DESIGN OF FSM ARCHITECTURE BASED ON TRAFFIC LIGHT CONTROLLING UNIT

Sk. Nageena, Assistant Professor, Dept. Ece, Sri Indu College Of Engineering And Technology
Gonugunta Raj Kumar, Assistant Professor, Dept. Ece, Sri Indu College Of Engineering And Technology
Suresh Ballala, Associate Professor, Dept. Ece, Sri Indu College Of Engineering And Technology

1.ABSTRACT

The purpose of this paper is to design and implementation of smart traffic light controller system using VHDL language and FPGA. A structure of four road intersection has been selected. The intersection to be controlled is between a busy (main street), and somewhat less busy (side street), with sensor for the side street and walk request button. Also, the system contains switches to control the traffic light manually. The intersection uses four timing parameters with ability to change these parameters manually. The system has been successfully tested with VHDL using Xilinx ISE 14.7i software environment and Chip-Scope, while, it is implemented in hardware using Xilinx Spartan 3E FPGA. It is easy to use and the cost for the same is also less as compared to the others. The designed traffic light control system is presented to work correctly as predictable.

KEYWORDs

Keywords: Traffic Light Controller (TLC); FSM; VHDL; Spartan 3E; FPGA; Xc3s500fg320-4.

2.INTRODUCTION

Traffic jamming is a critical predicament in many of the cities and towns all over the world. Traffic congestion has been causing many setbacks and challenges in the major and most occupied cities all over the globe. This traffic jam directly impacts the productivity of the workers, traders, suppliers and in all affecting the market and raising the prices of the commodities in a way light. The problem of heavy jam is happened because of never configure the level of jam in each way and set the delay time. Another problem represents when there is no jam, but the waiting still continues. The solution for these problems is to determine the level of jam and set the delay time. This problem need of evaluation of the traffic policeman, and then there is need for manual control of the

traffic. The target of this paper is to propose system provide solution for all above problems with least possible cost. Traffic light controller (TLC) can be implemented using microcontroller, FPGA, and ASIC design. FPGA has many advantages over microcontroller, some of these advantages are; the speed, number of input/output ports and performance which are all very important in TLC design, at the same time ASIC design is more expensive than FPGA. Nowadays, FPGA becomes one of the most successful of today's technologies for developing the systems which require a real time operation. FPGA is a reconfigurable integrated circuit that consists of two dimensional arrays of logic blocks and flip-flops with an electrically programmable interconnection between logic blocks. The reconfiguration property enables fast prototyping and updates for hardware devices even after market launch . Most of the TLCs implemented on FPGA are simple ones that have been implemented as examples of Finite State Machine (FSM).The VHDL language has been selected for programming the FPGA to fill two important needs in the design process.

3. TRAFFIC LIGHT CONTROLLER SYSTEM DESIGN

Figure 1 illustrates the structure of the selected traffic light model for four road intersections (one Main Street and three side streets). In general, Traffic Light Controller System consists of three lights (red, green and yellow) in each direction .The red light indicates to Stop, green light indicates to allow the traffic and yellow light indicates the caution that the traffic is going to be stopped in few seconds. While, turning in yellow and red lights at the same time indicates the caution that the traffic is going to be moving in few seconds. The intersection is fitted with a sensor for side street traffic and with walk request button. This traffic light controller also has provision for walk light (which consists of two lights red and green, where, green light allows the walkers to pass the street while red light avoids the walkers from passing the street) and for the traffic sensors in each one of the side streets. A simple block diagram of the traffic light controller system is exposed in Figure 2. The design is composed of finite state machine (FSM), data storage (D_RAM), timer, divider, and various synchronizers (latch, and synchronizer).

3.1. Finite state machines (FSM)

Finite State Machines (FSM) is the heart of the traffic light controller system. This FSM controls the loading of static data storage locations with timing

parameters, displaying these parameters by reading RAM locations, and the control of the actual traffic lights. There are four timing parameters in this system as displayed . They are the base interval (TBASE) for side green, an extended interval for main green and walk green light (TEXT), the time for yellow light (TYEL), and a blink interval (TBLINK). The user can specify the four timing parameters using two switches (L0, L1) manually. The FSM can execute four functions specified by two functions switches (F0, F1). These functions are listed , where, the user can execute one of four possible functions: writing new timing parameters, reading old timing parameters, running traffic light in normal mode, and running the traffic light in blinking mode as obtained . Besides, the idle state of the FSM is called the reset state, in this state, the lights are turned off and the system does not do anything. The system will stay in the reset state until the pressed.

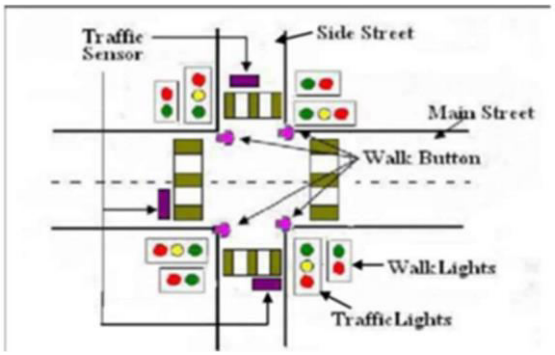

Fig 1.Traffic light module

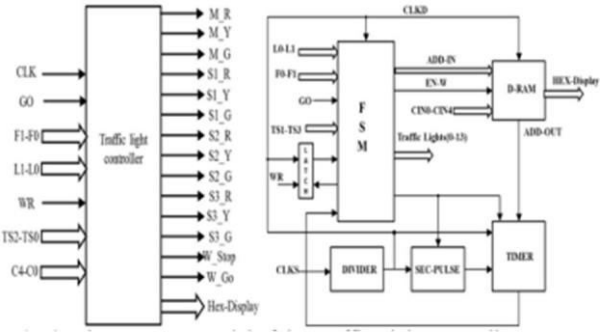

Fig 2.Structure module of Traffic light contoller

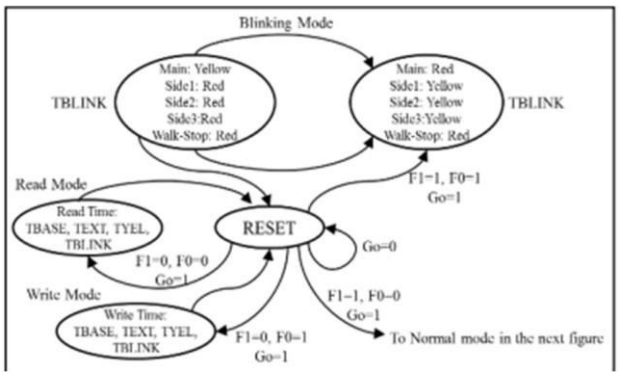

Fig 3. FSM Transition Diagram

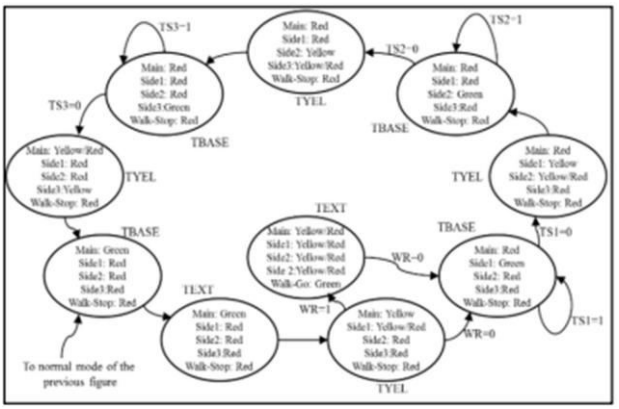

Fig 3a. FSM Transition Diagram

Using the writing function, the user can specify the any one of the four timing parameters as shown in Table 1 using (L1, L0) switches, the value of the parameter is

set using the (C4_C0) switches. For the reading operation, the user can use the same L1 and L0 switches to denote which of the four timing parameters to view on a set Hex-LEDs. In normal mode or blinking mode, the system just cycles through the various traffic light states. The regular controller has been designed with nine states as presented without taking the traffic sensors and walk request in the point view.

In the normal mode that is displayed in , the side street has a shorter green interval than the main street, but if there is traffic on the side street when the controller is about to cycle to turn that green light off, it will extend the green light by the shorter (side street) green interval. Thus the green light on the side street will stay on until traffic on the side street clears. Traffic sensor switch is used to simulate the effect waiting traffic on the side street, the system complies by keeping side street green until the traffic sensor is switch off. The walk light comes on after the main street yellow interval, and then only if the walk request button has been pushed. Late at night or when something in the system is not working, the light goes into the blinking mode this involves the lights blinking on and off, alternating between main yellow side red, and main red side yellow.

3.2. D_RAM

This component is used to store the four timing parameters which are declared in Table 1. Depending on the signal en_w, which select to read the contents by L0-L1 switches, or write new timing parameters by C0-C4 switches and display the contents on the HEX_LEDs.

3.3. Divider, sec_pulse, and timer

The divider component is used to generate the clock (1 MHz) for overall system from 50 MHz of the chip FPGA Spartan 3E. While, the Secpulse component is used to generate one second clock, which is used in the timing of the traffic light. The timer is implemented as counter.

3.4. Latch and sensors

Walk signal is latched so that when the user pushes the walk button once the signal is queued until the FSM need it. Figure 4 expresses schematic circuit of walk latch. Furthermore, there are three traffic sensors which are synchronized by simply passing it through a flip flop.

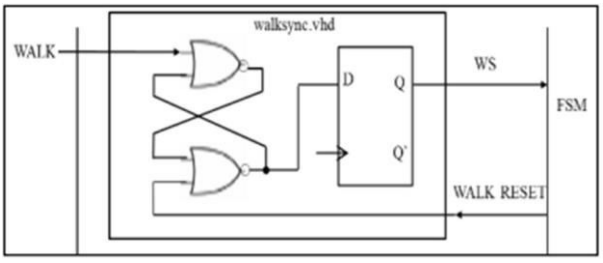

Fig 4. Walk Latch Schematic Circuit

5. HARDWARE IMPLEMENTATION

The traffic light controller system design is implemented by synthesizing the VHDL structural code design, then generating bit file using Xilinx ISE 14.7 tools. This bit file is downloaded to the FPGA Spartan 3E development kit xc3s500efg320. The system's outputs are more than the LED on FPGA, then, it is used the LEDs to display one state or use the supporting chip (expansion) external pin digilent (FX2 MIB) as confirmed in Figure 7. Figures 8 and 9 demonstrate the real time implementation of TLC and Chip-Scope implementation respectively. The system design gives the realization of the hardware system as well as the software. The hardware consumptions are listed .

Fig 5. Spartan 3-E Training Kit

5. SIMULATION RESULTS

The key advantage in using the VHDL in systems design is allowing the behavior of the required system to be described (modeled) and verified (simulated) before synthesis tools translate the design into real hardware (gates and wires). Figure 4 indicates the RTL and technology schematic diagram of the traffic light controller system. All component of the system are simulated using Xilinx ISE 14.7i. Viewing a schematic allows to see a technology level representation of HDL optimized for specific device architecture, that it may be assisted to discover the scheme issues early in design process. The simulation result of the traffic light controller system in reading and writing modes are exhibited in Figure 5. In this case, all traffic light outputs are off and HEX_LEDs display the output of memory location which represents the selected time mode. Furthermore, the normal mode is displayed in Figure 6b, which denotes the operation of TLC system as appeared in Figure 3. The synthesis process generates net list for each design element. Synthesis process checks code syntax and analyzes the hierarchy of the design to ensure that the design is optimized for the system.

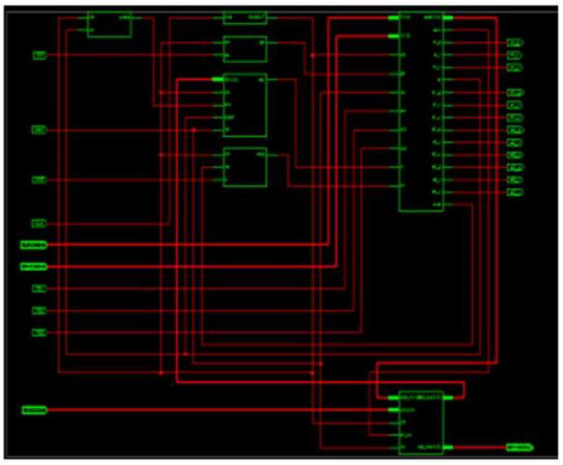

Fig 5.Technology Schematic

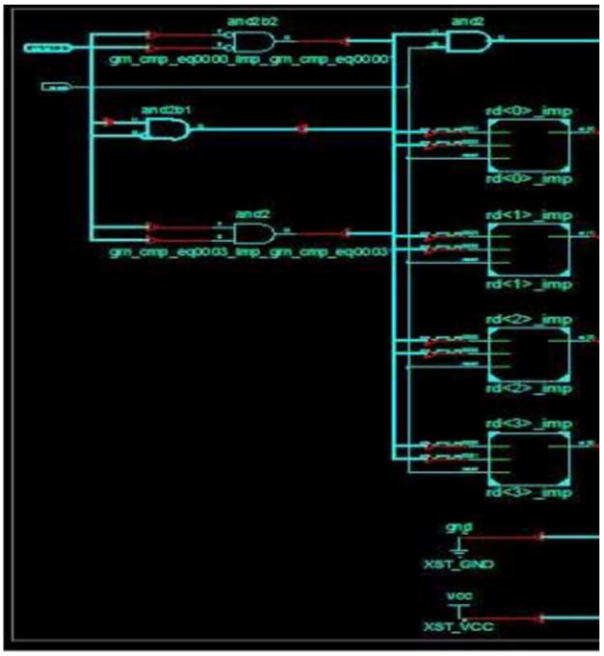

Fig 6. Schematic diagram

7. CONCLUSION

A smart Traffic Light Controller system is designed using FPGA for four roads intersection with traffic sensors and walk request signals. The system has been simulated using VHDL to realize alternating traffic light and FSM for efficient T.L.C. with ability to change its timing parameters manually. Each subcomponent is constructed and tested thoroughly before moving onto the next one. The design is robust; all the design decisions were inspected comprehensively before employment. Synchronization component are very important in the system design where they are implemented without any hazards in the system. Overall the design and implementation of the traffic light controller is respectable to design more complex system. The system is verified on FPGA Spartan 3E xc3s500efg320-4.

REFERENCES

Parag K. Lala, "Digital System Design Programmable Logic Devices", Page15-17, Chapter 1, B S Publications

https://www.csus.edu/indiv/p/pangj/spartan3e/handout_switches_buttons_pin.pdf

Hawkins, G. (1992a), "Evolution of the MUTCD: Early Editions of the MUTCD", ITE Journal , Vol. August, pp. 17–23

Hawkins, G. (1992c), "Evolution of the MUTCD: The MUTCD since World War II", ITE Journal , Vol. November, pp. 17–2