FPGA Implementation of VHDL Based Traffic Light Controller System

Sahil Gupta[1], Surbhi Sharma[2]

[1, 2]Department of Electronics & Communication Engineering, MIET, Jammu, J&K, India

Email address: [1]sahilgupta3@yahoo.in, [2]surbhi.ece@mietjammu.in

Abstract**—**Vehicular traffic at intersecting streets is typically controlled by traffic control lights. The function of traffic lights requires sophisticated control and coordination to ensure that traffic moves as smoothly and safely as possible. In recent times the electro-mechanical controllers are replaced by electronic circuits. The accuracy & fault tolerant drive towards electronic circuits. The idea is developed to meet the requirements of solid state traffic light controller by adopting FPGA (field programmable gate array) board and VHDL (VHSIC hardware description language) as the main controlling element, and LED'S (light emitting diodes) as the indication of light. VHSIC stands for Very High Speed Integrated Circuit. An FPGA is interfaced to LED'S provide for centralized control of the traffic signals. FPGA is programmed in such a way to adjust their timing and phasing to meet changing traffic conditions. The circuit besides being reliable and compact is also cost effective.

Keywords**—** Programmable logic controllers (PLC); counters; field-programmable gate array (FPGA); LEDs; xilinx.

I. INTRODUCTION

Traffic congestion is a severe problem in many modern cities around the world. Traffic congestion has been causing many critical problems and challenges in the major and most populated cities. To travel to different places within the city is becoming more difficult for the travellers in traffic. Due to these congestion problems, people lose time, miss opportunities, and get frustrated. Due to traffic congestions there is a loss in productivity from workers, trade opportunities are lost, delivery gets delayed, and thereby the costs goes on increasing. To solve these congestion problems, it is required to build new facilities & infrastructure but at the same time make it smart. The only disadvantage of making new roads on facilities is that it makes the surroundings more congested. So for that reason there is a need to change the system rather than making new infrastructure twice. Therefore many countries are working to manage their existing transportation systems to improve mobility, safety and traffic flows in order to reduce the demand of vehicle use. It can be implemented using simple electronic components such as LED as traffic light indicator and a FPGA for auto change of signal after a pre-specified time interval. FPGA is the brain of the system which initiates the traffic signal at a junction. The LED'S are automatically on and off by making the corresponding port pin high. At a particular instant only one green light holds and other lights hold at red. During transition from green to red, the present group yellow led and succeeding group yellow LED glows and then succeeding group LED changes to green. This process continues as a cycle [1], [2].

II. SYSTEM IMPLEMENTATION

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 crossing consisting of four main roads and each road is divided into two main roads (straight and cross). Eight traffic signals L1, L2 …L8 are used to design the system. There are four sensors on roads SW1, SW2, SW3 and SW4 which will be on the speed breaker of every lane. SW1, SW2, SW3 and SW4 sensors switches are linked with traffic signals on (L1, L6), (L2, L5), (L3, L8) and (L4, L7) respectively. Whenever any one of the sensors output is enabled, appropriate traffic starts to continue on the roads according to the position and priority of the switches and rest of the signals are off.

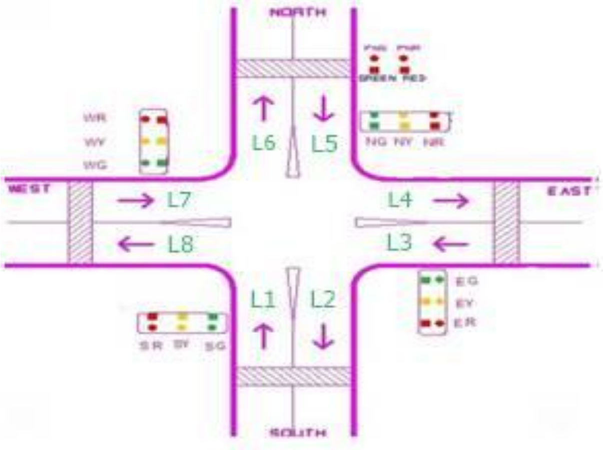


Fig. 1. Structure of any four way crossing.

In order to implement the applications indicated, a certain level of intelligence is required in both the traffic light and the regulator. Traditional traffic control systems are unidirectional, from regulator to traffic lights, without any response from the status of the traffic lights. One strategy for optimum control and traffic management is the coordination of traffic lights to create green waves. Currently, there exist

Sahil Gupta and Surbhi Sharma, "FPGA implementation of VHDL based traffic light controller system", *Journal of Scientific and Technical Advancements*, Volume 1, Issue 3, pp. 151-155, 2015.

different strategies to calculate green waves [5]. The main purpose of these techniques is to reduce the number of stops and minimize the travel times in trips. Weight sensors and counters are used to control the traffic with ease. And send this information to TLC for control of the traffic system.

III. Modelling Overview

FPGA is an Integrated Circuit consisting of an array of uncommitted elements; interconnection between these elements is user-programmable. Using Random Access Memory, high density logic is provided. FPGA is advantageous compared to microcontroller in terms of number of I/O (input & output) ports and performance. FPGA, an inexpensive solution compared to ASIC design; is effective with respect to cost in the case of production of large number of units but for fabrication in small number of units it is always costly and time consuming. The Design flow of FPGA shown in figure 2 is used to implement the traffic light controller using FPGA. The circuit description can be done using HDLs, followed by the functional simulation and synthesis. The design flow is followed till the timing simulation and then the generated file is downloaded into the target device (FPGA). [9].

The flow chart shown in figure 3 illustrates the actions to be taken by the users. Initially, all RED signals are ON and after few seconds, GREEN of a signal light in one particular direction will be ON to allow traffic in straight, right and left (left also sometimes needed) paths . The yellow light is split into two phases as yellow signal1 (Y1) and yellow signal2 (Y2). Pedestrian will be “OFF” in yellow signal1 (Y1) and pedestrian will be “ON” in yellow signal2 (Y2) so as to allow the pedestrians to cross the road.

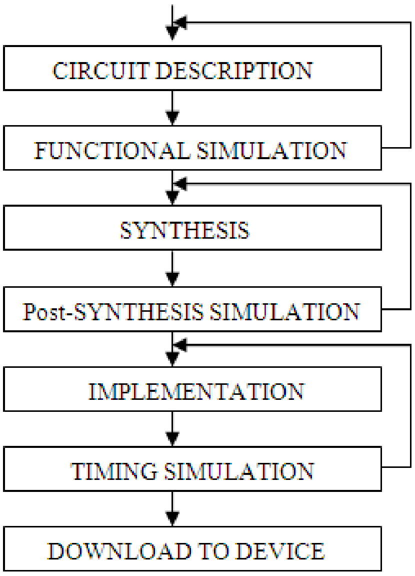


Fig. 2. Design flow to implement the traffic light controller using FPGA.

At first the North traffic will be allowed to move and then traffic in the East, South and West direction will be allowed to move in sequence. Program modifications in Traffic Light Controller can be easily per requirements can be done easily in the t i.e., suppose the traffic on main road should be allowed for more time and for side roads the traffic should be allowed for less time; then the clock is divided in such a way that for main road the clock period will be more and for side roads the clock period will be less, this is because the main road traffic is heavy when compared to the side road traffic [6]. In general TLC System will be having three lights (red, green and yellow) in each direction where red light stands for traffic to be stopped, green light stands for traffic to be allowed and yellow light stands for traffic is going to be stopped in few seconds. But yellow light is split into two phases and are included in the signalling lights along with red and green lights in order to indicate that in the first phase of yellow light, pedestrian will be OFF and in the second phase, pedestrian will be ON. The sequential order of the flow chart helps the programmer in the design regarding the flow of the program. North/ south-bound traffic will start with a green signal light while all the other lanes being red, the traffic will be stopped. After a predetermined time the north/south traffic light turns yellow and then to red, allowing the east/west signal light to be green and the same sequence as the north/south-bound traffic is followed.

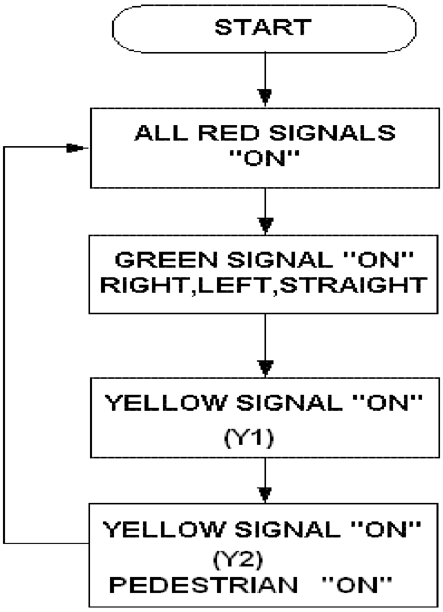

START
ALL RED SIGNALS "ON"
GREEN SIGNAL "ON" RIGHT,LEFT,STRAIGHT
YELLOW SIGNAL "ON" (Y1)
YELLOW SIGNAL "ON" (Y2) PEDESTRIAN "ON"

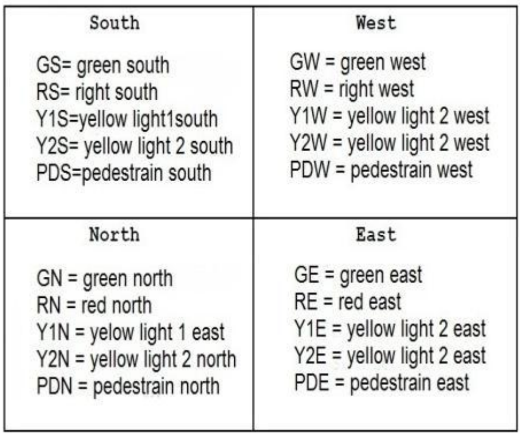
Fig. 3. Flow Chart for actions taken by certain road users.

The system will continue to be in this loop until an indication of a vehicle in a left turn lane occurs. When the signal light turns yellow, the controller scans the inputs. if high, then the program will jump to a subroutine which has a different light sequence. This sequence controls the main lights along with the left turn lights. After completion of the subroutine sequence, the program returns to the main loop. The flow chart can be applied to any number of road structures. a four road structure is considered in which the four directions labelled with four labels namely north, south, east and west. Each traffic lane has set of three traffic light signals, "red, yellow, and green", which operates similar to genera signalling lights i.e., it changes from red to green and then to yellow and after that back to red signal [4.]

IV. State Diagram

The TLC state diagram shown in Fig. 4 illustrates that whenever cnt=00 and dir=00,then green light in north direction will be ON for few seconds and red signal light in all other directions namely west, south and east will be ON when cnt=01 and dir=00 then yellow light (y1) will be ON for few seconds and when cnt=01 yellow light (y2) and pedestrian north will be ON and then dir is incremented by one and cnt is assigned to zero.

Table I. Abbreviations used in TLC state diagram.



South	West

GS= green south
RS= right south
Y1S=yellow light1south
Y2S= yellow light 2 south
PDS=pedestrain south
	GW = green west
	RW = right west
	Y1W = yellow light 2 west
	Y2W = yellow light 2 west
	PDW = pedestrain west

North	**East**

GN = green north
RN = red north
Y1N = yelow light 1 east
Y2N = yellow light 2 north
PDN = pedestrain north
	GE = green east
	RE = red east
	Y1E = yellow light 2 east
	Y2E = yellow light 2 east
	PDE = pedestrain east

So when cnt=00 and dir=01, the green light in east direction will be ON for few seconds and all red lights in other directions be ON. Whenever cnt=01 and dir=01 then yellow light (y1) will be ON for few seconds and when cnt=01 yellow light (y2) and pedestrian east will be ON and then dir is incremented by one and cnt is assigned to zero. So whenever cnt=00 and dir=10, the green light in south direction will be ON for few seconds and all red lights in other directions will be ON. Whenever cnt=01 and dir=10 then yellow light (y1) will be ON for few seconds and when cnt=01 yellow light (y2) and pedestrian south will be ON and then dir is incremented by one and cnt is assigned to zero. So whenever cnt=00 and dir=11, the green light in west direction will be ON for few seconds and all red lights in other directions will be ON. Whenever cnt=01 and dir=11 then yellow light (y1) will be ON for few seconds and when cnt=01 yellow light (y2) and pedestrian west will be ON and then dir is assigned to 00 and cnt is assigned to zero. This sequence repeats and the traffic flow will be controlled by assigning time periods in all the four directions. Table I specifies the abbreviations used in TLC state diagram. Labeling for each lane is done by assigning the direction label in order to distinguish the outputs from each other with their states. A traffic light controller program has two inputs namely clock and reset. When the two variables are "1" then the TLC will start working. Initially that is when reset is "0" then the red signal lights in all the directions will be ON and when reset is "1", then the traffic light controller system will be on assigning cnt and dir variables to "00" where cnt and dir respectively represent the states and the four directions in the state machine.

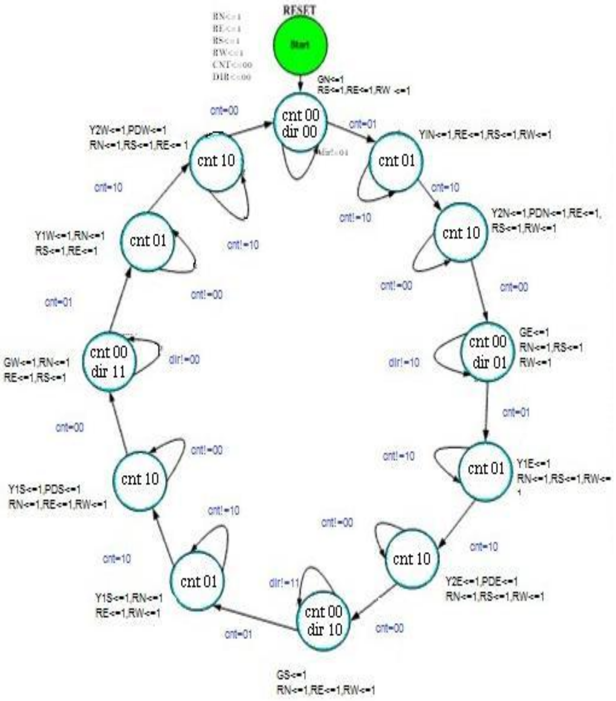


Fig. 4. State diagram.

Now by the use of sensors we can change the delay from one state to other i.e. indirectly controlling the time for which traffic light remain green or red. For instance, if there is congestion in traffic on the lane L1 than the timing for green light in lane L1 will be increased so as to pass the traffic through [9]. Similarly with other lanes up and down counter will count the number of vehicles and will tell the condition of the traffic. And the weight sensors will be diverging heavy and light vehicles to their respective lanes.

V. Implementation

The state machine is coded using the Hardware Description Language, VHDL. Spartan-3E trainer kit is shown in figure 5. Figure 6 shows the FPGA Implementation of TLC. Using Xilinx ISE tool, this code is dumped into Spartan-3E FPGA trainer kit. Weight sensors and signal lights at each lane have their set of traffic light signal "Red, Yellow, and Green". Operation of the signal light is similar to common traffic light signal. Along with these specifications, each lane has sensor of the corresponding road. The first sensor detects the presence of vehicles and its weight and the second sensor determines the volume of the traffic corresponding to that lane by counting incoming and outgoing vehicles. Through the two sensors, the expected time for green signal ON and when the signal light at each lane should be changed to green.

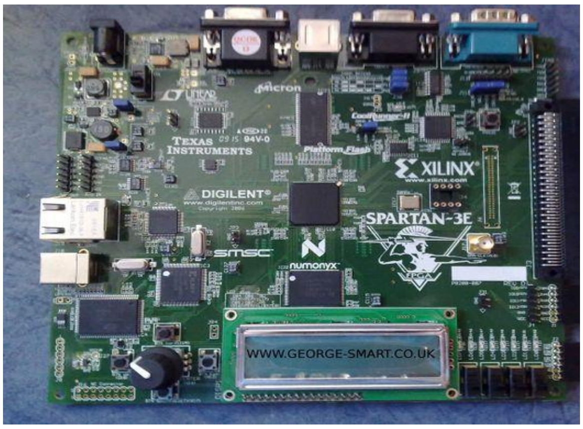


Fig. 5.Spartan-3E.

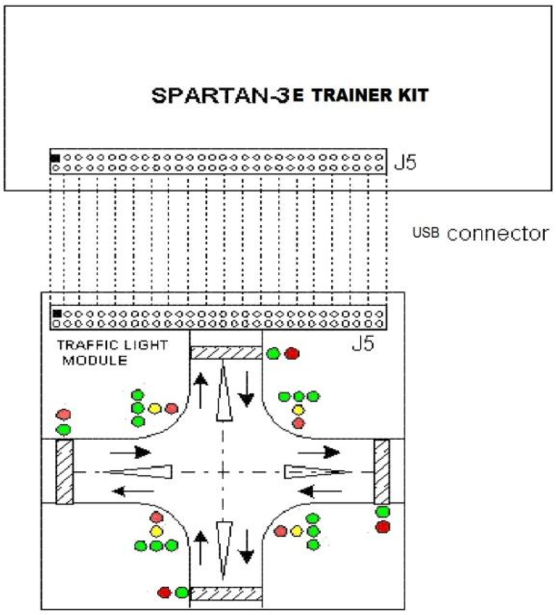


Fig. 6. FPGA implementation.

VI. Observations

As compare to other methods the implementation of traffic light system using FPGA is cost effective moreover the time delay can be changed simply by changing the software. Xilinx Spartan-3 FPGA are ideal for low-cost, high-volume applications and are targeted as replacements Compared to other structure description in VHDL language, state machine has program level bright, structure clear, easy to read and understand benefits. Using fixed point, parallel computational structures, FPGA provides computational speeds as much as 100 times greater than those possible with Digital Signal Processors (DSP).

VII. Timing Results

Timing Summary:
Speed Grade: -4
Minimum period: 10.102ns
Maximum Frequency: 98.990MHz
Maximum output required time after clock: 10.512ns

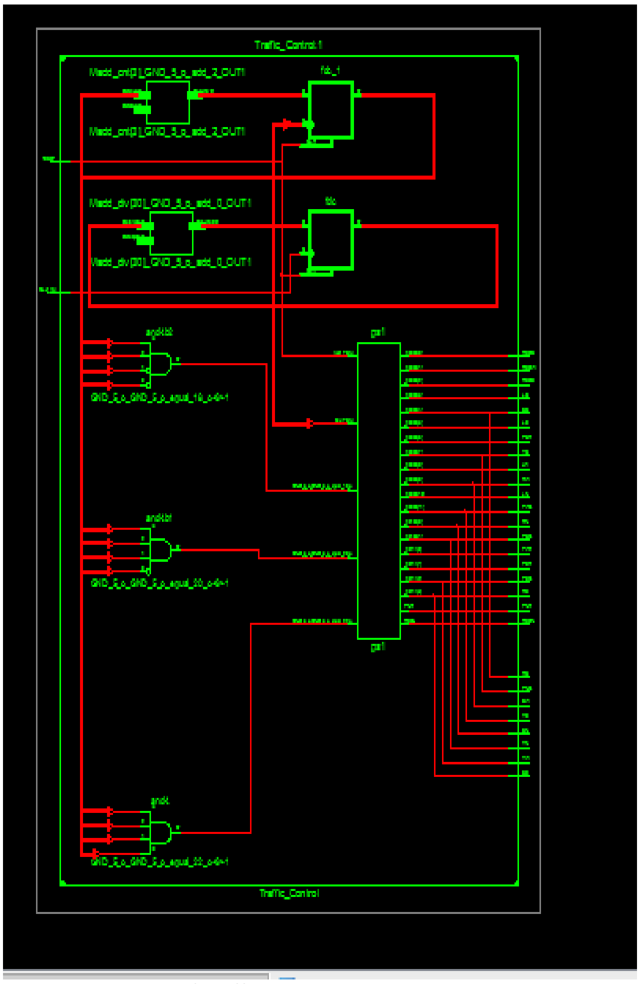


Fig.7. RTL Schematic of Traffic Light Controller

VIII. Conclusion

The modern ways of multi-way traffic management improves the traffic condition up to a large extent. These more complex controllers can be well handled using states machines. Methods to reduce the states in the state machine also help in reducing the required hardware thus leading to low power and area efficient design. In addition to the general procedure the ChipScope Pro & VIO of Xilinx tool gives the flexibility in verification for the design with large number of inputs & outputs, also used for easy implementation of the design into the FPGA Spartan-3. This design uses the VHDL language. In the establishment of the general expectation function, it uses the hierarchical design to realize alternating lit the traffic lights, the countdown time display and vehicles' and pedestrians' safe passing command. The program's data can be set base on actual conditions (flexible modification). The system has several benefits over ordinary traffic light controllers such as simple structure, high reliability, low costs, easy installation and maintenance and so on. So, the system owns a broader application prospect. In the future, we will further improve the function of some modules, such as FPGA kit validation and connecting real time sensors.

REFRENCES

[1] M. Voinescu, A. Udrea, and S. Caramihal “On urban traffic modelling and control,” *Journal of Control Engineering and Applied Informatics*, vol. 11, no. 1, 2009.

[2] C. M. Mwangi, S. M. Kang’ethe, and G. N. Nyakoe, “Design and simulation of a fuzzy logic traffic signal controller for a signalized intersection,” *Scientific Conference Proceeding*.

[3] O. Tomescu, I. M. Moise, A. E. Stanciu, and I. Băţroş, “Adaptive traffic light control system using AD HOC vehicular communications network,” *U.P.B. Science Bull.*, Series D, vol. 74, issue 2, 2012.

[4] “Design of an intelligence traffic light controller (ITLC) with VHDL,” Taehee Hm, Chiho Lin, Dept. of Computer Science, Semyung University.

[5] S. Nath, C. Pal, S. Sau, S. Mukherjee, A. Roy, A. Guchhait, and D. Kandar “Design of an FPGA based intelligence traffic light controller with VHDL,” *International Conference on Radar, Communication and Computing (ICRCC)*, SKP Engineering College, Triuvannamalai, TN., India, pp. 92-97, 2012.

[6] W. M. E. Medany, M. R. Hussain, “FPGA-based advanced real traffic light controller system design,” *4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, CE Department, IT College, UOB, 32038 Bahrain, pp. 100-105, 2007.

[7] X. Xiong, X. 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, 2014.

[8] M. A. Qureshi, A. Aziz, and S. H. Raza, “An FPGA based semi automated traffic control system using verilog HDL”.

[9] B. Dilip, Y. Alekhya, P. D. Bharthi, “FPGA implementation of advanced traffic light controller using verilog HDL,” *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 1, issue 7, 2012.