ISSN: 2277-9655
Impact Factor: 4.116

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

A REAL-TIME REMOTE CONTROL AND MONITORING SYSTEM USING ZYNQ SOC FPGA BASED WEB SERVER

M. A. Majeed*, B. Vamsi Krishna, Dr.M. Kama Raju
* PG Scholar, ECE, Gudlavalleru Engineering College, Gudlavalleru, India
Assistant Professor, ECE, Gudlavalleru Engineering College, Gudlavalleru
Professor and HOD, ECE, Gudlavalleru Engineering College, Gudlavalleru, India

DOI: 10.5281/zenodo.59670

ABSTRACT

For the economic development and progress of any country, roads play an important role by providing transportation ease for goods as well as for passengers. The number of vehicles on road is increasing day by day which are controlled by the traditional Traffic Light Controller in the countries like India and other developing countries. In traditional Traffic Light Control system, fixed time is allotted for traffic on each road to pass irrespective of the traffic density. This is an inefficient controlling method which consumes time, effort and fuel of users unnecessarily. But while coming for control system implementation, they consist of parallel operations and conditional executions which cannot be effectively performed on present day FPGAs. This problem is addressed by using Real-Time Traffic Light Control system on Zynq Soc FPGA. This new generation of reconfigurable FPGA has experienced a rapid development in a present day environment which was accompanied by an increased need to monitor, maintain and update of the system. On the other hand, Web technology has become widely used with an ascending trend to migrate everything online and to remotely control systems.

This paper describes an approach for the development of a real-time remote control and monitoring system using an FPGA based web server implemented on a Zynq SoC FPGA, which involves Client-Server architecture that act as operating system. The open source Light weight IP (LwIP) standalone TCP/IP protocol stack is used to implement networking capability and the FPGA based web server enables to access the control system by using a web browser to send the information and updates the system.

KEYWORDS: FPGA WebServer, LightWeight IP(LwIP), Zynq SoC FPGA.

INTRODUCTION

The reconfigurable System like FPGA platform have the potential to provide the performance benefits of ASICs and the flexibility of processors. As FPGAs can provide trade-off between performance and flexibility, they become the primary source of computation in many critical embedded systems[1]. The Zynq-7000 family is based on the Xilinx All Programmable SoC (APSoC) architecture is shown in Fig.1. The Zynq-7000 family offers the flexibility and scalability of an FPGA. These products integrate a feature-rich dual-core ARM Cortex-A9 based processing system (PS) and 28 nm Xilinx programmable logic (PL) in a single device. The ARM Cortex-A9 CPUs are the heart of the processing system and include on-chip memory, external memory interfaces, and a rich set of peripheral connectivity interfaces such as Ethernet (LAN) is used for general input/output functionality and PMOD's are used as transceivers.

ISSN: 2277-9655
Impact Factor: 4.116

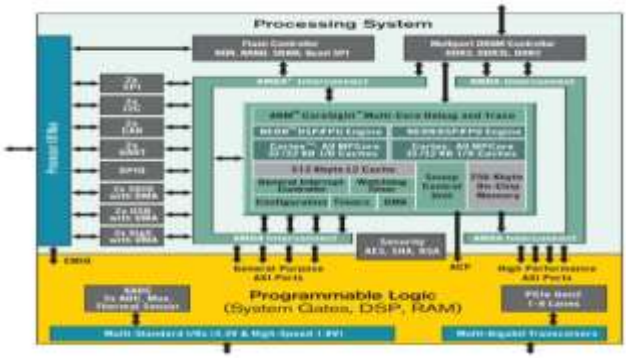

Fig.1. Zynq SoC FPGA Architecture

The hardware and software platform architecture will be based on Xilinx All-programmable technologies and solutions by exploiting all the features of Xilinx's Zynq FPGA. Thereby, the embedded Web server is designed as a unified system which includes both hardware and software components [1].

- Hardware platform is mainly based on a Dual core ARM cortex A9 hardcore processor with Ethernet controller of 10/100 Mbps.
- Software platform is based on Client-Server architecture and LwIP standalone protocol stack.

The objective of this paper is Zynq Processor Configuration for LwIP stack used for Ethernet connectivity, and development of application program on LwIP stack.

A Client-Server architecture is created on LWIP which is used for communication between a Web Server and controller, then finally a real time traffic light control system is developed in this design and verified through hardware.

This paper is organized as follows: In section 2 an overview of some related works in embedded Web server is provided, ZYNQ processor configuration for Ethernet connectivity is explained in section 3, In section 4 a Real time traffic control application is specified, Experimental results are addressed in section 5, Conclusion in section 6.

RELATED WORK

In [1] Ahmed Hanafi and Mohammed Karim proposed an Embedded Web Server for Real-time Remote Control and Monitoring of an FPGA-based On-Board Computer System.several. In this several studies have emerged suggesting the implementation of the embedded Web server technology to provide a monitor and control system based on an Internet browser and an embedded TCP/IP protocol suite. The article aims to contribute to the efforts of designer community to demonstrate the effectiveness of using Web access ability to improve communication with embedded platforms.

Mr. Shashikant, V. Lahade, Mr. S.R. Hirekhan, proposed an Intelligent and Adaptive Traffic Light Controller (IA-TLC) using FPGA [2]. The problem is addressed by implementing on FPGA using verilog as an example of FSM with 35 states. In IA-TLC density of traffic is sensed by using IR sensors throughout day and night, and accordingly time is allotted for users to pass. Other advantages of the proposed system are: i) System senses emergency vehicles on the individual road moreover it gives priority to the traffic of that particular road where the emergency vehicles is sensed. ii) Finds out defaulter who crosses the red signal by capturing images using camera.

As the embedded web server technology is the combination of embedded hardware device and embedded software that provides a flexible management function for control and monitoring features, networking capabilities. Jie Xiao and Fen Shi Zengi proposed a Design and implementation of embedded web server[3], which represents a remote monitoring, control and diagnostic system based on an embedded boa server transplanted and built under an ARM Linux operation system, which takes ARM processor as a hardcore processor.

ZYNQ PROCESSOR CONFIGURATION

The ZYNQ Processor configuration is performed by using Xilinx Platform Studio (XPS) Tool Suite provided by the Embedded Development Kit (EDK) of Xilinx. The Xilinx EDK is a design suite of tools and Intellectual Property (IP) that enables to design a complete embedded processor system for implementation in a Xilinx Field Programmable

Gate Array (FPGA) device. Whereas the Software Development Kit (SDK) is an integrated development environment used for C/C++ embedded software application creation and verification.
The following steps resemble the steps to configure the embedded processor:

- The use of XPS primarily is for embedded processor hardware system development. Specification of the microprocessor, peripherals, and the interconnection of these components along with their respective detailed configuration take place in XPS.
- The use of SDK is for software development. SDK is also available as a Standalone Application.
- Once the Zynq FPGA is configured with the bit stream containing the embedded design, download and debug the Executable and Linkable Format (ELF) file from Software project created within SDK.

The Xilinx Platform Studio is used to configure and build the hardware specification with the processor core, memory-controller, I/O peripherals, etc. This Xilinx Platform Studio converts the designer's platform specification into a synthesizable RTL description and writes a set of scripts to automate the implementation of the embedded system from RTL to the bit stream-file.

In this section, the first part resembles with the Zynq processor configuration for LWIP stack to provide Ethernet connectivity. So to configure the peripherals with the processor, import Zynq configuration files from the Xilinx 14.7 ISE tool which are represented with .xml format. Then select the required peripherals like Enet, UART, GPIO and a Timer peripheral. So, after configuring the required peripherals to the processor, the configured Zynq architecture looks as shown in Fig.2

With the completion of required peripherals to the processor, configure the number of GPIOs required. In this design 4GPIOs are necessary to provide communication from processing system to programmable logic and from programmable logic to processing system. Once the GPIOs are configured based on the requirement then generate netlist. Once the netlist is generated in XPS, then go to ISE and generate top HDL source which creates system top (processor top). The port mapping of GPIOs is in Fig.3.

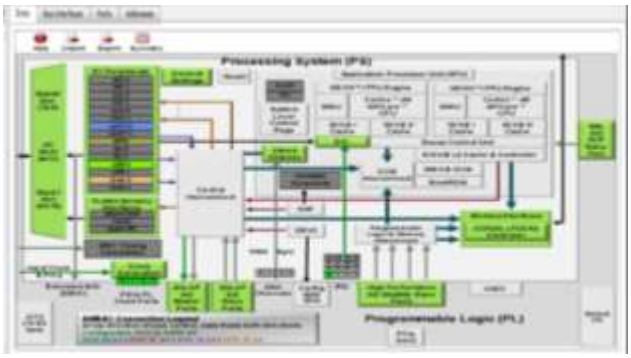

Fig.2. Zynq Processor After Peripheral Configuration

Fig.3. Bus Interconnection for LWIP in Zynq

ISSN: 2277-9655
Impact Factor: 4.116

After the completion of all these steps, generate programming file (bit file) for the system top.
The second step of this section deals with SDK. Here, the RTL design is exported to SDK for software application development and the communication between RTL and SDK is also developed in c-code. The software development of LwIP is in Fig.4.

Fig.4. Software development for LwIP

REAL TIME TRAFFIC CONTROL APPLICATION

On account of tremendous growth in the industrial sector, people need to travel from one place to another by road for employment and social activities [2]. Also, in the economic development and progress of any country, roads play an important role by providing transportation ease for goods as well as for passengers.
In this paper, a real time traffic light control system is implemented an application on configured Zynq FPGA by considering a single junction. The application is developed by considering all the possible ways to clear the traffic. The possible traffic modes are normal mode, low density mode, high density mode, emergency mode and free zone mode.

A. Operation of Traffic Control Modes

The basic operation of traffic control modes is shown in the Fig.5. It consists of a start signal and four enable signals.

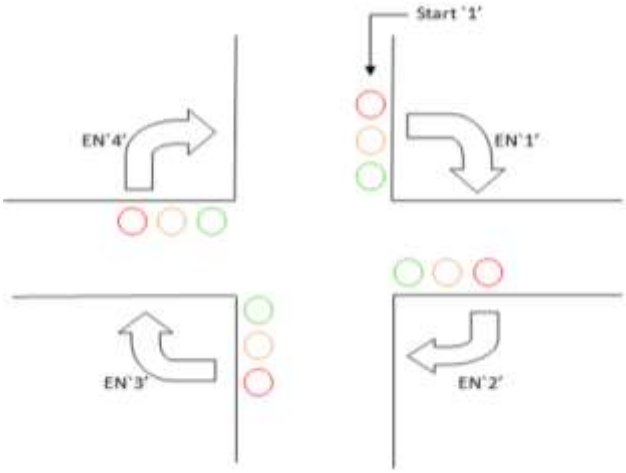

Fig.5. Operation of Traffic Control Mode

The operation of each mode starts with a Start signal. Once the start signal goes high, the green light glows for the prescribed time period until the counter runs. After the completion of relevant time for green light, an orange light glows in the same direction. With the completion of counter time for green light and orange light, the counter generates an enable signal as en1 to shift the direction for other side. By generating an enable signal in the current direction, the

present state goes with red indication. In the second direction, the same operation continues and after the completion of the prescribed time in the second direction, it generates an enable signal as en2. With the generation of enable signal en2, the counter shifts to third direction by enabling red signal in the present state.

The same operation continues in the remaining two directions and at the end of controller time in fourth direction, the counter shifts to first direction. Then after, the first direction starts its signaling after receiving the enable signal en4. In this way the designed traffic modes will perform their operation.

B. Client

Here the web page is designed in html language with a unique IP address as 192.168.1.150. The Zybo board includes Zynq FPGA act as web server and perform its operation when it receives the commands from html code. The purpose of web page is to monitor the real-time traffic and update the system according to current traffic density. The designed web page as a client to control the traffic is in Fig 6.

Fig.6. Web Page Shows Traffic Information

C. Pmod Connectors

The hardware used to implement the design is ZYBO board. This Zybo board have six pmod connectors, each Pmod connector falls into one of four categories: standard, MIO connected, XADC and high-speed Pmods. The purpose of Pmod connectors is to interface the server with display unit. In this application, a standard Pmod connector and high-speed Pmod connectors are used.

Therefore, the client as a controller will monitor the current status of the traffic through web page and selects the required traffic mode in the web page. Then web server receives the commands through Ethernet connectivity and performs the relevant operation which can be observed in the display unit.

EXPERIMENTAL RESULTS

The application is simulated using ISim and the results are verified through hardware.

A. Simulation Results

The logic is simulated using ISim by giving the inputs through test bench. The simulation results for logical analysis are shown below.

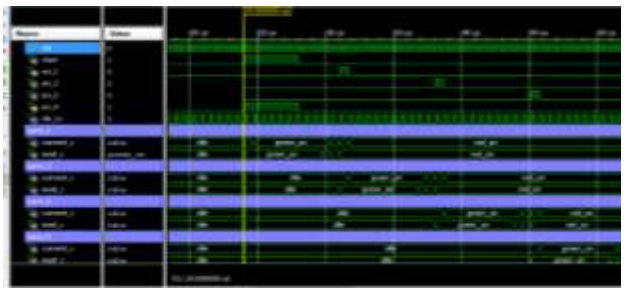

Fig.7. Simulation Result for North Signalling

ISSN: 2277-9655
Impact Factor: 4.116

Fig.8. Simulation Result for South Signalling

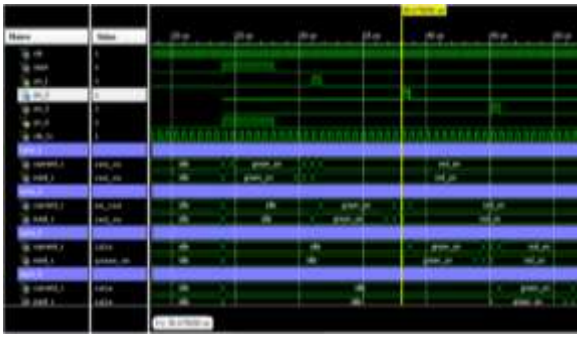

Fig.9. Simulation Result for East Signalling

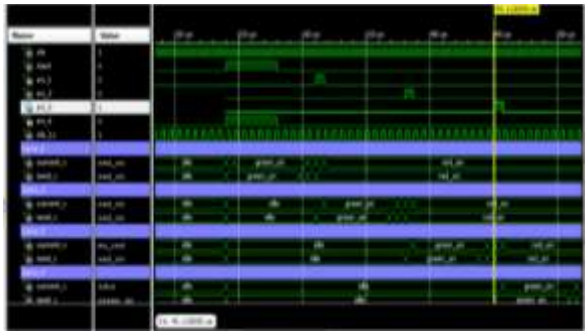

Fig.10. Simulation Result for West Signalling

B. Hardware Results

For normal mode, the web page and its hardware result are shown in Fig.11 and Fig.12.

Fig.11. Normal Mode Set by Client in Web Page

ISSN: 2277-9655
Impact Factor: 4.116

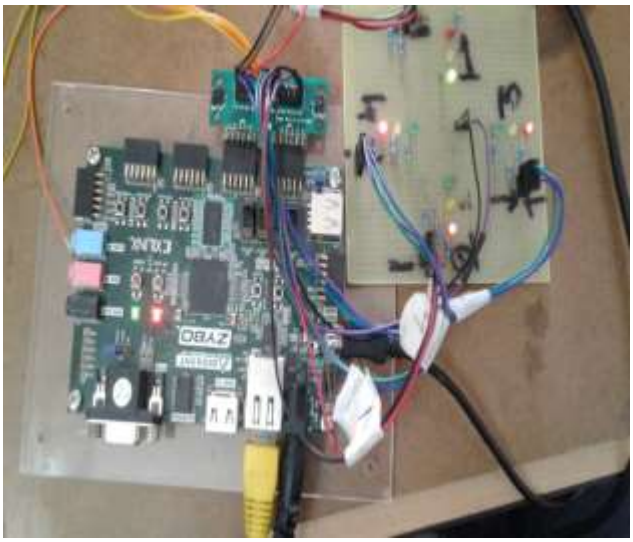

Fig.12. Normal Mode operation of North signalling

For Low-density mode, the inputs are given from Chip Scope VIO to the hardware and its web page is shown in Fig.13.

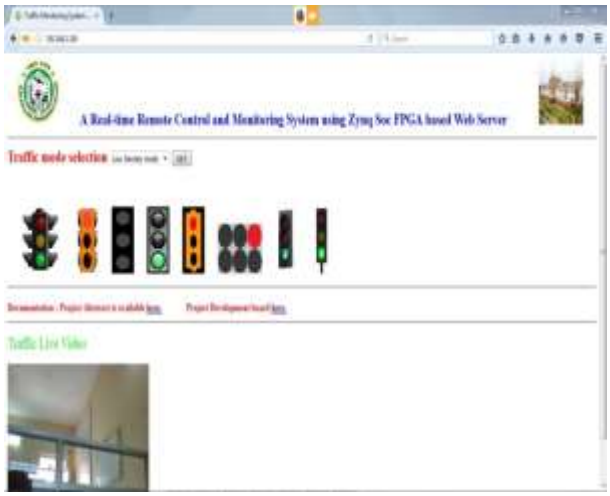

Fig.13. Low-density Mode Set by Client in Web Page

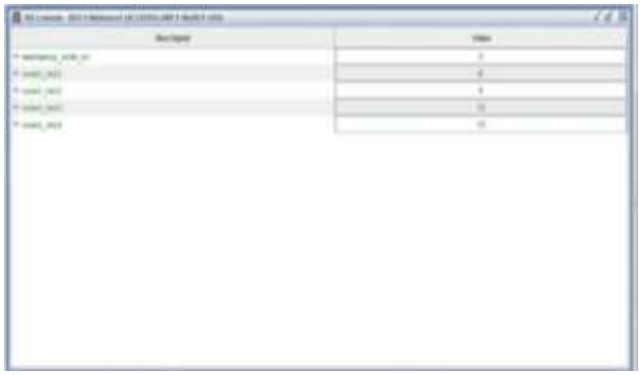

Fig.14. VIO for Low-density Mode

ISSN: 2277-9655
Impact Factor: 4.116

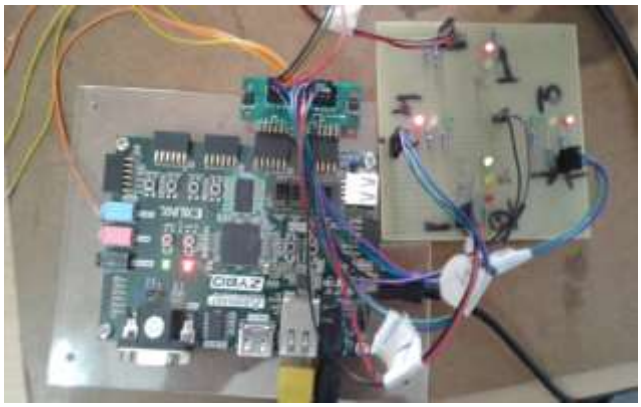

Fig.15. Low-density Mode Operation of South Signalling

For High-density mode also, the inputs are given from Chip Scope VIO to the hardware and its web page is shown in Fig.16.

Fig.16. High-density Mode set by client in web page

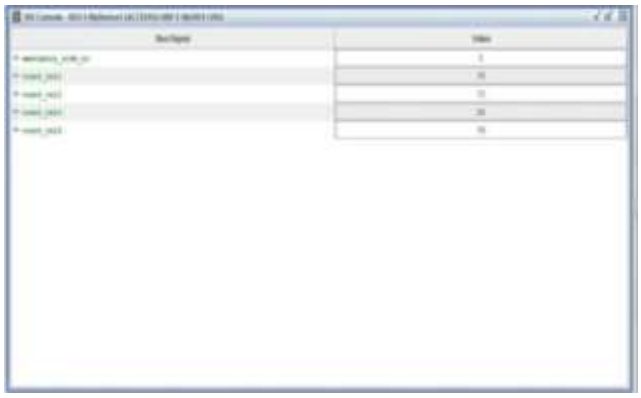

Fig.17. VIO for High-density Mode

ISSN: 2277-9655
Impact Factor: 4.116

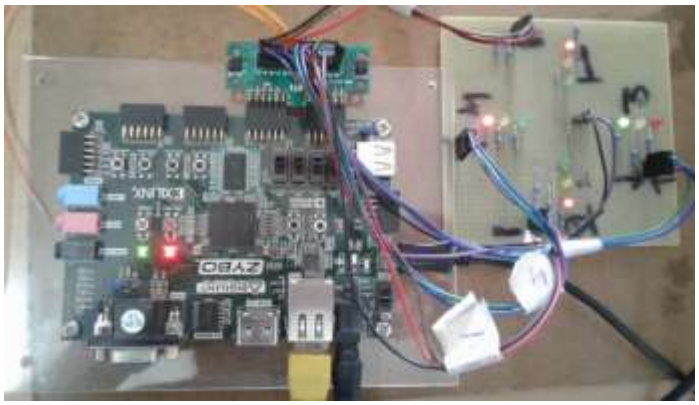

Fig 18: High-density Mode Operation of East Signalling

For Emergency mode, direction is set in VIO and its web page is shown in Fig.19.

Fig.19. Emergency Mode Set by Client in Web Page

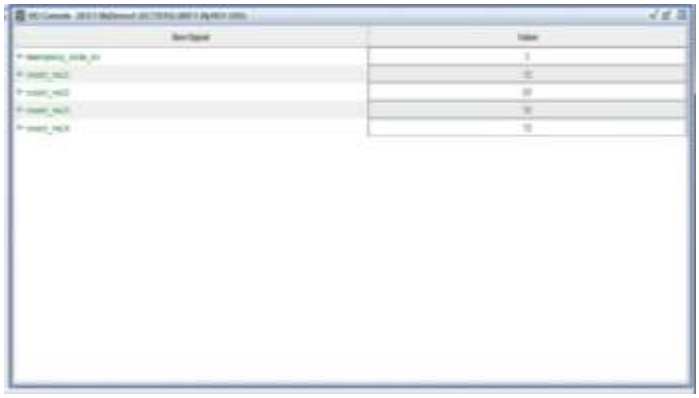

Fig.20. VIO for Emergency Mode set For West Direction

ISSN: 2277-9655
Impact Factor: 4.116

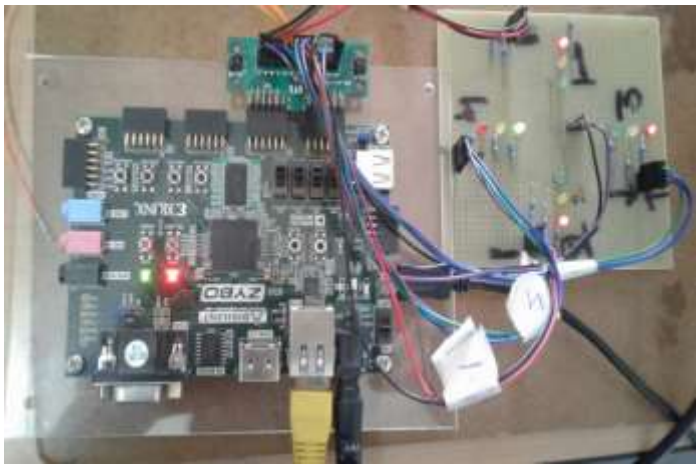

Fig.21. Emergency Mode Operation of West Signalling

For Free Zone mode, the web page and its hardware result are shown in Fig.22 and Fig.23.

Fig 22: Free Zone Mode Set by Client in Web Page

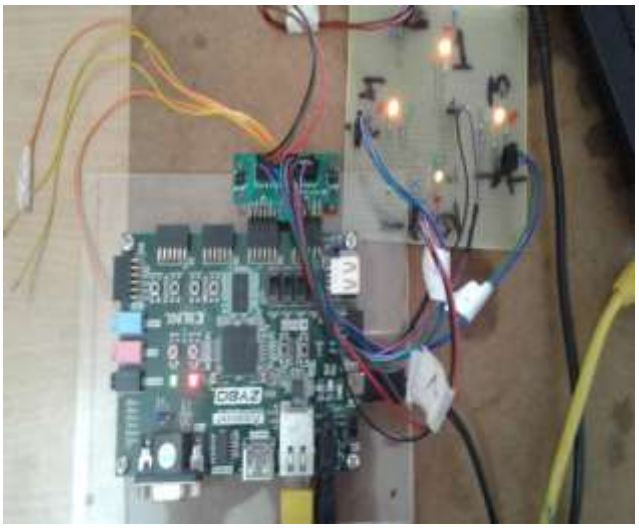

Fig 23: Hardware Result for Free Zone Mode

CONCLUSION

This paper demonstrates successfully a method concerning the implementation of Xilinx Zynq FPGA based Web server and an application of real-time traffic light control system developed by configuring the processor present in Zynq FPGA. The software platform is verified through ISim and hardware platform is verified through ZYBO board consisting of Zynq FPGA.

REFERENCES

[1] Ahmed Hanafi and Mohammed Karim, "Embedded Web Server forReal-time Remote Control and Monitoring of an FPGA-based On-Board Computer System," Intelligent Systems and Computer Vision (ISCV). IEEE, 2015, pp.1-6, 25-26 March 2015.

[2] Mr. Shashikant, V. Lahade, Mr. S.R. Hirekhan, "Intelligent and Adaptive Traffic Light Controller(IA-TLC) using FPGA," Industrial Instrumentation and Control (ICIC), 2015 International Conference on 28-30 May 2015.

[3] Jie Xiao and FenShi Zeng, "Design and implementation of embedded Web server," Computer Science & Education (ICCSE), 2012, 7th International Conference, pp.479-482, 14-17 July 2012.

[4] R. Sharan Soni and D. Asati, "Development of Embedded Web Server Configured on FPGA Using Soft-core Processor and Web Client on PC," International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-1, Issue-5, June 2012.

[5] WM EI-Medany, &MR Hussain, "FPGA-Based Advanced Real Traffic Light Controller System Design", 4th IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems:Technology and Applications proceeding, ISBN: 978-1-4244-1347-8, pg. 100 - 105,2007.

[6] Taehee Han; Chiho Lin, "Design of an intelligence traffic light controller (ITLC) with VHDL," *Proceedings 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering (TENCON '02)*, 28-31 Oct. 2002, vol 3, pp:1749 -1752.

[7] Arash Farhadi Beldachi, Jose L. Nunez-Yanez, "Accurate Power control and monitoring in ZYNQ boards," Field Programmable Logic and Applications (FPL), 2014 24th International Conference, pp.1-4 on 2-4 Sept. 2014.

[8] Tao Xue, Weibin Pan, Guanghua Gong, Ming Zeng, Hui Gong, Jianmin Li, "Design of Giga bit Ethernet readout module based on ZYNQ for HPGe," ," Real Time Conference (RT), 2014 19th IEEE-NPSS.IEEE, pp 1-4, 26-30 May 2014.

[9] Stephen MacMahon, Nan Zang, Anirudha Sarangi "LightWeight IP(lwIP) Application Examples", XAPP 1026 (V3.1) , April 21, 2011.

[10] Mrinal J. Sarmah and Radhey S. Pandey "System Monitoring using the Zynq-7000 AP SoC Processing System with the XADC AXI Interface**",** XAPP1182 (v1.0) November 18, 2013.