International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878, Impact Factor (2014): 3.05

Advance Traffic Light Control System Based on FPGA

Shilpa U. Holambe[1], D. B. Andore[2]

[1]Department of E&TC, MIT Aurangabad, Maharashtra, India
[2]Prof. Department of E&TC, MIT Aurangabad, Maharashtra, India

Abstract: ***Traffic light 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 of accidents, time losses. This system proposed, in this paper aims at reducing waiting times of the vehicles at traffic signals. Traffic Light Control (TLC) system also based on microcontroller and microprocessor. But the disadvantage of with microcontroller or microprocessor is that it works on fixed time, which is functioning according to the program that does not have the flexibility of modification on real time basis. This proposed system using FPGA with traffic sensors to control traffic according requirement means designer can change the program if it require and thus reduces the waiting time. The hardware design has been developed using Verilog Hardware Description Language (HDL) programming. The output of system has been tested using Xilinx. The implementation of traffic Light Controller also through Application Specific Integrated Circuit. But implementation with FPGA is less expensive compared to ASIC design. This paper presents the FPGA implemented low cost advanced TLC system. Coding of the design is done using Verilog HDL and the design is tested and simulated on Spartan-3E FPGA development kit.***

Keywords: Virtual Input Output, Integrated Controller, Field Programmable Gate Array, HDL, Xilinx

1. Introduction

Traffic problem of roads especially in the modern cities is increases day by day as number of vehicles on the road is increasing. Due to limited resources provided by current traffic system like TLC with microcontroller or microprocessor are leading to ever increasing travelling times, and waiting time of road users. The Advanced Traffic Light System proposed in this project aims at minimizing the waiting times of vehicles at the traffic signals. Field programmable gate arrays (FPGAs) are extensively used in electronic systems, for rapid prototyping and verification of a conceptual design, especially when the mass-production of a conventional IC becomes prohibitively expensive due to the small quantity. Many electronic system designs that used to be built in conventional silicon VLSI are now implemented in Field Programmable Gate Arrays. In this system design traffic of four road is controlled using FPGA for 24 hour and minimize waiting times for the road users. The design is implemented in Verilog HDL Hardware Description language. FPGA is an integrated circuit that contains an array of identical logic cells which are programmed by user. It provides high density logic together with RAM memory in each device. To code the TLC system Verilog HDL is used. Verilog HDL is used because of the difficulty in writing a VHDL due to strongly typed language which has to integrate the source code. The traffic light control system works on the specific switching sequence of Red, Green and Yellow lights in a particular way with given time form in program. This Traffic Light sequence is generated using a specific switching mechanism which will help to control a traffic light system on a road in a specified sequence.

2. Survey on Traffic Light Control System

In many cities Traffic Light Controller (TLC) is based on microcontroller and microprocessor. These TLC systems with microcontroller and microprocessor have limitations because it uses the pre-defined hardware, which is works as given program that does not have the flexibility of modification on real time basis. This program is fixed which is not reprogrammable or erasable by designer .Due to the fixed time intervals of green, orange and red signals the waiting time is more. If waiting time of vehicles is more than fuel loss also occurred. So we have to implement some advanced system for traffic control due to this road user can save their time.

The implementation of traffic Light Controller can be through Application Specific Integrated Circuit. ASIC design is more expensive than FPGA. Most of the TLCs implemented on FPGA are simple ones that have been implemented as examples of FSM.

Traffic Light Control System can be implemented with Programmable Logic Device (PLD) and Complex Programmable Logic Device (CPLD). PLD like PALs and GALs are available only in small sizes, equivalent to a hundred of logic gates. So traffic light control system is not controlled by PLDs which is having more crowds of vehicles on road.

Complex Programmable Logic Device (CPLD) is also used for TLC system. CPLD having large number of logic gates Now, CPLD can replace thousands, or even hundreds of thousands, of logic gates. But CPLDs doesn't have much memory. Due to lack of memory devices require lots of flip flops which complicate the design of system. When comparison of response time for various frequencies, for both is observed CPLD was performing twice as better than PLD.

PLD based circuit shows a delayed response. The response with respect to clock, found that delay response of PLD is twice as much than the delay response of CPLD at a nano second level. Traffic system which requires fast response, CPLD may be the best choice. But further More to implement more complex circuits and tested the capability;

the CPLD is not useful because not having very large number of gates capacity.

CPLDs having thousand to ten thousand of logic gates available. FPGA is the perfect replacement for CPLD. CPLD and FPGA is having somewhat same features but FPGA is having more logic gates availability. FPGAs typically range from tens of thousands to several million which is more than CPLD.

FPGA which offers many advantages over microcontrollers such as fast speed, number of input/output ports, and performance which are all very important in TLC design. FPGAs are famous for their low-cost, high-volume applications and are very good as replacements for fixed-logic gate arrays. The FPGA is not only available for a very low cost, but it integrates many architectural features associated with high-end programmable logic. Due to these advantageous features like low cost and integrated features has made FPGA an ideal. By using Application Specific Integrated Circuits (ASIC) traffic light system become very expensive .So that FPGA replaces ASIC designs also.

3. Field Programmable Logic Array

A field-programmable gate array (FPGA) is a semiconductor device that can be configured by the designer after manufacturing hence the name "field-programmable". By using a logic circuit diagram or a source code in a hardware description language (HDL) FPGAs are programmed. This program is reprogrammable by designer when it necessary .If FPGA is programmed by user then also user can edit or change the program. Implemented program in FPGA shows the working of chip or kit. They can be used to implement any logical function that an application-specific integrated circuit (ASIC) could perform, but the ability to update the functionality after shipping offers advantages for many applications. Field Programmable Gate Arrays (FPGAs) are expansively used in quick prototyping and verification of conceptual design and also used in electronic systems where the mask-production of a custom IC becomes really expensive due to the small quantity. The system has been implemented in hardware using Spartan-3E FPGA. FPGA design flow is shown in Fig1. According to that start with circuit description in which all the circuit is designed by logic gats which is done by using Hardware Description Language (HDL) .Then functional description was done which is followed synthesis and post Synthesis simulation.

After that implementation and time simulation is occurred and generated file is downloaded in to the target device this system used as target device is FPGA kit. Design or circuit description can be done by using HDLs which is followed by 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).

FPGAs have gained rapid acceptance and growth over the past decade because they can be applied to a very wide range of applications. A list of typical applications includes: random logic, integrating multiple SPLDs, device controllers, communication encoding and filtering, small to medium sized systems with SRAM blocks. Other interesting applications of FPGAs are prototyping of designs later to be implemented in gate arrays, and also emulation of entire large hardware systems.

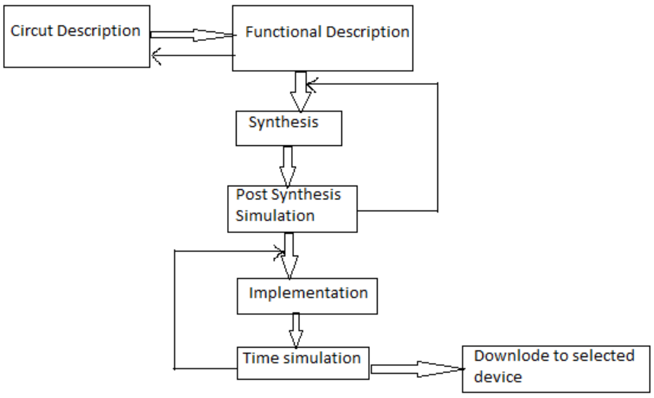

Figure 1: FPGA Design Flow

3. 1. Verilog Hardware Description Language (HDL)

Verilog is Hardware Description languages that are used to write programs for electronic chips. This language is used in electronic devices that do not share a computer's basic architecture. Verilog is relatively recent, and follows the coding methods of the C programming language. Verilog uses weak typing, which is the opposite of a strongly typed language like VHDL. It is the case sensitivity. Verilog is case sensitive, and would not recognize a variable if the case used is not consistent with what it was previously. In general, Verilog is easier to learn than VHDL. This is due, in part, to the popularity of the C programming language, making most programmers familiar with the conventions that are used in Verilog. Verilog has no concept of packages, and all programming must be done with the simple data types that are provided by the programmer. Originally a modeling language for a very efficient event-driven digital logic simulates. Later pushed into use as a specification language for logic synthesis. Now, one of the two most commonly-used languages in digital hardware design is VHDL and other is Verilog HDL.. Virtually every chip (FPGA, ASIC, etc.) is designed in part using one of these two languages Combines structural and behavioral modeling styles.

3. 2. Xilinx

System is coded by using Verilog HDL this code is dumped in FPGA development kit by using Xilinx ISE tools. When you open a project file from a previous release the ISE software prompts you to migrate your project. If u click backup and migrate or migrate only the software automatic converts your project file to the current release. After you convert your project you cannot open it in previous versions of ISE software.

If your design includes IP modules that were created using CORE Generator software or Xilinx platform studio (XPS) and you need to modify these modules you may be required to update the core .However if the core net list is present and you do not need to modify the core, updates are not required and the existing net list is used during implementation.

Table 1: Tabular Form of States

Input as Clock and reset		Directions of lanes			
S	S	North	East	South	West
Initial Condition		Red	Red	Red	Red
0	0	Green	Yellow	Red	Red
0	1	Red	Green	Yellow	Red
1	0	Red	Red	Green	Yellow
1	1	Yellow	Red	Red	Green

4. Hardware Implementation

4. 1. TLC Structure

Figure 2 shows the structure of the four roads (square) that has been used as a to design proposed system. In this structure four traffic signals are present. Road structure shows four traffic lane represented by north side lane, east side lane, south side lane and west side lane. Every lane has their own separate traffic light system which is having regular as usual red, yellow and green lights. the north side lane has north green ,north red, and north yellow light which is presented by NG,NR,NY respectively. Similarly east side also having EG, ER, EY respectively. And the lights of south side are presented by SG, SR, SY. Similarly the west side lights also presented as WG, WR, WY.

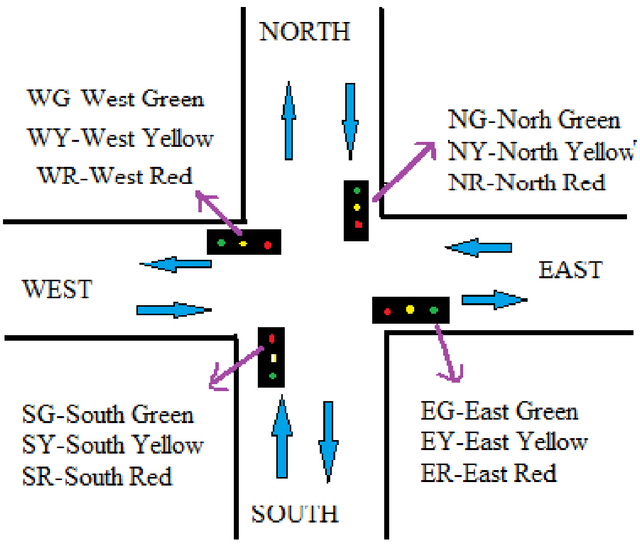

Figure 2: Four Road of TLC Structure

Traffic Light Controller can be designed by starting with certain assumptions. Initially Red signal is ON in North East, West and South direction.

There are four traffic light signals, in the below figure which are to be controlled. These four signals have same priority as they all are main roads. Now when the Reset is made high the North traffic will be allowed to move and traffic in all the remaining directions are stopped. Later the traffic in all the other direction is allowed to move in the sequence.

4. 2. State Description

The sequence of traffic is as shown in Figure4 fist North rode allow to move the traffic after that East rode allow traffic to reach their place . After east South rode allow moving the traffic on rode and then West rode allow to move the vehicles. The advantage of this particular Traffic Light Controller program is that modification can be done easily as per the requirements i.e., suppose the traffic on main road and the side road can be controlled by changing the states accordingly, when the main road traffic is heavy as compared to the side road traffic at that time the time simulation of main road is large than side rode means green light glowing time of main side rode is large than side rode because number of vehicles are more than side rode.

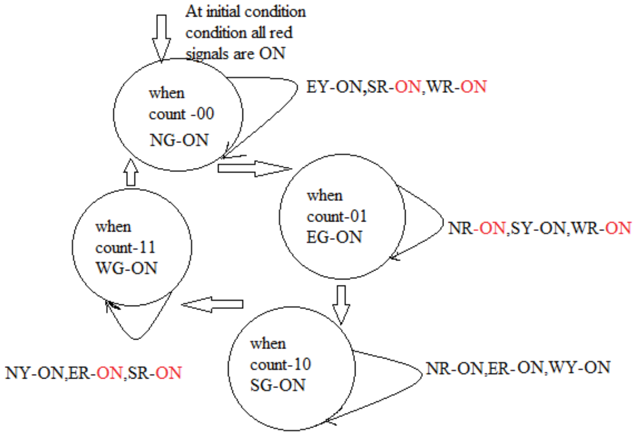

Figure 3: State Diagram

There is two inputs are present namely clock and reset. The TLC states shown in Figure3 which works on the changing count of given inputs. Initially the all side rode signals was red which is initial condition after that execution of program is started as shown in fig4.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 li ON for few seconds 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 Whenever cnt=01 and dir=01 then yellow will be ON for few seconds 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 yellow light (y1) will be ON for few seconds 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 then yellow light (y1) will be ON for few seconds 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.

4. 3. FPGA Model

Spartan -3 families offers densities ranging from 50,000 to five million system gates. It is programmed by loading configuration data into robust, reprogrammable, static CMOS configuration latches (CCL) that collectively control all functional elements and routing resources. Spartan-3 FPGA platform also allows the user to make significant changes while keeping original device pin outs thus eliminating the need to re-tool PC boards.

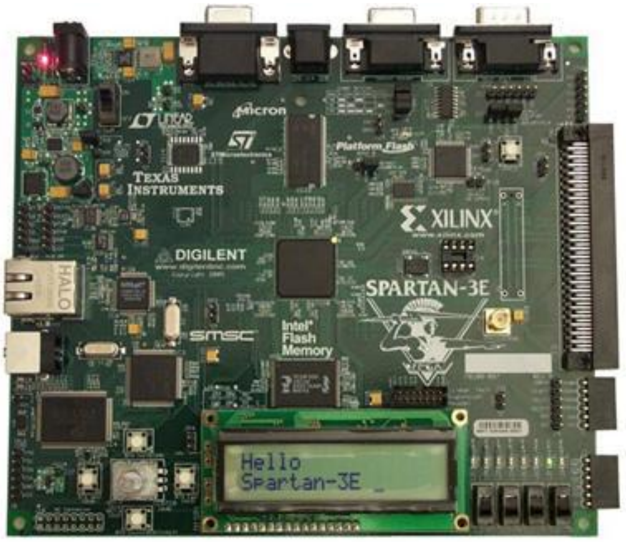

Figure 4: FPGA Spartan-3E Development Kit

We can easily upgrade, modify, and test the designs even in the field itself. Embedded capabilities make Spartan-3 devices ideal as coprocessors or pre-and post-processors, offloading highly computational functions from a programmable DSP to enhance system performance.

Contemporary FPGAs have large resources of logic gates and RAM blocks to implement complex digital computations. As FPGA designs employ very fast I/Os and bidirectional data buses it becomes a challenge to verify correct timing of valid data within setup time and hold time.

5. Simulation Result

Figure 6.Shows the simulation results for the controller with sensors output = logic '0', i.e. when traffic is slag, and the transition time will be less.

Figure7 Shows the simulation results for the controller with sensors output = logic '1', i.e. when traffic is crowded, the transition time will be more.

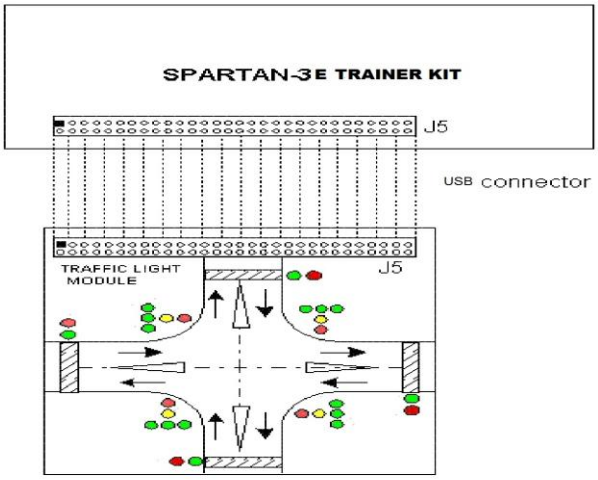

Figure 5: FPGA Implementation with TLC Model

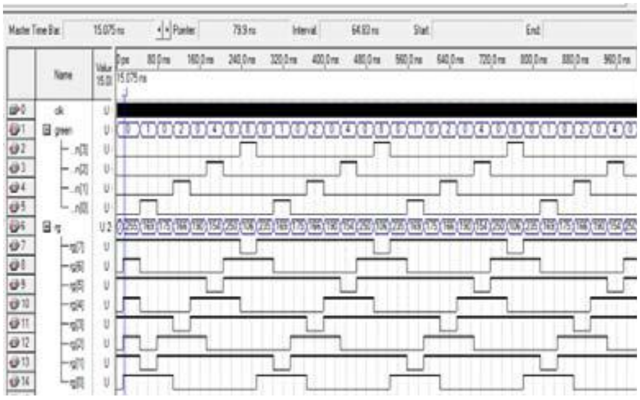

Figure 6: Simulation Result "S" ='0'

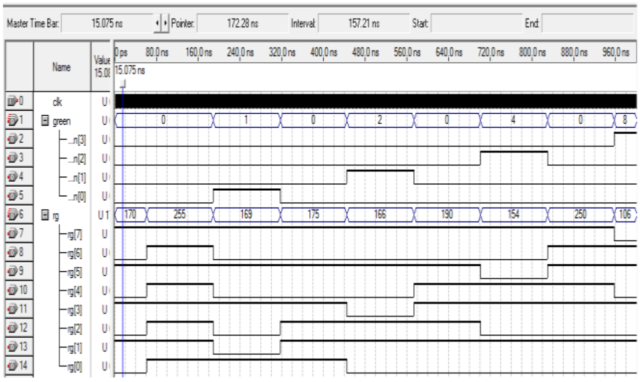

Figure 7: Simulation Results "S" ='1'

6. Conclusion

The proposed system implemented advanced traffic light control system which control complex traffic in modern cities. This system uses FPGA which made this system advanced FPGA is device which is configured by designer or user. The very useful application of FPGA is that designer can change the program at any instant which is easy to reprogram. User can change the program as per requirement. Verilog HDL is used to circuit description, code is generated which is dumped in to the FPGA by using Xilinx. Spartan 3E FPGA series is used as development kit. Now a day's problems related to traffic are very serious issue due this problems number of accidents increases rapidly in modern cities. Because of lack of management in TLC system road user loss their valuable time. So to overcome these disadvantages we need some what advanced TLC system. FPGA is very good replacement for that traditional TLC systems with microcontroller having fixed time slots. This four rode TLC structure with FPGA can solve any complexity related to traffic .FPGA is many times advantageous than microcontroller, ASIC designs and also having low cost.

Acknowledgment

It is my pleasure to get this opportunity to thank my beloved and respected Guide Prof. D.B. Andore who imparted valuable basic knowledge of Electronics and telecommunications specifically related to Microwave Domain. We are grateful to Elec. & Comm. MIT, Aurangabad for providing us infrastructure facilities and moral support.

References

[1] Jose E. Ortiz and Robert H. Klenke. "Simple Traffic Light Controller: A Digital Systems Design Project," IEEE SoutheastCon 2010(SoutheastCon), Concord, NC, March 2010, pp. 85-88

[2] W.M. El-Medany and M.R. Hussain. "FPGA-Based Advanced Real Traffic Light Controller System Design," Technology and Applications, 2007. IDAACS 2007. 4th IEEE Workshop, pp. 100-105

[3] M.F.M. Sabri, M.H. Husin, W.A.W.Z Abidin, K.M. Tay and H.M. Basri "Design of FPGA-based Traffic Light Controller System," Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference, Shanghai, June 2011 pp.114-118

[4] S. Shi, T. Hongli and Z. Yandong "Design of Intelligent Traffic Light Controller Based on VHDL," Knowledge Discovery and Data Mining, 2009. WKDD 2009. Second International Workshop, pp. 272 – 275

[5] H. Taehee and L. Chiho "Design of an intelligence traffic light controller (ITLC) with VHDL," TENCON '02. Proceedings. 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering, Oct. 2002 , pp. 1749 - 1752 vol.3

[6] L. Zhenggang, X. jiaolong, Z. Mingyun and D. Hongwei "FPGA-Based Dual-Mode Traffic Lights System Design," Information Science and Engineering (ICISE), 2009 1st International Conference, Dec. 2009, pp.558-561

[7] M.S. Islam, M.S. Bhuyan, M.A. Azim, L.K. Teng and M. Othman, "Hardware Implementation of Traffic Controller using Fuzzy Expert System," Evolving Fuzzy Systems, 2006 International Symposium, pp.325-330

[8] M. Giradkar and M. Khanapurkar, "Design and Implementation of Adaptive Front Light System of Vehicle Using FPGA Based LIN Controller," Emerging Trends in Engineering and Technology (ICETET), 2011 4th International Conference, pp.256-261

[9] M. Ramzanzad and H.R. Kanan, "A new method for design and implementation of intelligent traffic control system based on fuzzy logic using FPGA," Fuzzy Systems (IFSC), 2013 13th Iranian Conference, pp.1-4

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