INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & MANAGEMENT

AN OVERVIEW ON DESIGNS OF INTELLIGENT TRAFFIC LIGHT CONTROLLER

Prof.Padmini G. Kaushik, Vishal D. Dahake*, Chunendra G. Meshram, Nilesh A.Take, Nikhil D.Choudhari, Pratibha S.Lakade

*Dept. of Electronics & telecommunication engineering, J.D.I.E.T. Yavatmal,Maharashtra, India

ABSTRACT

The traffic in urban areas is mainly regularized by traffic lights, which may contribute to the unnecessary long waiting times for vehicles if not efficiently configured. This efficient configuration can be the part of traffic light controller system, but still the most of the traffic lights are based on a 'fixed cycle' protocol. This paper concerned with an overview on FPGA design implementation of a low cost 24-hour advanced traffic light controller system that was built as a VLSI design using VHDL and its comparison with traffic light controller system using IR sensors and microcontroller. The traffic light is one of the real and complex traffic lights for four roads and motorway with sensors. FPGA Design Based on dual-mode traffic light control system has a simple circuit with high reliability and high computing speed, better scalability, as well as reconfiguration of good characteristics. The microcontroller used in the system is 89V51RD2 which is MCS-51 family based. The system is having IR transmitter and receiver which are mounted on the either sides of roads respectively. This paper also makes attempt to supply the benefits and disadvantages of both the systems.

KEYWORDS: Traffic Light Control System, Field-programmable Gate Array (FPGA) ,Xilinx, VHDL, VLSI, 89V51RD2,IR sensors.

INTRODUCTION

The use of FPGAs is growing in embedded designs and the embedded systems software market must continue to change to support this shift. The FPGA-based design is having more flexibility but the embedded software solution, specifically the RTOS (Real Time Operating System), middleware, and drivers should complement the benefits of the FPGA. The RTOS have to be well designed and easy-to-use, with the necessary and should offer sufficient middleware stacks and drivers. In addition to this offerings, the RTOS should also maintain the embedded designs related characteristics, which are small code size, low interrupt latency, and full services needed in many application. The use of an FPGA in embedded system is a growing trend. Because of the flexibility of the FPGA, it is applicable design in a wide range of markets.

There are two kinds of the VHDL design that are modeling and synthesis. The modeling VHDL design has significant advantage in complicated embedded system design. In addition, the VHDL should not be thought as a programming language. This language is designed to describe the complex logic circuit. A specified model is a very helpful point to start programming the system. [2]

The microcontroller 89V51RD2 used in the system is of MCS-51 family based. The system is having IR transmitter and receiver which are mounted on the either sides of roads respectively. When IR system gets activated, if any vehicle passes on road between IR transmitter and receiver then microcontroller controls the IR system and counts number of vehicles passing on road and also store vehicles count in its memory. Based on varying vehicles count, the microcontroller takes decision and updates the results according to traffic light delays. The traffic light is situated at a certain distance from the IR system. Thus based on vehicle count, microcontroller allots different ranges for traffic light delays and updates those accordingly.

The main objective of this paper is to overview the design of a 24-hour traffic light controller to manage the traffic movement of four roads at the same time, and achieve maximum utilization for the four roads using FPGA and microcontroller 89V51RD2.

The rest of this paper is organized as follows: section 2 reviews FPGA Design Based on dual-mode traffic light control system. In section 3, we review the traffic light controller system using IR sensors and microcontroller. Section 4 we discuss comparison details and in section 5 conclude the paper. At the end of the paper is a list of references.

OVERVIEW ON FPGA DESIGN

FPGAs are being forecasted for use in several industries, from consumer electronics to military aerospace, automotive to industrial control. The FPGA and ASIC tradeoffs can create a difficult decision with development teams. The FPGA provides synthesized development with much more flexibility in the ability to customize the part specifically for their application. However, the lower costs deals with ASIC provide a good advantage when viewed from a business point of view. In several cases, the issue can be confounded with the use of embedded cores within an FPGA which are having several vendors offering FPGAs to the market, in which two of the largest, Xilinx and Altera, offering embedded designed cores and configurable cores. Both the companies offer a wide range of choices to designer for use in their FPGA-based system.

The process of implementing a specified design on an FPGA system can be broken down into several stages, approximately definable as design entry or capture, synthesis, and place and route. The designers can make the choice between using schematic or HDL-based design entry comes down to their conception of their specified design. For designer who thinks in software or algorithmic-like terms, HDLs are the better choice. HDLs is most suitable for highly complex designs, especially when the designer has to handle the logic structure. HDLs can also be very useful for designing smaller functions when you haven't the time or inclination to work through the actual hardware implementation.

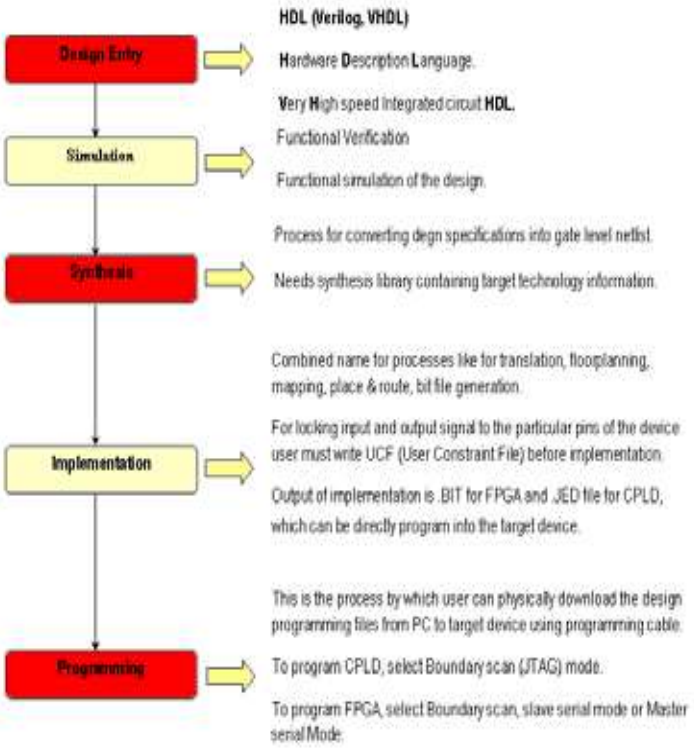


FIG.1. Design Flow for FPGA

RTL Simulation:

The design entry is simulated at the register-transfer level (RTL). This is the first simulation stage, because the design must be simulated at several successive levels of abstraction as it progress toward physical implementation on the FPGA system. RTL simulation offers the highest performance speed. The next step is RTL simulation is to

convert the RTL representation of the specified design into a bit-stream file that can be interfaced to the FPGA. The next step is FPGA synthesis, which translates the VHDL code into a device portlist format that can be understood by a bit-stream converter.

Synthesis Process:
The synthesis process can be broken down into three steps. First, the HDL code is converted into device port list format. Then the resulting file is converted into a hexadecimal bit-stream file, or .bit file. This step is necessary to change the list of required specified devices and interconnects into hexadecimal bits to download to the FPGA. Lastly, the .bit file is downloaded to the physical FPGA system. This final step completes the FPGA synthesis procedure by programming the specified design onto the physical FPGA system.

Gate Level Simulation Process:
Functional simulation is done after synthesis and before physical system implementation. This step ensures correct complex logic functionality. After implementation, there's a final verification step with full timing information. After placement and routing, the logic and routing delays are back-annotated to the gate-level port list for this final simulation. At this point, because of timing, simulation is a much longer process. Often, developer substitute static timing analysis for timing simulation. Static timing analysis calculates the timing of complex combinational paths between registers and compares it against the developer's timing constraints. At the completion of placement and routing, to configure the device a binary programming file is created.

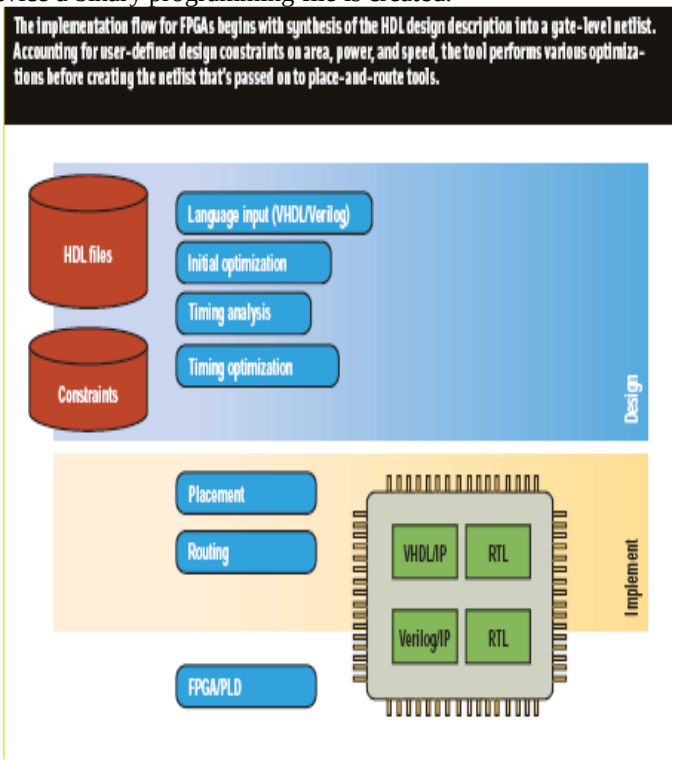


FIG.2. Design Flow for FPGA

VHDL:
VHDL is a hardware descriptive language which is derived from the Ada programming language, its language requirements make it more verbose than Verilog. The extended verbosity is intended to make designs self documenting. Also, the strong typing requires additional coding to explicitly convert from one data type to another data type (integer to bit-vector, for example). The developers of VHDL emphasized semantics that were unambiguous and designs which are easily portable from one tool to the next. Hence, race conditions, as an artifact of the language and tool implementation, are not a concern for VHDL users. Several related standards have been developed to increase the flexibility of the language. Any specified VHDL design today depends on at least IEEE-Std 1164 (std_logic type), and many also depend on standard Numeric and Math packages.
Overview of Spartan-2:

The Spartan 2 designer kit is an easy-to-use evaluation board for the DS33Z11 Ethernet transport-over-serial link device. The Spartan 2 is used with a resource card for the serial link. The serial link resource cards are having transceiver, transformers, and network connections. Dallas' Chip View software is provided with the designer kit, giving point-and-click access to configuration and status registers from a Windows®-based PC. Onboard LEDs shows the received loss of signal, queue overflow, Ethernet link, T_x/R_x, and interrupt status.

Features:
•It can demonstrates Key Functions of DS33Z11 Ethernet Transport Chipset
•It includes Resource Card with DS21348 LIU, DS2155 T1E1 SCT, and DS3170 T3E3 SCT, Transformers, BNC and RJ48 Network Connectors and Termination
•It also provides Support for Hardware and Software Modes
•On-Board MMC2107 Processor and Chip View Software Provide Point-and-Click Access to the DS33Z11, DS2155, and DS3170 Register Sets
•All DS33Z11 Interface Pins are accessible for External Data Source/Sink
•It is also having LEDs for Loss-of-Signal, Queue Overflow, Ethernet Link, T_x/R_x, and Interrupt Status
•It is Easy-to-Read Silk Screen Labels Identify the Signals associated with all connectors, Jumpers, and LEDs
Design kit contents:
•DS33Z11DK Main Board
•Single-Port Serial Card with DS2155 T1/E1 SCT, DS21348 T1/E1 LIU, and DS3170 T3/E3 SCT
•CD_ROM
•Chip View Software and Manual
•DS33Z11DK Data Sheet
•Configuration Files

The basic layout is having all the input, output, processing, and storage blocks which includes LEDS and Test point block, the SDRAM block, hardware mode switches and a serial port for programming using a JTAG and then the main processor or FPGA part which is interfaced with the remaining devices as shown above. The storage and output module contains the HEX-LED used to display the alphanumeric value of the system's timing parameters and the RAM which are used to store those parameters.

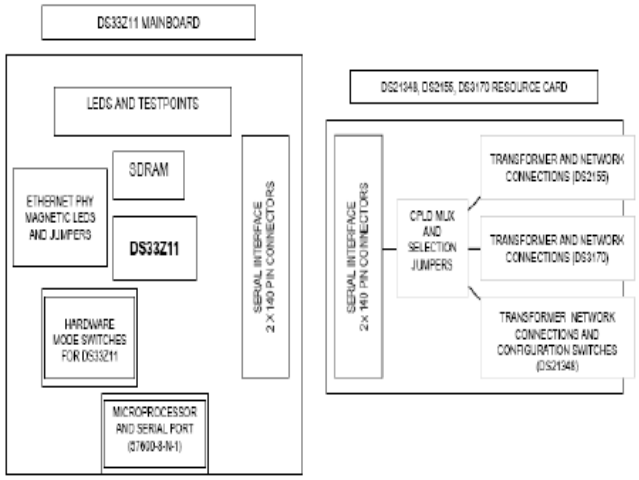


FIG.3. Floor plan of Spartan 2

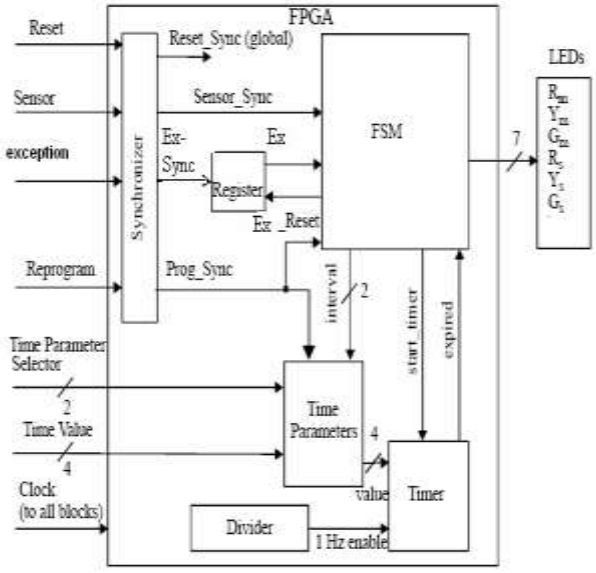


FIG.4.Block Diagram of TLC

The display on the Spartan kit was done using the HEX-LED pad .The system uses pin numbers attribute in top level VHDL source file to force the VHDL compiler to allots the data lines for the RAM data bus on a particular set of pins.

OVERVIEW ON MICROCONTROLLER SYSTEM

In this section, the IR sensor and wireless network is used in traffic light control system. A lot of progress can be gained in this area, and intelligent traffic control gained interest of several governments and commercial companies. ITS research includes car safety systems; simulating effects of the changes urban infrastructure, route planning, optimization of transport, and smart infrastructures. Its main goals are to improve safety, minimizing travel time, and increasing the capacity of infrastructures. Such developments are advantageous to health, economy, and the environment, and this shows in the specified budget for ITS.

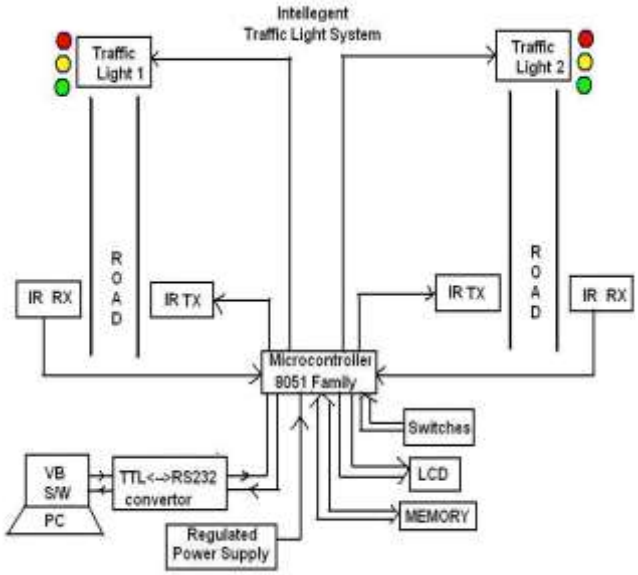


FIG.5. Block Diagram of system

Design and development of the system:

Development of the specified intelligent traffic light controller and monitoring system having lots of study and implementation work. The implementation work is been overviewed and divided into points as discussed below.

Power Supply:

As per the power requirement, the hardware of the specified intelligent traffic light control and monitoring system includes supply of +5V w.r.t. GND is developed as shown in Figure 6.

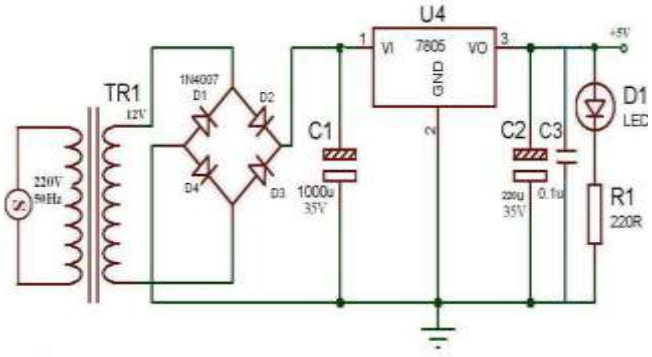


FIG 6. Circuit Diagram of Power Supply

The complete circuit is having TTL logic level of 0-5V. It comprise of 0V to 9V transformer to step down the 220V AC supply to 9V AC. A bridge rectifier converts the 9V into 9V√2 DC. It is again filtered through a 1000uF capacitor, then regulated using 7805 to get +5V. Further filtering 220uF capacitor is done to isolate the output voltage of +5V from noise [14].

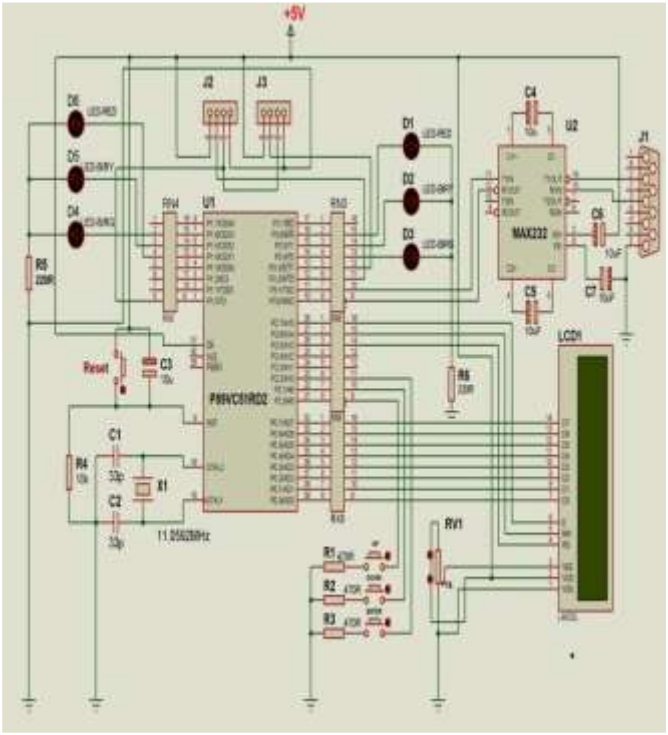


FIG.7. Circuit diagram of microcontroller board

The circuit shown in the Figure 7 is complete circuit diagram of intelligent traffic light controller system which is showing the interfacing of some peripherals and ICs with the microcontroller P89V51RD2. This circuit is made in the software Proteus, which is basically a circuit making and simulation window based software. Microcontroller receives the 11.0592MHz from the crystal oscillator at XTAL1 and XTAL2 pins. Reset switch is connected at pin 9 of micro-controller provides manual reset of the microcontroller. Pull-up network resistances of 10K are provided at each port for properly differentiating between high and low TTL signal. LCD display is connected with its three control signal RS (Resister Select), R/W (Read/Write) and E (Enable) is used to display the outputs and status messages for the user record. Port P0 is used to provide data parallel to LCD to display as character. RV1 potentiometer controls the LCD contrast [14].

Figure 8 shows circuit diagram of IR transmitter and receiver situated on road. The system made here monitors and controls the traffic movement for two roads respectively. Connectors J1 and J2 of microcontroller board connects the IR sensor on road with the microcontroller board. Pin P1.0 of port P1 drives the IR transmitter by generating 38KHZ 50% duty cycle square wave. The IR sensor's output is connected to pin P3.2 and P3.3 for both traffic light respectively. P1.0 control the IR LED D1 through Darlington pair formed by the transistors Q1 and Q2. The Darlington pair just amplifies the current through IR LED. The IR sensor TSOP1738 detects the IR wave from IR LED and provides active low output at its pin3. So whenever any vehicle passes through IR LED and IR sensor, the TSOP1738 pro-vide high output at its pin 3.

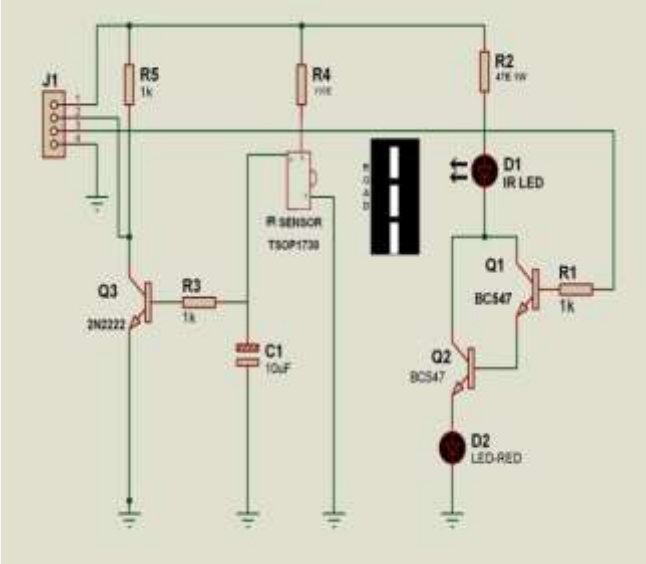


FIG.8. Circuit Diagram of Sensor on road

Development of Software of Intelligent Traffic Light Control and Monitoring System:

Assembler:

Assembler is used to convert the assembly language code to machine level code. A51 assembler is used for this purpose. Figure 6 shows the A51 assembler GUI.

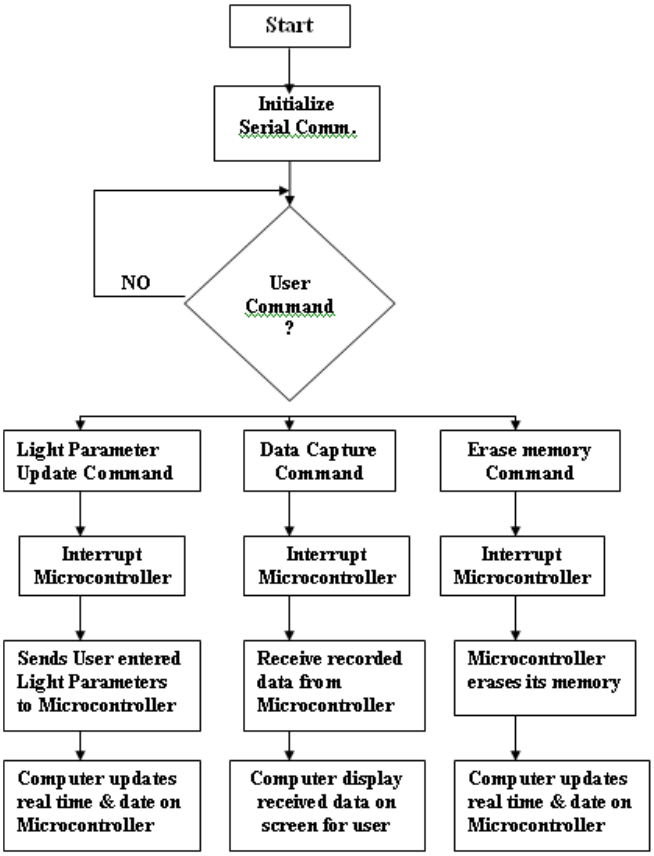
In-System Programmer:

P89V51RD2 microcontroller is having the feature of In-System programming. Hence it does not require any separate hardware programmer to get programmed. It includes an inbuilt boot program, which helps it during programming, erasing and in-application programming. The object code formed by the assembler or cross compiler is loaded into the software called IN-System programmer. Figure 10 shows a Flash magic in-system programmer. This window based programmer communicates with the microcontroller boot program through serial port.

Computer Software:

The software for recorded data monitoring and system control is made in visual basic. The flowchart of the VB program is given aside. Data transfer is initiated from computer through a user click in Visual Basic, which in turn requests the microcontroller serially. The microcontroller during its main loop continuously polls the serial port, as in flow chart. Microcontroller after getting the data transfer request from PC stops its recording task temporarily and switches to data transfer mode. 9600 baud rate is used in this communication. Serial port communication parameters are initially configured for 9600 baud rate, with 8 bit data word, no parity, and 1 stop bit.

ISSN: 2277-5528
Impact Factor: 2.745 (SIJF)



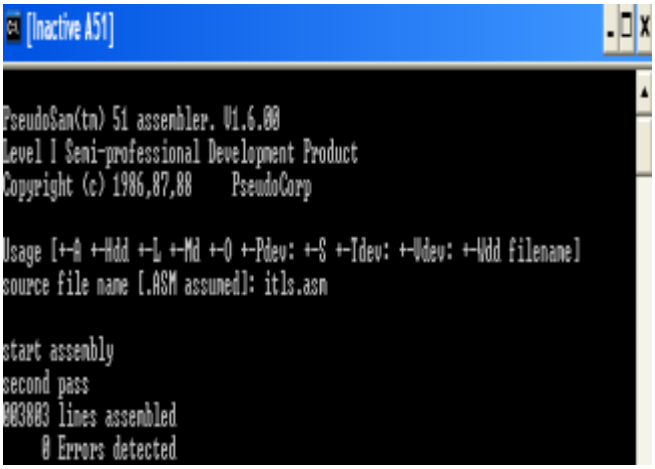


FIG.9. Assembler GUI

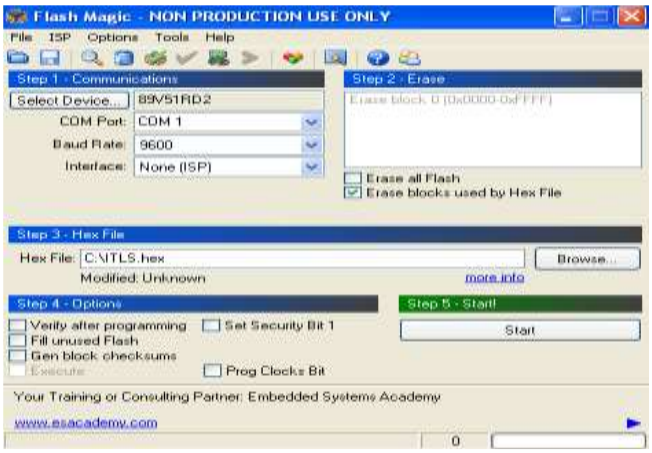


FIG.10. In-system Programmer GUI

SIMULATION RESULTS AND COMPARISON OF BOTH SYSTEMS

There are several problems and errors using the course of designing and building this traffic light controller,. Firstly when we consider FPGA system these included errors that unable to fit design into the FPGA, having FSM transition into unexpected states at unexpected times, and having the system occasionally not see the expired signal generated by the timer module. FPGA uses a very specialized path for clock signals for all of its flip flops. This makes it uneasy to route internally generated signals to the clock inputs of other components on the same FPGA. After lot of attempts we noticed that the clock frequency of the FPGA has to be considered and so output of the clock generator and divider modules as enable signals. The FPGA hardware is able to route and generate flip flop enable signals much better than clock signals. The next major problem was in correctly handling the expired and starts timer signals. As explained in paper [3], we are using the Xilinx 8.2 ISE simulator for the testing and code building purpose. The simulator had too many functionalities to know all and so we had to get completely acquainted with the simulator before starting of and the debugging process to was a tough because the simulator we felt is not that user friendly as it does not give proper solution to many errors which are internal in nature and not related to the code

In second case when we are considering microcontroller system Results include the successful operation of the intelligent traffic light control and monitoring system. The IR sensor with IR transmitter is placed at a gap. Gap acting as a prototype indicating a road. The system is placed near road as a standalone device. Whenever any obstacle like vehicle passes between IR transmitter and IR sensor, microcontroller detects and increase number of vehicle count in a recording interval for particular traffic light. Traffic light is placed ahead of IR sensor at a distance so that decision taken by microcontroller to control traffic light can help in reducing the congestion at traffic light. On basis of analysis administrator can command the microcontroller of update the configuration of parameter to respective lights. The graphical user interface to update con-figuration, By using this system configuration author [14] tries to reduce the possibilities of traffic jams, caused by traffic lights, to an extent and successfully gets the results.

CONCLUSION & FUTURE SCOPE

Many such implementations are possible wherein we try to overview robust embedded processors for the purpose of controlling and handling of variety of timing tasks implementing all the features of an FPGA into the system making the system robust, precise and accurate. Here we have made comparison between two such embedded systems that deal with traffic light controllers as a part of our work.

In future these systems can be used to inform people about different places traffic condition data transfer between the microcontroller or FPGA and computer can also be done through telephone network , data call activated SIM These technique allows the operator to gather the recorded data from a far end to his home system without going there. Traffic lights can be increased to N number and traffic light control can be done for whole city by sitting on a single place or automatically.

REFERENCES

1. Design of FPGA-Based Traffic Light Controller System M.F.M.Sabri, M.H.Husin, W.A.W.Z.Abidin, K.M.Tay H.M.Basri,Dept. of Electronics engineering, Faculty of engineering University Malaysia Sarawak, Sarawak Malaysia msmfaizrizwan@feng.unimas.my, 978-1-4244 87288-8/11
2. F Jose E. Ortiz and Robert H. Klenke. "Simple Traffic Light Controller: A digital system design project " Southast Con 2010 (Southeas tCon), Concord, NC, March 2010, pp. 85-88
3. K.Shriniwas,Shrikant.P.KA.S.-THESIS REPORT (N.I.T.,ROURKELA) -Implementation of FPGA design in embedded system
4. Reference book of Digital design: Morris Mano M
5. Reference book of Digital design: Frank Vahid
6. Roess, R. P., and W. R. McShane. 1998. Traffic engineering. 2d ed. Engelwood Cliffs,NJ: Prentice Hall.
7. Clement, S., and J. Koshi. 1997. Traffic signal determination: The cabal model. 1997
8. Genetic Algorithms in Engineering Systems: Innovations and Applications Conference, pp.63-68.
9. McShane, W., and R. Roess. 1990. Traffic engineering. Engelwood Cliffs, NJ: Prentice Hall.
10. Kuwahara, M., and M. Koshi. 1990. Decision of timings of signal program switching in
11. Pretime multi-program control. In Transportation and traffic theory, edited by M. Koshi,531-50. New York: Elsevier.
12. Nakats, T., S. Seki, S. Shibliya, and I. Tkaku. 1994. Artificial intelligence approach for optimizing traffic timings on urban road networks. Proceedings of the 1994 Vehicle Navigation and Information Systems Conference, pp. 199-202.
13. Wilshire, R., R. Black, R. Grochoske, and J. Higinbotham. 1985. Traffic control systems handbook. FHW A-IP-85-12. Washington, DC: Federal Highway Administration, U.S. Department of Transportation.
14. Promila Sinhmar-Intelligent traffic light controller and density controller using IR sensor and microcontroller, IJATER Volume 2 issue 2 March2012.