International Journal of Electronics and Communication Engineering.
ISSN 0974-2166 Volume 8, Number 1 (2015), pp. 75-80
© International Research Publication House
http://www.irphouse.com

Implementation Of Ethernet Based Data Transfer Using FPGA

Shruthi Sangani (M.Tech), K.Vikram and S.R. Pankaj Kumar

Dept. of E.C.E (VLSI Design) GITAM University
Visakhapatnam, AP – 530045 India.

Sc – „D ‟, ES – II, ESM Wing DLRL, Chandrayangutta
Hyderabad, Telangana – 500053 India.
Sc – „F ‟, ES – II, ESM Wing DLRL, Chandrayangutta
Hyderabad, Telangana – 500053 India

Abstract:

This paper presents the implementation of embedded processor inside FPGA (Field Programmable Gate Array) such that it can receive Ethernet packets, extract the actual data, process it and finally transmit it to other subsystems if required. The implementation platform is a development board which has a Virtex-5 FPGA, ML507 in specific. Using the XPS (Xilinx Platform Studio) tool, the embedded processor PowerPC 440 is configured inside the Virtex-5 FPGA. The software part of the processor is configured in SDK (Software Development Kit). The implementation requires ML507 development board, Ethernet cross cable, RS232 serial cable, Power Supply, Timer and a High end PC. After establishing the link between the PC and development board using Ethernet interface, the commands are sent from the SCD (System Controller Display) to the Board which are packed according to the IRS (Interface Requirement Specifications) and are sent to the board and the output is given to the GPIO (General Purpose Input Output) pins as well as the other subsystems.

Keywords: FPGA, PowerPC 440, Virtex-5, EDK, XPS, SDK, lwIP.

I. Introduction

FPGAs are semiconductor devices that are based around a matrix of CLBs (configurable logic blocks) connected via programmable interconnects. FPGAs can be reprogrammed to desired application or functionality requirements after manufacturing. With the advancement of FPGAs a new trend of implementing the microprocessors on the FPGAs has emerged in the design community. In Ethernet communication, the data is secured with a header, so this requires a technique which is used to extract the actual data from the Ethernet data. If a processor is used, this can be done quite easily with good speed and also the actual data can be further processed if required. The implementation platform is the ML507 board which supports the embedded processor PowerPC 440. PowerPC is a superscalar 32-bit embedded processor developed by IBM. It has a 32 KB instruction cache, 32 KB data cache, 128-bit PLB (Processor Local Bus) and a High-speed memory controller interface. The implementation is achieved using the Embedded Development Kit provided by Xilinx, which helps to design the complete embedded processor more quickly and easily. The project block diagram is shown below in Fig. 1.

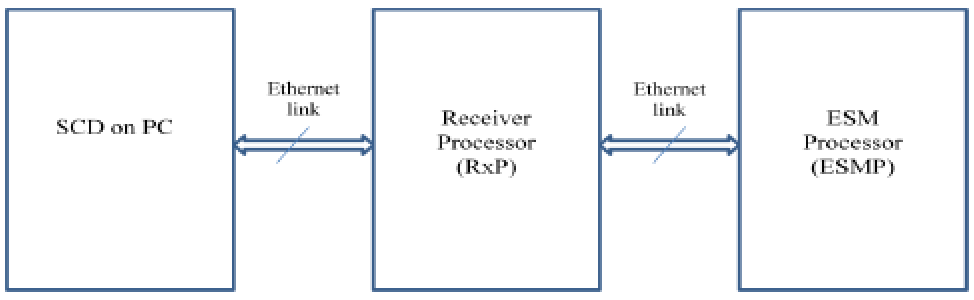


Fig. 1 Block Diagram

II. Experimental setup

The implementation platform is the ML507 development board which has a Virtex-5 FPGA. The ML507 board supports PowerPC 440 embedded processor. The system design is divided into two -

- One is the Hardware design, which includes the designing methods using XPS, EDK and
- Other is the Software design, which includes the designing methods using SDK.

A. Hardware Design

Apart from PowerPC 440, other components required by the design are a Timer, Block RAM interrupt controller, RS232, Ethernet MAC and a memory device DDR_SDRAM. The Base System Builder (BSB) Wizard inside XPS tool is used for generating the embedded system that is supported on the ML507 board. This wizard is used for the selection of the board, processor, cache and the required peripherals. The hardware design also includes the bus interfaces and ports information too. Lastly, the addresses are automatically generated and the “system.bit” file is created which includes the entire hardware design of the board and also the “system.bmm” file is created which includes the block RAM memory mapping configuration. The GPIO pins are used for checking the output using oscilloscope where the input commands are initiated using SCD.

B. Software Design

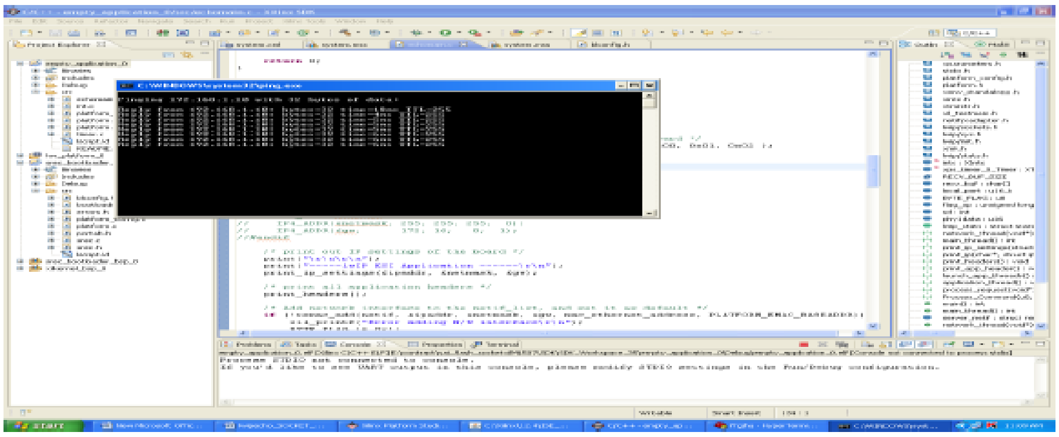
The software part of the design is configured using the SDK tool. In order to configure the software platform, the entire hardware design is launched and exported to the SDK environment. In SDK, the required operating System is selected along with the Light Weight Internet Protocol (lwIP). Now, to meet our application requirement, a C- program is written wherein the server works on TCP data and listens for the input at the specified port and simply echoes back whatever data is sent to that port. In our design, the IP address 192.168.1.10 is binded with our board‟ s MAC address. Also our design is configured such that it will be listening for the input at the port 5550. After the data is echoed back from the board to the PC, another embedded application software program is written in order to send the commands initialized from the SCD to the Target Board. The output is finally verified using the GPIO pins and is forwarded to other subsystems if required.

III. Implementation

A. Data Transfer Using Echoback

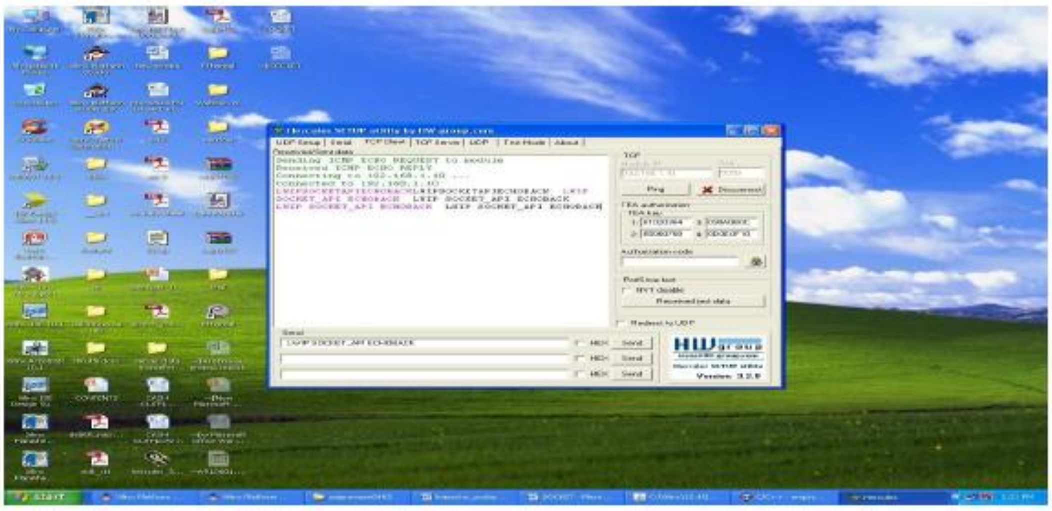
The FPGA board is connected to an Ethernet port on the host computer via an Ethernet cable. Next an IP address is assigned to the Ethernet interface on the host computer. The IP address of the PC and the board must be in the same subnet. The software application assigns a default IP address of 192.168.1.10 to the board. So in our design the PC is assigned with the IP address 192.168.1.11. Lastly, the application software is stored in the nonvolatile memory like Flash (or) PROM for permanent storage.

The C-program written in SDK is compiled with the GNU Compiler tool. The compiled C-files along with the libraries generate the “system.elf” (Executable and Linkable File). The final stage of designing is the association of the hardware and software platforms and the downloading of the entire image into the FPGA. For this we use the program FPGA option which links the “system.bit” file generated at the end of hardware implementation and the compiled “system.elf” file. The result is a “download.bit” file and this is downloaded into the FPGA using the JTAG downloading cable. After successful download, the output is viewed in HyperTerminal. Now we will be able to ping to the IP address 192.168.1.10 from the PC, the ping result is seen in the Fig. 2. Pinging is a test which confirms the link establishment between the board and the PC. After the link is established, the data transfer between the board and PC takes place i.e. the data is echoed back from the board to the PC using HW Tool which is shown in Fig. 3.



```
Pinging 192.168.1.10 with 32 bytes of data:
[illegible]
```

Fig. 2 PING result



```
Sending ICMP ECHO REQUEST to module
Received ICMP ECHO REPLY
Connecting to 192.168.1.10 ...
Connected to 192.168.1.10
LWIPSOCKETAPIECHOBACKLWIPSOCKETAPIECHOBACK  LWIP
SOCKET_API ECHOBACK  LWIP SOCKET_API ECHOBACK
LWIP SOCKET_API ECHOBACK  LWIP SOCKET_API ECHOBACK
```

Fig. 3 ECHOBACK result

B. Data transfer using the commands initialized from SCD

The SCD is nothing but a GUI (Graphical User Interface) developed using Visual C Basic in order to transmit the commands to the Target Board. The commands transmitted are health status, reset, set time, set threshold value etc. Since there is no hardware which gives the inputs, a SCD is used as a DRS (Device Replacement Software) which provides the inputs to the Target Hardware. Embedded application software is developed which is used to decode the information sent from the transmitter side, process it, understand the function to be performed and do the necessary at the receiver side. Apart from the commands sent, there are responses like Acknowledgement (or) Negative Acknowledgement which are sent from the Target Hardware to the SCD.

- **Health Status Command**

In this command the health of the board is known. If the link is established and the board and PC are connected the health is “OK” else “NOTOK”. The results are shown below –

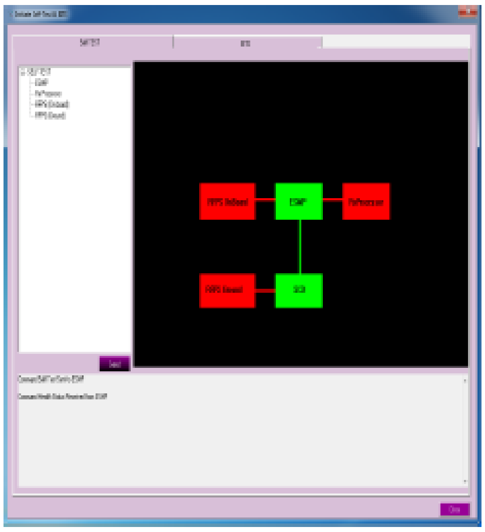


Fig. 4 HEALTH ‘OK’ result

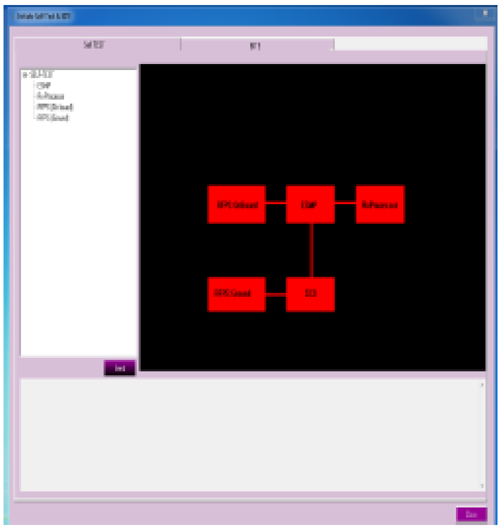
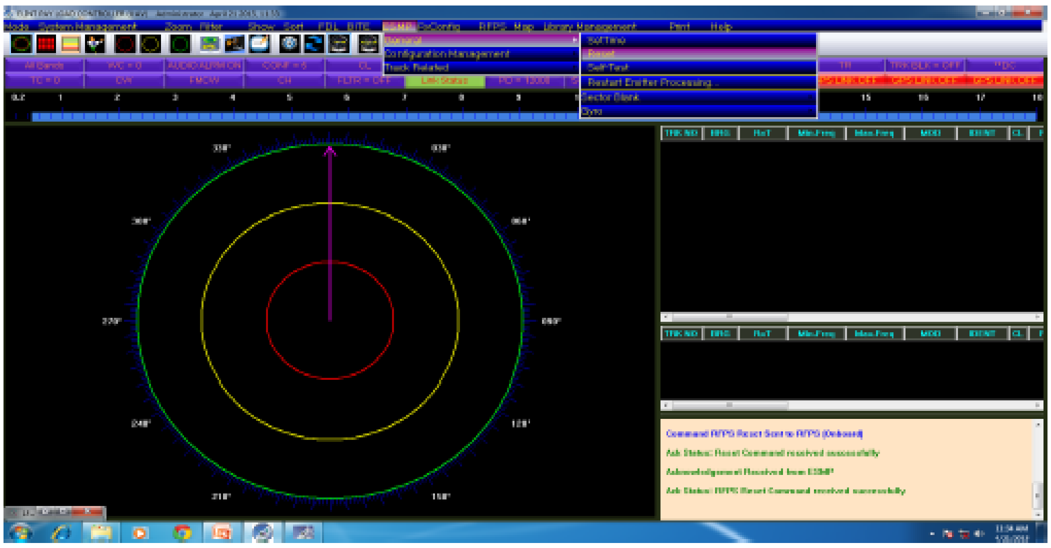


Fig. 5 HEALTH ‘NOTOK’ result

- **Reset Command**

This command is used to reset the Target Board so that the process can be repeated again for other applications. Acknowledgement is also sent from the board to the SCD because the protocol used is TCP/IP. The result is shown in Fig. 6.



Command RFPS Reset Sent to RFPS (Onboard)
Ack Status: Reset Command received successfully
Acknowledgement Received from ESMP
Ack Status: RFPS Reset Command received successfully

Fig. 6 RESET COMMAND result

IV. Conclusion
Ethernet communication has been successfully established between SCD and Development board. Subsequently, the commands have communicated between the SCD and Development board using TCP/IP protocol. All the commands have been successfully tested for its complete functionality and the results are observed on HyperTerminal, SCD and in HW Tool.

Acknowledgements
The authors wish to thank Director, DLRL for permitting to carry out this project. Authors also wish to thank Mr. S.R. Pankaj Kumar, Scientist-“F”, DLRL, Mr. K. Vikram, Scientist-“D”, DLRL, Mr. K. Kalyan Babu, Assistant Professor, Gitam University and Mr. G.V.K Sharma, Associate Professor, Gitam University, Dr. I. Srinivas Rao, Assistant Professor for their valuable guidance, help and insightful comments.

References

1. Andrew S. Tanenbaum, “Computer Networks”, Third Edition, Prentice Hall 1996.
2. Stallings, W:”Data and Computer Communications” 4th Edition, New York: Macmillan, 1994.
3. Ethernet: The Definitive Guide by Charles E. Spurgeon.
4. Transmission Control Protocol (RFC 793). Sep 1981. Information Sciences Institute, University of Southern California.
5. An Ethernet Address Resolution Protocol (RFC 826). Nov 1982. David C. Plummer, Network Working Group.
6. Ethernet – (http://www.geocities.com/SiliconValley/Haven/4824/ethernet.html).
7. Auto-Negotiation -(http://www.ethermanage.com/ethernet/autoneg.html). Charles Spurgeon, Bellereti.
8. IEEE Std 802.3-2002 Standard for Information Technology, IEEE Standards Association.
9. L80227 10BASE-T/100BASE-TX Ethernet PHY Technical Manual, Oct 2002. LSI Logic Corporation.

Reference Website:

1. www.xilinx.com