ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 7, September 2012

FPGA Implementation of an Advanced Traffic Light Controller using Verilog HDL

B. Dilip, Y. Alekhya, P. Divya Bharathi

***Abstract*— Traffic lights are the signaling devices used to manage traffic on multi-way road. These are positioned to control the competing flow of the traffic at the road intersections to avoid collisions. By displaying lights (red, yellow and green), they alternate the way of multi-road users. The implementation of traffic Light Controller can be through a Microcontroller, Field Programmable Gate Array or Application Specific Integrated Circuit. FPGA implementation is advantageous over ASIC and microcontroller; number of IO ports and performance compared to microcontroller and implementation with FPGA is less expensive compared to ASIC design. This paper presents the FPGA implemented low cost advanced TLC system using ChipScope Pro and Virtual Input Output. The TLC implemented is one of the real and complex signaling lights in Kingdom of Bahrain, for pedestrian way included four roads and sensors and camera assisted motorway. The system has been implemented in hardware using Spartan-3E FPGA.**

***Index Terms*—ChipScope Pro, Virtual Input Output, Integrated Controller, Field Programmable Gate Array.**

I. Introduction

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 IO (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 Fig. 1 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). Verilog is used as HDL for circuit description to code the TLC module. Verilog HDL is used because of the difficulty in writing a VHDL code which has to integrate the source code, ChipScope Pro-Integrated Controller (ICON) and Virtual Input Output (VIO).

B. Dilip, Department of ECE, MVGR College of Engineering, Vizianagaram, India, Ph: 9963414948.
Y. Alekhya, Department of ECE, MVGR College of Engineering, Vizianagaram, India.
P. Divya Bharathi, Department of ECE, MVGR College of Engineering, Vizianagaram, India.

II. TLC Flow Chart

The Flow Chart shown in Fig. 2 illustrates the actions to be taken by the road 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 [2], [3]. 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 [4].

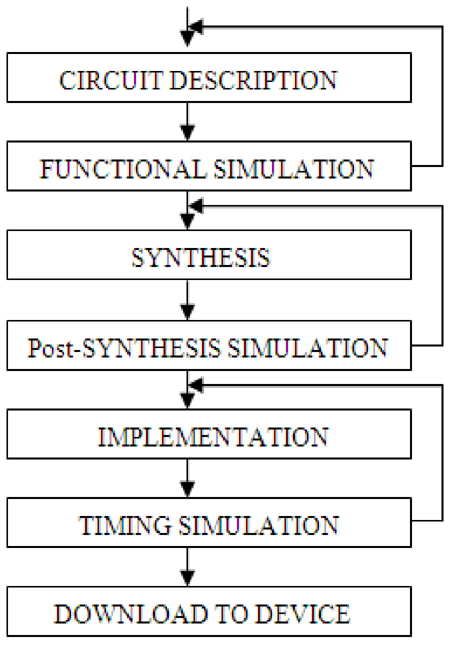


Fig. 1 FPGA Design Flow

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. The advantage of writing Traffic Light Controller program is that in a program, modifications as per requirements can be done easily 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 [5]. 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 in this paper, yellow light is split into two phases and are included in the signaling 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.

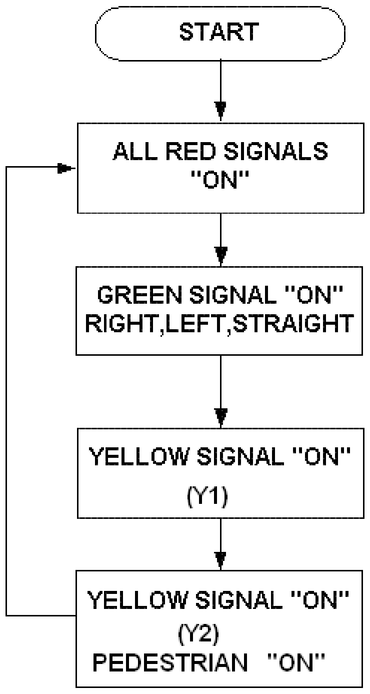

START
ALL RED SIGNALS "ON"
GREEN SIGNAL "ON" RIGHT,LEFT,STRAIGHT
YELLOW SIGNAL "ON" (Y1)
YELLOW SIGNAL "ON" (Y2) PEDESTRIAN "ON"

Fig. 2 TLC Flow Chart

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. 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. In this paper, a four road structure is considered in which the four directions labeled 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 signaling lights i.e., it changes from red to green and then to yellow and after that back to red signal.

III. State Diagram

The TLC state diagram shown in Fig. 3 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. 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.

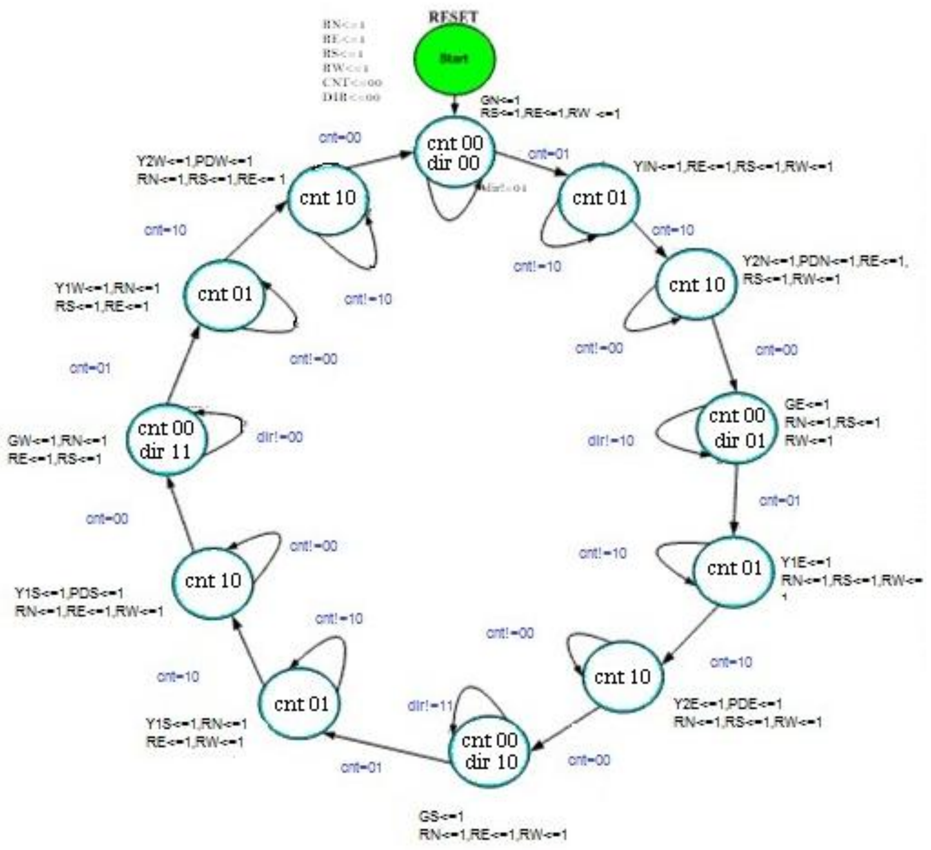
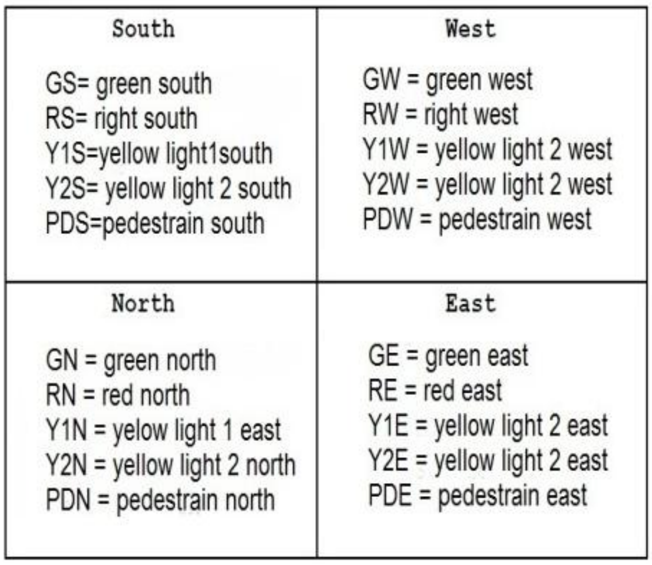


Fig. 3 TLC State Diagram

Table I. Terms used in 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

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. In the traffic light controller program there will be 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.

IV. Hardware Implementation

Fig. 4 depicts a general four road structure which consists of north, east, west and south directions each with a set of three lights namely green, yellow and red. Green light in a direction will be ON when left, straight and right side is set to be free for traffic in that direction. The figure shows the design of traffic light model. To distinguish each lane and the traffic signal lights, they are labeled separately with North, East, South and West. Signal lights at each lane have their set of traffic light signal “Red, Yellow, and Green”. Operation of this signal light is similar to common traffic light signal.

Along with these specifications, each lane has a light to represent a sensor of the corresponding road. Linear sensor or electromagnetic sensor is suitable for design of a real traffic light system. The first sensor detects the presence of vehicles and the second sensor determines the volume of the traffic corresponding to that lane. Through the two sensors, we will know the expected time for green signal ON and when the signal light at each lane should be changed to green.

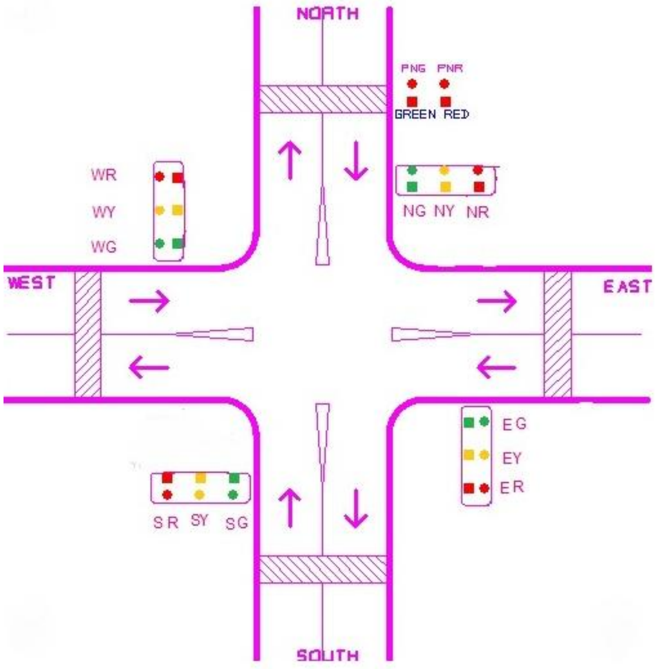


Fig. 4 Structure of TLC

The state machine is coded using the Hardware Description Language, Verilog. Spartan-3E trainer kit is shown in Fig. 5. Fig. 6 shows the FPGA Implementation of TLC. Using Xilinx ISE tool, this code is dumped into Spartan-3E FPGA trainer kit and the outputs here we considered are more than the LEDs on the FPGA, thereby we are using the ChipScope Pro.

Fig. 5 Spartan-3E

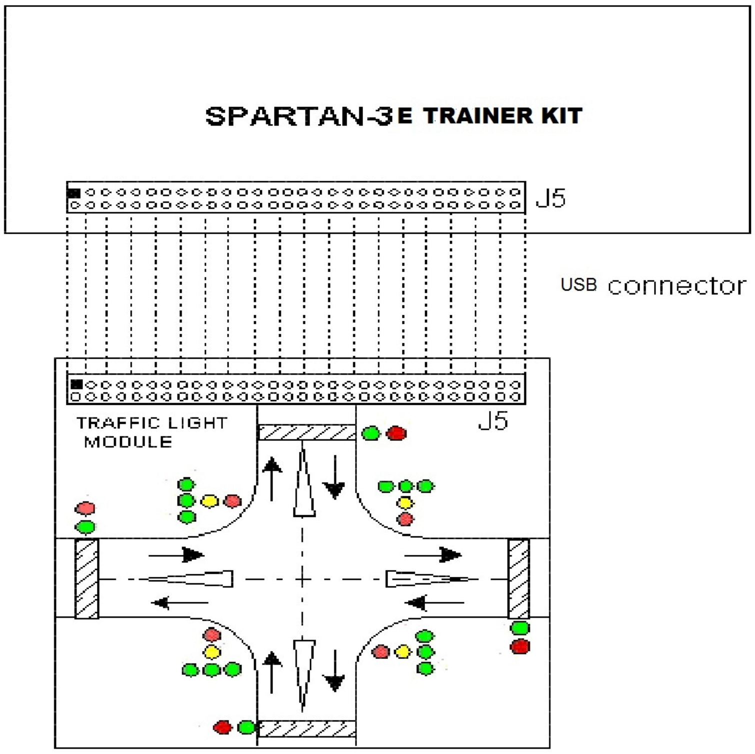
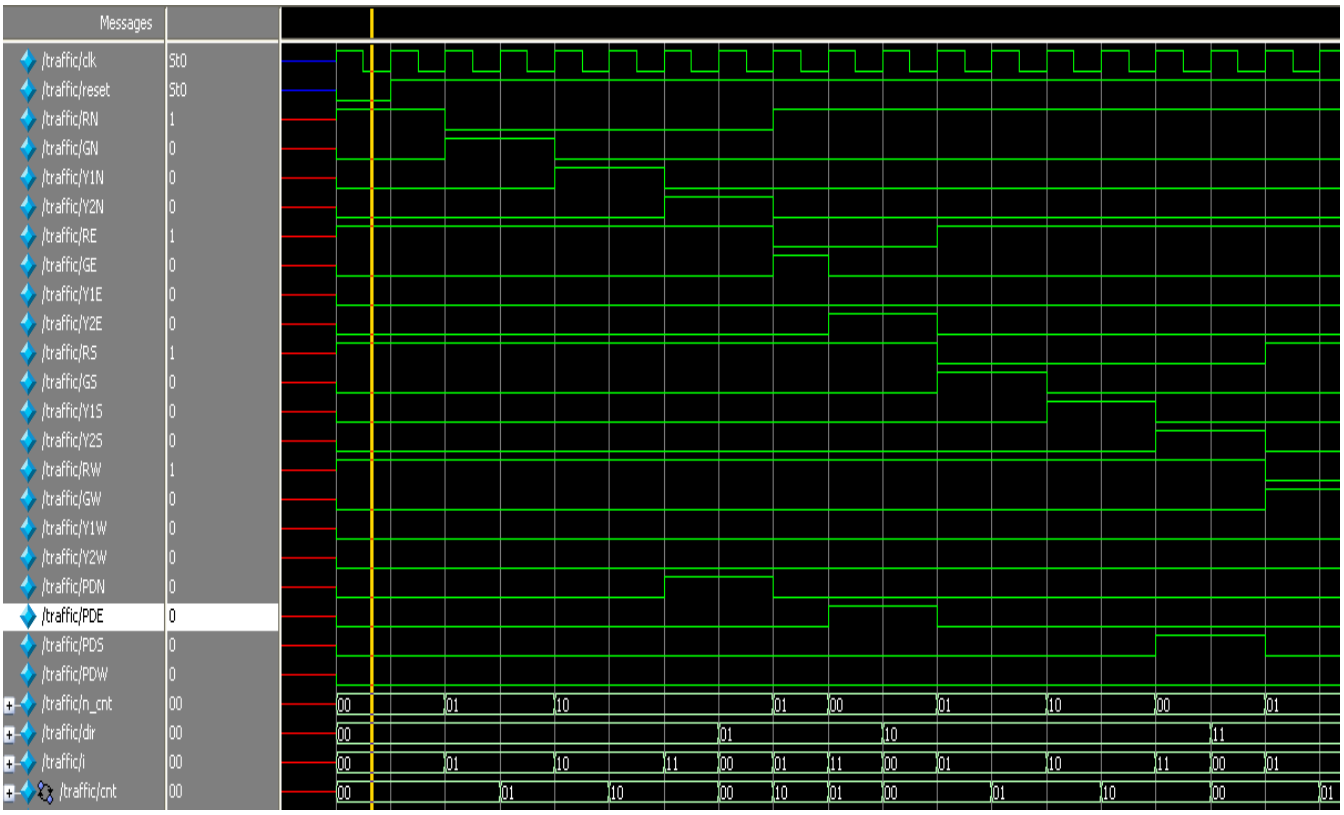


Fig. 6 FPGA Implementation

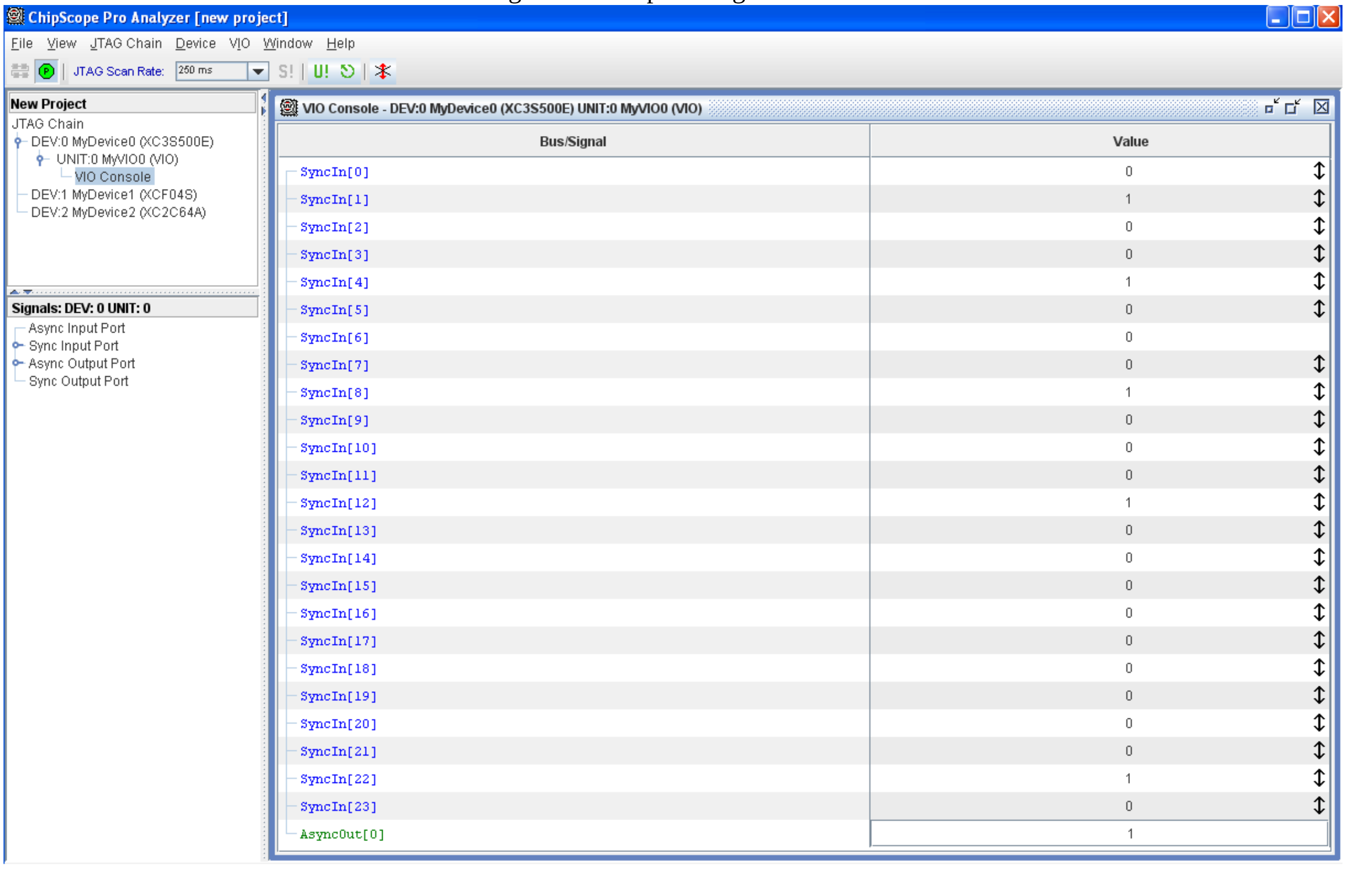
V. Experimental Results

The Traffic Light Controller has large number of outputs. Hence through the concept of Virtual Input Output (VIO), using ChipScope Pro, the output of the TLC is verified with Spartan-3E FPGA. Fig. 7 and Fig. 8 show the outputs through Modelsim and ChipScope Pro.



Messages	
/traffic/clk	St0
/traffic/reset	St0
/traffic/RN	1
/traffic/GN	0
/traffic/Y1N	0
/traffic/Y2N	0
/traffic/RE	1
/traffic/GE	0
/traffic/Y1E	0
/traffic/Y2E	0
/traffic/RS	1
/traffic/GS	0
/traffic/Y1S	0
/traffic/Y2S	0
/traffic/RW	1
/traffic/GW	0
/traffic/Y1W	0
/traffic/Y2W	0
/traffic/PDN	0
/traffic/PDE	0
/traffic/PDS	0
/traffic/PDW	0
/traffic/n_cnt	00
/traffic/dir	00
/traffic/i	00
/traffic/cnt	00

Fig. 7 TLC output using Modelsim


ChipScope Pro Analyzer [new project]
File View JTAG Chain Device VIO Window Help
JTAG Scan Rate: 250 ms
New Project
JTAG Chain
DEV:0 MyDevice0 (XC3S500E)
UNIT:0 MyVIO0 (VIO)
VIO Console
DEV:1 MyDevice1 (XCF04S)
DEV:2 MyDevice2 (XC2C64A)
Signals: DEV: 0 UNIT: 0
Async Input Port
Sync Input Port
Async Output Port
Sync Output Port
VIO Console - DEV:0 MyDevice0 (XC3S500E) UNIT:0 MyVIO0 (VIO)

Bus/Signal	Value
SyncIn[0]	0
SyncIn[1]	1
SyncIn[2]	0
SyncIn[3]	0
SyncIn[4]	1
SyncIn[5]	0
SyncIn[6]	0
SyncIn[7]	0
SyncIn[8]	1
SyncIn[9]	0
SyncIn[10]	0
SyncIn[11]	0
SyncIn[12]	1
SyncIn[13]	0
SyncIn[14]	0
SyncIn[15]	0
SyncIn[16]	0
SyncIn[17]	0
SyncIn[18]	0
SyncIn[19]	0
SyncIn[20]	0
SyncIn[21]	0
SyncIn[22]	1
SyncIn[23]	0
AsyncOut[0]	1

Fig. 8 TLC output using ChipScope Pro

VI. Conclusion

The modern ways of multi-way traffic management improves the traffic condition up to a large extent. Advanced signaling controllers contribute to the improvement of the urban traffic; which is proportional to the complexity of the controller. 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-3E.

REFERENCES

[1] Y.S. Huang, T.H. Chung, and T. Lin, "Design and Analysis Urban Traffic Lights Using Timed Colour Petri Nets," in Proc. of *IEEE International Conf. on Networking, Sensing and Control*, pp. 248-253, 2006.
[2] J. Niittymaki And M. Pursula, "Signal control using fuzzy logic," *Fuzzy Sets and Systems*, vol. 116, no. 1, pp. 11-22, 2000.
[3] C.P. Pappis, and E.H. Mamdani, "A fuzzy logic controller for a traffic junction," *IEEE Trans. Systems, Man, and CyberNetics*, vol. 7, no.10, pp. 707-717,1977.
[4] J. Niittymaki And S. Kikuchi, "Application of fuzzy logic to the control of a pedestrian crossing signal," Transportation Research Record: *Journal of the Transportation Research Board*, vol. 1651, pp. 30-38, 1998.
[5] S. Chiu, "Adaptive traffic signal control using fuzzy logic," in Proc. of the *IEEE Intelligent Vehicles Symposium*, pp. 98-107, 1992.
[6] J. Niittymaki, "Installation and experiences of field testing a fuzzy signal controller," *European Journal of Operational Research*, pp. 273-281, 2001.

B. Dilip obtained his B.Tech (ECE) degree from Pydah College of Engg & Tech, JNTUK in 2010. He is pursuing his M.Tech (VLSI) from MVGR College of Engg. He has a publication in an Int'l Journal and 2 presentations at Int'l & National conferences. He was awarded Interscience Young Investigator by Interscience Research Network. He was qualified in Gate'10. His areas of interest are Logic Design and VLSI.

Y. Alekhya did her B.Tech in Electronics and Communication Engineering from M.V.G.R College Of Engineering, JNTU Kakinada in 2009. She is pursuing her M.Tech (VLSI) from M.V.G.R College Of Engineering. She has a publication in an international journal and presented many papers in various National and International conferences. Her research interests include Communications and VLSI.

P. Divya Bharathi obtained her B.Tech degree from MVGR College of Engineering, JNTU Kakinada in 2012. She has got selected for TCS in campus drive'11.She was an IEEE student branch member. She received a Merit Certificate for being selected into the finals of Inter-Collegiate E-Plus Club Challenge-2011. She has presented many papers at various symposiums. Her areas of interest include Embedded Systems and Digital circuits.