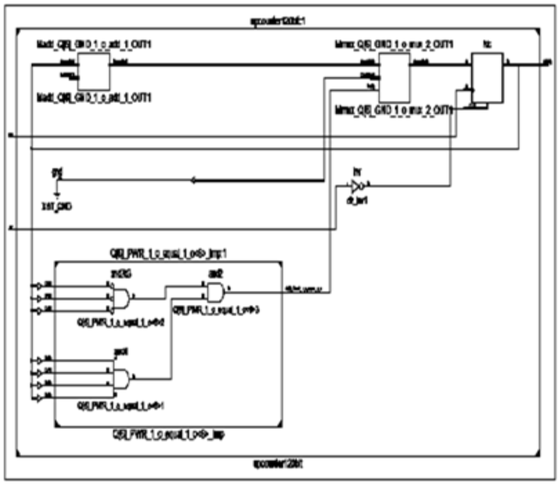


FIGURE 12: RTL VIEW

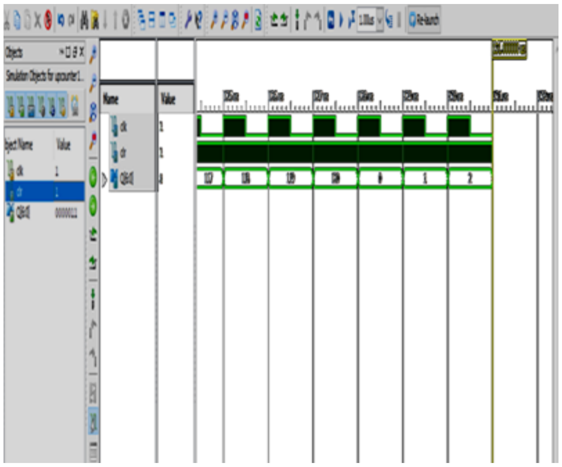
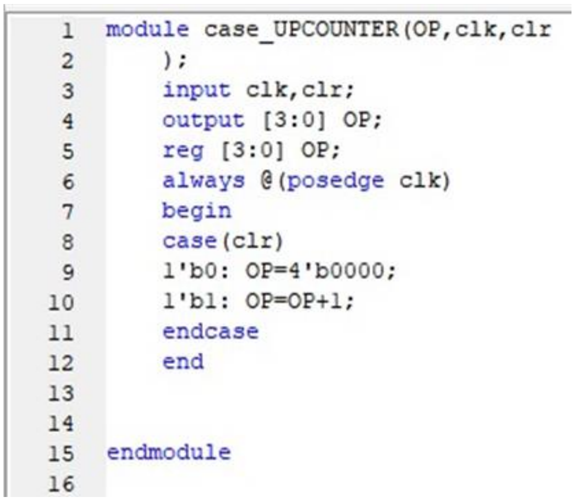


FIGURE 13: SIMULATION

4.2.2 CASE-UPCOUNTER



```
module case_UPCOUNTER(OP,clk,clr
    );
    input clk,clr;
    output [3:0] OP;
    reg [3:0] OP;
    always @(posedge clk)
    begin
    case(clr)
    1'b0: OP=4'b0000;
    1'b1: OP=OP+1;
    endcase
    end


endmodule
```

FIGURE 14: VHDL CODE CASE UPCOUNTER

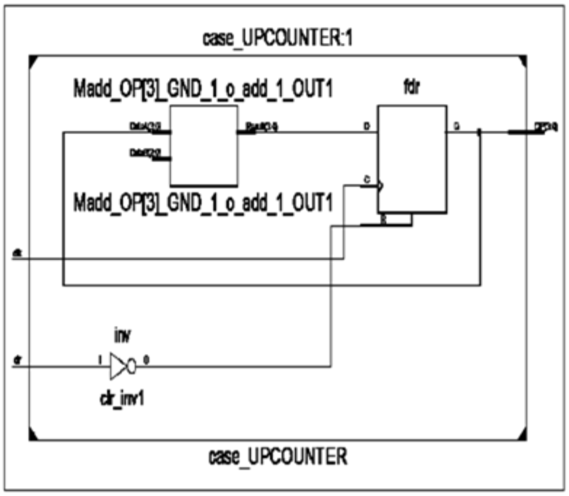


FIGURE 15: RTL VIEW

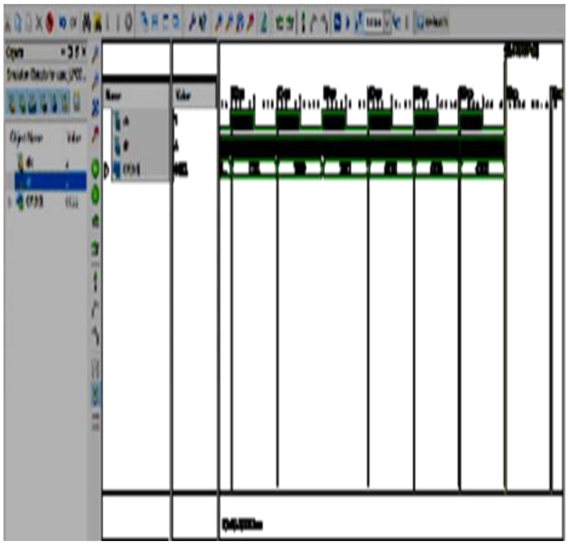
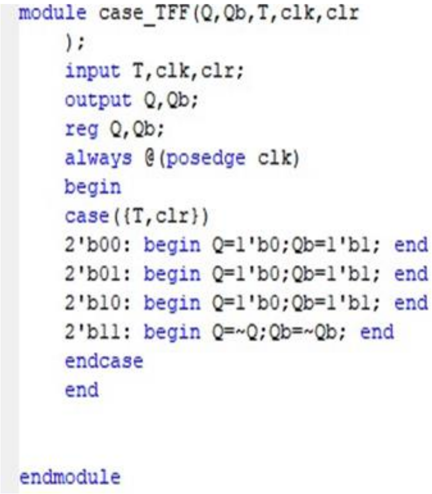


FIGURE 16: SIMULATION

4.2.3 T-FLIP FLOP



```
module case_TFF(Q,Qb,T,clk,clr
    );
    input T,clk,clr;
    output Q,Qb;
    reg Q,Qb;
    always @(posedge clk)
    begin
    case({T,clr})
    2'b00: begin Q=1'b0;Qb=1'b1; end
    2'b01: begin Q=1'b0;Qb=1'b1; end
    2'b10: begin Q=1'b0;Qb=1'b1; end
    2'b11: begin Q=~Q;Qb=~Qb; end
    endcase
    end


endmodule
```

FIGURE 17: VHDL CODE T-FLIP FLOP

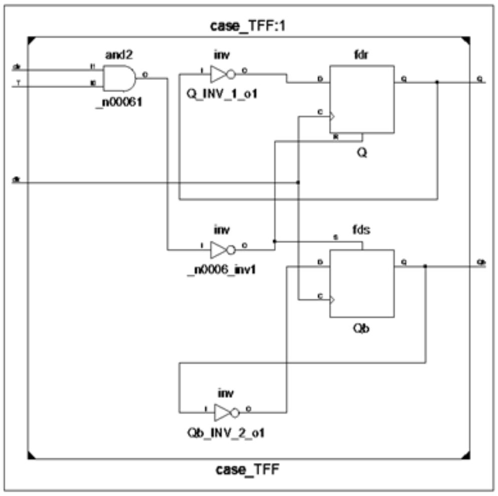


FIGURE 15: RTL VIEW

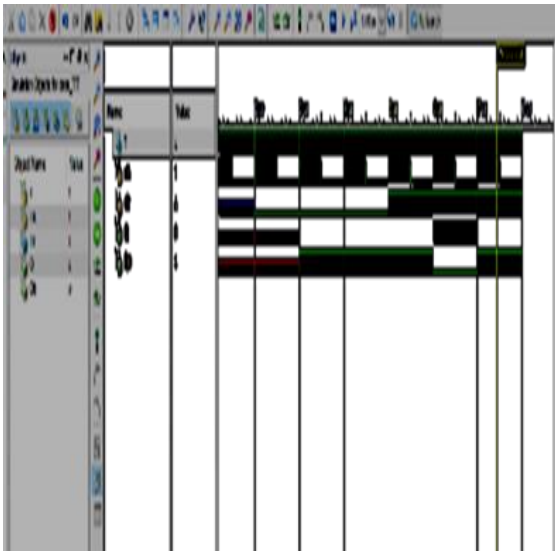
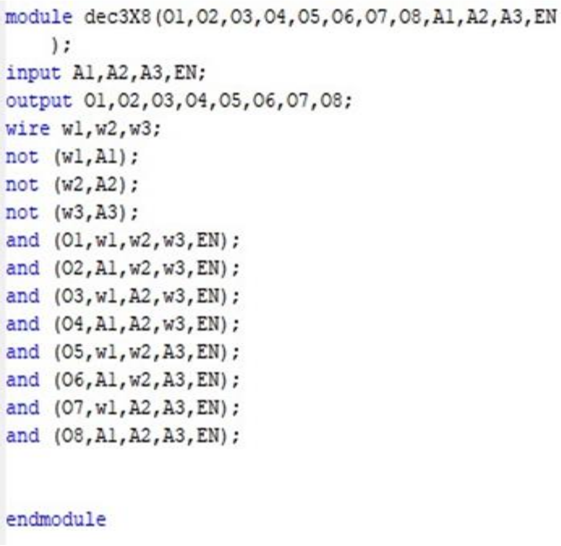


FIGURE 16: SIMULATION

4.2.4 DECODER 3X8



```
module dec3X8(O1,O2,O3,O4,O5,O6,O7,O8,A1,A2,A3,EN
    );
input A1,A2,A3,EN;
output O1,O2,O3,O4,O5,O6,O7,O8;
wire w1,w2,w3;
not (w1,A1);
not (w2,A2);
not (w3,A3);
and (O1,w1,w2,w3,EN);
and (O2,A1,w2,w3,EN);
and (O3,w1,A2,w3,EN);
and (O4,A1,A2,w3,EN);
and (O5,w1,w2,A3,EN);
and (O6,A1,w2,A3,EN);
and (O7,w1,A2,A3,EN);
and (O8,A1,A2,A3,EN);


endmodule
```

FIGURE 17: VHDL CODE

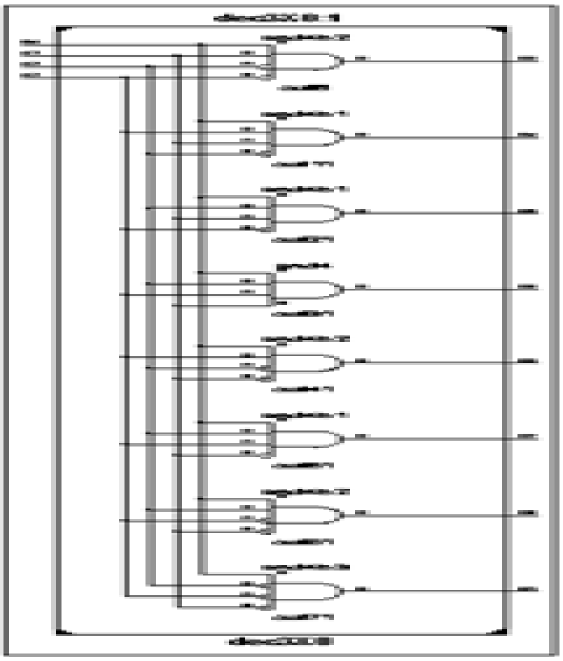


FIGURE 18: RTL VIEW

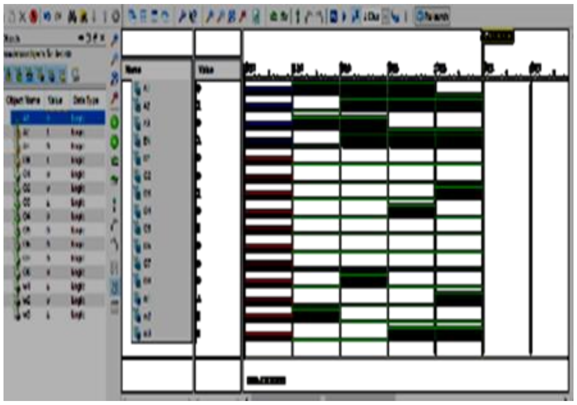
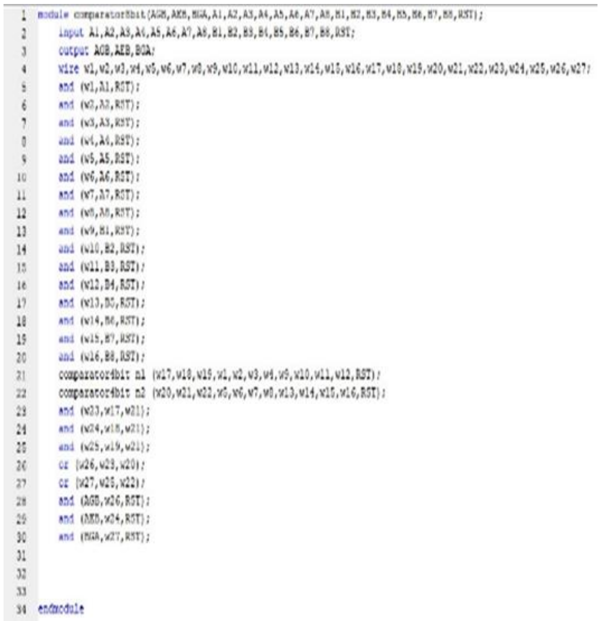


FIGURE 19: SIMULATION

4.2.5 COMAPARATOR



```
module comparator8bit(AGB,AEB,BGA,A1,A2,A3,A4,A5,A6,A7,A8,B1,B2,B3,B4,B5,B6,B7,B8,RST);
    input A1,A2,A3,A4,A5,A6,A7,A8,B1,B2,B3,B4,B5,B6,B7,B8,RST;
    output AGB,AEB,BGA;
    wire w1,w2,w3,w4,w5,w6,w7,w8,w9,w10,w11,w12,w13,w14,w15,w16,w17,w18,w19,w20,w21,w22,w23,w24,w25,w26,w27;
    and (w1,A1,RST);
    and (w2,A2,RST);
    and (w3,A3,RST);
    and (w4,A4,RST);
    and (w5,A5,RST);
    and (w6,A6,RST);
    and (w7,A7,RST);
    and (w8,A8,RST);
    and (w9,B1,RST);
    and (w10,B2,RST);
    and (w11,B3,RST);
    and (w12,B4,RST);
    and (w13,B5,RST);
    and (w14,B6,RST);
    and (w15,B7,RST);
    and (w16,B8,RST);
    comparator4bit m1 (w17,w18,w19,w1,w2,w3,w4,w9,w10,w11,w12,RST);
    comparator4bit m2 (w20,w21,w22,w5,w6,w7,w8,w13,w14,w15,w16,RST);
    and (w23,w17,w21);
    and (w24,w18,w21);
    and (w25,w19,w21);
    or (w26,w23,w20);
    or (w27,w25,w22);
    and (AGB,w26,RST);
    and (AEB,w24,RST);
    and (BGA,w27,RST);



endmodule
```

FIGURE 20: VHDL CODE

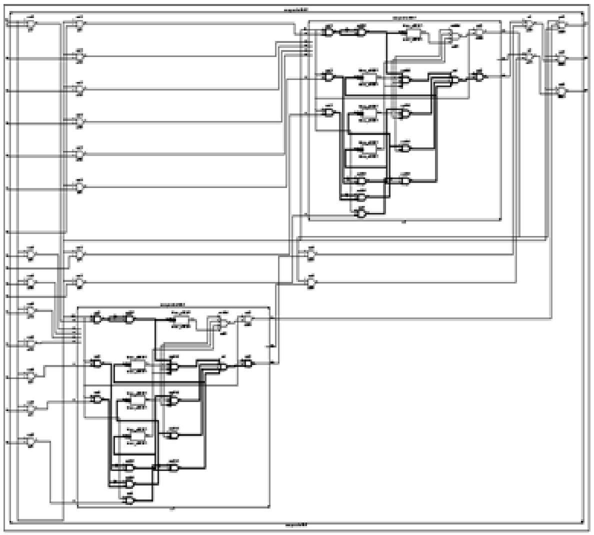


FIGURE 21: RTL VIEW

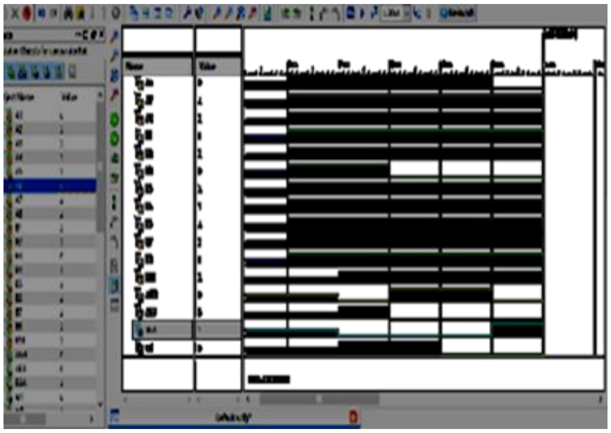


FIGURE 22: SIMULATION

VI.CONCLUSION

In this paper, a digital designing of schematic which improve our basic concept and Finite State Machine (FSM) is used to implement an intelligent and a efficient traffic light controller. This design can be further extended by FPGA implementation on Hyper Terminal, LCD and VGA. This proposed project takes care of traffic on any intersection having four roads. This system reduces waiting time for vehicles at junctions. It also reduces traffic on roads which increases due to the long waiting time.

VII.ACKNOWLEDGEMENT:

The authors would like to thank Dr. RK Yadav sir, Head of the Department, Electronics and Communication, for giving this opportunity to them. They would also like to thank Dr. Pavan Shukla sir and Mr. Sandeep Bhatia sir for their constant support & thoughtful discussions during this project.

REFERENCES

[1] M. Morris Mano, Michael D. Ciletti: Digital Design (6th edition) (18 May 2018).

[2] Design and implementation of smart traffic light controller using VHDL language Noor Gadawe, Sahar Qaddoori (December 2019).

[3] A. Raza, A. Kumar, E. Chaudhary, "Traffic Light Controller Using VHDL", International Journal of Modern Trends in Engineering and Research (IJMTER), Vol. 04, Issue 4, PP. 57-62, April 2017.

[4] Mohammed Fayaz, Pooja K, Pranitha P Reddy, Swathi T, Density based Traffic Control System with Ambulance Detection, IJERT, ISSN: 2278-0181, RTESIT – 2019 (VOLUME 7 – ISSUE 08).

[5] Chan, S.S., Deshpande, R.S. & Rana, J.G. (2009) Design of an Intelligent Traffic Light Controller Using Embedded System.

[6] Apoorva S N, Simran R Karthik, Rakesh M B," Traffic Light Controller based on FPGA".

[7] Shilpa U. Holambe, D. B. Andore, 'Advanced Traffic Light Controller System Based on the FPGA'.

[8] Verilog HDL: A Guide to Digital Design and Synthesize by Samir Palnitkar.

[9] P.G. Prasad, S.V. Krishna, "Design and Implementation of ITLC System Using FPGA", Journal of Electronics and Communication Engineering (IOSR-JECE), Vol. 12, Issue 5, Ver. II, PP. 07-13, Sep.- Oct. 2017

[10] Sajid M. Sheikh, Lebone Powder & Ibo Ngebani, a smart microprocessor-based four way stop road traffic controller, University of Botswana, Department of Electrical Engineering, Gaborone, Botswana. International Journal of Electrical and Electronics Engineering (IJEEE) ISSN(P): 2278-9944; ISSN(E): 2278-9952 Vol. 7, Issue 4, Jun – Jul 2018, 9-22

[11] Y. N. Udoakah, and I. G. Okure, dept. of electrical/electronic & computer engr', university of uyo, uyo, akwa ibom state. Nigeria. Design and implementation of a density-based traffic light control with surveillance system, Vol. 36, No. 4, October 2017, pp. 1239 – 124. Print ISSN: 0331-8443, Electronic ISSN: 2467-8821.

[12] Michael Osigbemeh, Michael Onuu, Olumuyiwa Asaolu, Design and development of an improved traffic light control system using hybrid lighting system. journal of traffic and transportation engineering (english edition) 2017

[13] Book: Verilog HDL: A Guide to Digital Design and Synthesize by Samir Palnitkar.

[14] D.Bhavana "Traffic Light Controller Using FPGA" Int. Journal of Engineering Research and Applications ISSN: 2248-9622, Vol. 5, Issue 4, (Part -6) April 2015

[15] "Design of FPGA Based Intelligence Traffic Light Controller", Sourav Nath, Suman Sau, Abhishek Roy ,2012 Conference on Radar, Communication & Computing, SKP College, Tiruvannamalai, TN, December.

[16] "FPGA Implementation of an Advanced Traffic Light Controller using Verilog HDL", Dilip, Y. Alekhya, P. Divya Bharathi, Advanced Research in Computer Engineering & Technology; Volume 1, Issue 7, pp: 2278 – 1323,2012.

[17] "FPGA-Based Advanced Real Traffic Light Controller System Design". El-Medany, W.M.; Univ. Of Bahrain Sakhir Hussain, M.R.DOI:10.1109/IDAACS.2007.4488383 Publisher: IEEE

[18] "Structural modeling of intelligent traffic light controller on fpga", surabhi s.gaopande, ashwini d. Bharade. Department of Electronics

Engineering, Visvesvaraya National Institute of Technology, India.

[19] Machines, M. Ali Qureshi, Abdul Aziz, and S. Hammad Raza," A Verilog Model of Adaptable Traffic Control System Using Mealy State", International Journal of Computer and Electrical Engineering, Vol. 4, No. 3, June 2012.

[20] Hao Dong, Xingguo Xiong and Xuan Zhang, "Design and Implementation of a Real-time Traffic Light Control System Based on FPGA", ASEE 2014 Zone I Conference, University of Bridgeport, Bridgpeort, CT, USA, April 3-5, 2014.Finite State Machines, retrieved March 21,2017.