

Traffic Light Controller based on FPGA

Apoorva S N¹, Simran R Karthik², Rakesh M B³

^{1,2}student, Dept. of Telecommunication, SIT, Tumkur, Karnataka, India

³Asst. Professor, Dept. of Telecommunication, SIT, Tumkur, Karnataka, India

Abstract - Traffic light control systems are widely used to monitor and control the movement of automobiles at the junction of several roads. Traffic signals are essential to guarantee safe driving at road intersections and aim to realize smooth movement of vehicles. But there is an inherent problem in the current traffic signal system. There is a lack of switching on and off of traffic signals based on the number of vehicles on the roads which results in long waiting times for vehicles on busy routes as there is uniform waiting time for all routes. So an approach is taken where different routes at the junction will have different waiting times. This project work tries to achieve the different waiting times based on the assumption that some routes have lesser traffic than other. Here the functional simulation is achieved through Verilog programming language and implemented on Xilinx 14.7. The switching of different traffic lights through several states is achieved through Moore state diagram. The design is implemented on Spartan 6 family. The implementation results show that duration for which green light is on is different for different routes. Thus the traffic congestion can be reduced as the vehicles on the route with the greater vehicle density will not have to wait for a longer duration for the routes with lesser vehicle density.

Key Words: Traffic light, Verilog, FPGA, Moore.

1. INTRODUCTION

Traffic lights have been utilized to schedule and control the competing traffic flows at each road intersection using light cycle schedules. The traffic light sequence works on the specific switching of Red, Green and Yellow lights in a particular way with stipulated time form. They provide safe scheduling that allows all traffic flows to share the road intersection. The constant queuing delay at each road intersection decreases the traffic flow fluency and then decreases the traffic efficiency all over the road network.

Conventional traffic control systems has a major drawback: Due to lack of adjustments in timing of traffic signals, the traffic has to wait for long duration on the lane with few vehicles, while on same lane the traffic cannot pass through in short time due to rush on lane. So, there is a need to develop a secure, fast and reliable traffic control system capable to control the vehicular traffic in rush hours without a need of traffic sergeant.

Here, a real traffic control system using Moore state machine has been developed. The design is simulated and

implemented on Spartan-6 XC6SLX9 FPGA development kit. The design of adaptable traffic control system is carried out for a junction consisting of four roads. To some extent, traffic jam or unreasonably latency time of stoppage of vehicles is solved. An efficient traffic control system is designed using Moore finite state machines.

1.1 Road Structure

Generally, a traffic signal system has three lights. A green light on the bottom of the signal indicates the traffic to proceed, a yellow light in the middle warns the traffic to slow and prepare to stop, and red light on the top indicates the traffic to stop. Figure-1 shows structure of any junction consisting of four main roads and each road is divided into two main roads (straight and cross). Eight traffic signals L1, L2, ..., L8 have been used. The signals on straight roads are L1, L3, L5 and L7, while L2, L4, L6 and L8 are traffic signals on cross roads. Traffic signals on straight roads have four lights: one red, one yellow and two green, while signals on cross roads have two lights: red and green.

Based on the assumption made L1, L3, L5 and L7 have denser traffic compared to L2, L4, L6 and L8. In L1, L3, L5 and L7 lanes, ON timing of green light will be doubled.

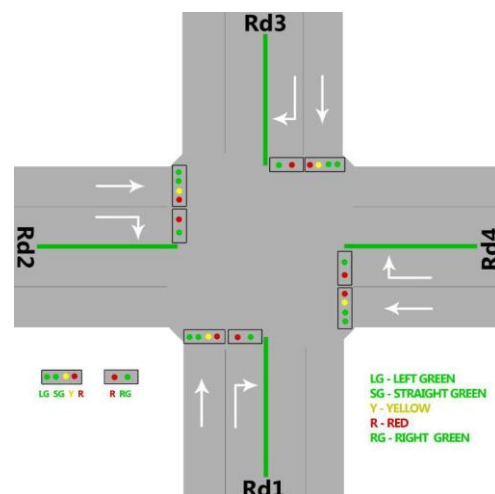


Fig. 1: Structure of junction consisting of four main roads

1.2 Moore FSM

A general model of a Moore sequential machine is shown in Figure-2. Its output is generated from the state register block. The next state is determined using the present (current) input and the present (current) state [5]. Here the state register is also modeled using D flip-flops.

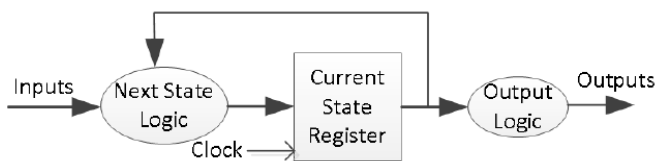


Fig.2:[5]Block diagram of Moore FSM

2. STATE TABLE

The timing states of traffic lights are shown in Table 1. The timing of signal lights can be increased or decreased according to the traffic density on the roads.

TABLE 1: TIMING STATES OF TRAFFIC LIGHTS

STATE	INPUT	TIME DURATION	L1	L2	L3	L4	L5	L6	L7	L8
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

2.1 State Diagram

The graphical representation of the sequences of traffic lights with the help of state diagram is shown in Fig. 3.

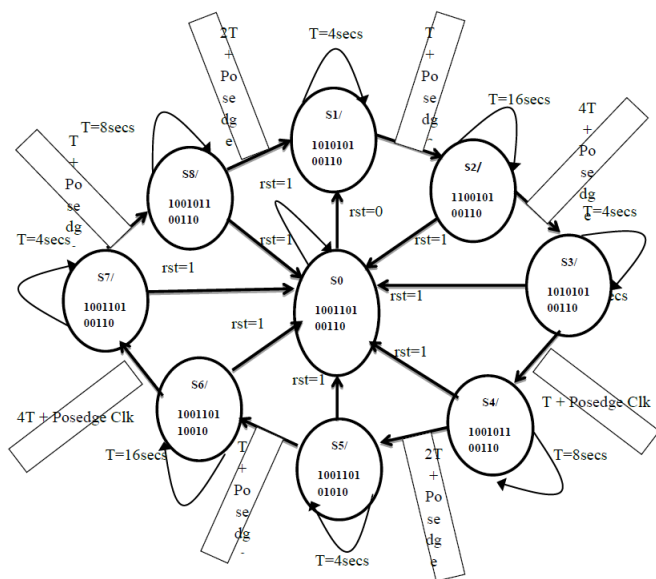


Fig.3 : State diagram for four road junction

3.ALGORITHM

- Step 1: Start
- Step 2: Input variables declared clk,rst Output variables declared [4:0]L15, [1:0]L26, [4:0]L37, [1:0]L48
- Step 3: Initially before resetting the TLC, red lights of all traffic signals L1-L8 are ON.
- Step 4: After resetting the TLC, yellow light of signal L1 & L5 are ON and red lights on the remaining signals are ON.
- Step 5: After a delay of four seconds, green lights of traffic signals L1 & L5 are ON while red light on the remaining signals are ON.
- Step 6: After sixteen seconds, yellow light of signals L1 & L5 are again ON while red light on the remaining signals are ON.
- Step 7: After four seconds, red light of signals L1 & L5 are ON and green light on L2 and L6 are ON.
- Step 8: After eight seconds, red light of signals L2 & L6 are ON and yellow light on L3 and L7 are ON.
- Step 9: After four seconds, green light of signals L3 & L7 are ON while red lights on the remaining signals are ON.
- Step 10: After sixteen seconds, yellow light of the signals L3 & L7 are ON and red lights on the remaining signals are ON.
- Step 11: After four seconds, green light on the signals L4 and L8 are ON and red lights on the remaining signals are ON.
- Step 12: After eight seconds, yellow light of signal L1 and L5 is ON and red lights on the remaining signals are ON.
- Step 13: The sequence repeats until the reset of TLC is enabled.
- Step 14: End

4. RESULTS & DISCUSSION

The design is simulated with Xilinx ISE Simulator to verify that the design behaves as expected in the target device both in terms of functionality and timing. Figure-4 shows the timing waveform of the design obtained with Xilinx ISE Simulator.

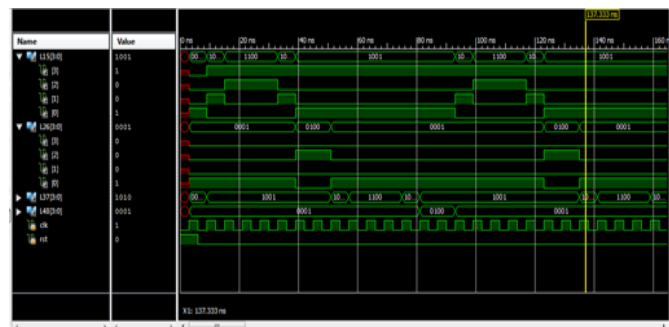


Fig.4 : Timing waveform1

From Figure-4 when rst=1, L15,L26,L37 and L48 are at 4'b0001.

When rst=0:

- L15,L26,L37 and L48 are at 4'b0001 for 4 clock pulses.
- L15 will be 4'b1010 for next 4 clock pulses, L37 will be 4'b1001 and L26,L48 will be 2'b10
- L15 will be 4'b1100 for next 16 clock pulses, L37 will be 4'b1001 and L26,L48 will be 2'b10
- L26 will be 2'b01 for next 8 clock pulses, L15 and L37 will be 4'b1001 and L48 will be 2'b10
- L37 will be 4'b1010 for next 4 clock pulses, L15 will be 4'b1001 and L26,L48 will be 2'b10
- L37 will be 4'b1100 for next 16 clock pulses, L15 will be 4'b1001 and L26,L48 will be 2'b10
- L48 will be 2'b01 for next 8 clock pulses, L15 and L37 will be 4'b1001 and L26 will be 2'b10



Fig.6: Timing waveform

From Figure 6

- L15 will be 4'b1010 for next 4 clock pulses, L37 will be 4'b1001 and L26,L48 will be 2'b10
- L15 will be 4'b1100 for next 16 clock pulses, L37 will be 4'b1001 and L26,L48 will be 2'b10
- L26 will be 2'b01 for next 8 clock pulses, L15 and L37 will be 4'b1001 and L48 will be 2'b10
- L37 will be 4'b1010 for next 4 clock pulses, L15 will be 4'b1001 and L26,L48 will be 2'b10
- L37 will be 4'b1100 for next 16 clock pulses, L15 will be 4'b1001 and L26,L48 will be 2'b10
- L48 will be 2'b01 for next 8 clock pulses, L15 and L37 will be 4'b1001 and L26 will be 2'b10



Fig.5 : Timing waveform2

From Figure-5

- L15 will be 4'b1010 for next 4 clock pulses, L37 will be 4'b1001 and L26,L48 will be 2'b10
- L15 will be 4'b1100 for next 16 clock pulses, L37 will be 4'b1001 and L26,L48 will be 2'b10
- L26 will be 2'b01 for next 8 clock pulses, L15 and L37 will be 4'b1001 and L48 will be 2'b10
- L37 will be 4'b1010 for next 4 clock pulses, L15 will be 4'b1001 and L26,L48 will be 2'b10
- L37 will be 4'b1100 for next 16 clock pulses, L15 will be 4'b1001 and L26,L48 will be 2'b10
- L48 will be 2'b01 for next 8 clock pulses, L15 and L37 will be 4'b1001 and L26 will be 2'b10

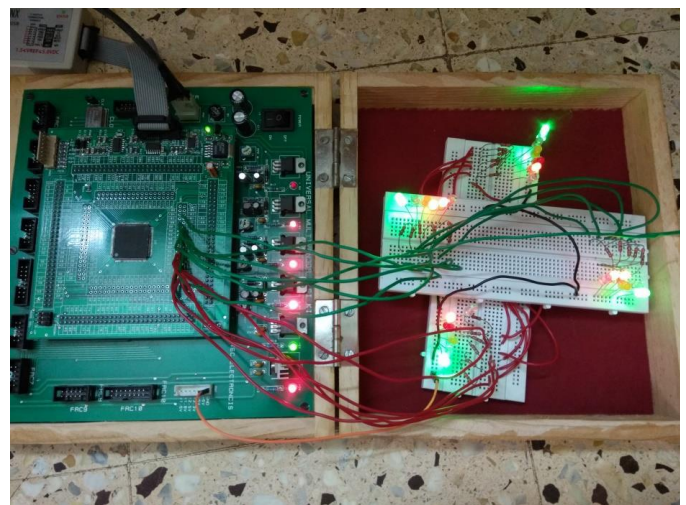


Fig.7 : Blinking of Led's

Led's are interfaced to FPGA kit where following design properties are selected:

family: Spartan 6

device: XC6SLX9

package: TQG144

Led's getting turned on based on the traffic waiting time is shown in Figure-7.

5. CONCLUSIONS

In this project, a Moore finite state machine (FSM) is used to design an efficient and intelligent traffic light controller. The language used for the implementation is Verilog. The design is implemented on *Spartan-6*, *XC6SLX9* FPGA development kit. This model takes care of traffic on any junction consisting of four roads. The system helps to reduce the unnecessary waiting period for vehicles at junctions. It also reduces the density of vehicles on roads which increases due to the long waiting period.

REFERENCES

- [1] 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.
- [2] Hao Dong, Xingguo Xiong and Xuan Zhang, "Design and Implementation of a Real-time Traffic Light Control System Based on FPG", ASEE 2014 Zone I Conference, University of Bridgeport, Bridgeport, CT, USA, April 3-5, 2014.
- [3] D.Bhavana, D.Ravi Tej, Priyanshi Jain, G.Mounika and R.Mohini .Bhavana, "Traffic Light Controller Using Fpga", D.Bhavana et al. Int. Journal of Engineering Research and Applications ISSN : 2248-9622, Vol. 5, Issue 4, (Part -6), pp.165-168, April 2015.
- [4] B. Dilip, Y. Alekhya and P. Divya Bharathi, "FPGA Implementation of an Advanced Traffic Light Controller using Verilog HDL", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 1, Issue 7, September 2012.
- [5] Finite State Machines, retrieved March 21, 2017, from <https://www.xilinx.com>