ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 5, May 2017

Data Transfer From Fpga To Pc Through Ethernet

[1] Sandhyarani N, [2] P C Srikanth

[1]PG Student, [2] Former Head of The HOD

Abstract - **This paper represents the implementation of embedded processor inside FPGA (Field programmable Gate Array) such that it can send data through Ethernet to Pc. The implementation platform is a development board which has a waxwing Spartan 6 FPGA. Using the XPS (Xilinx Platform Studio) tool, microblaze processor is specified. The software part of the processor is configured I SDK (software development kit).The implementation requires development board, Ethernet, power supply, high end PC. Using the application of echo I have written the c program. through the telnet software serial port results will be come.**

Index Terms—Cluster head, cooperative node, routing, Energy Efficient Uneven Clustering (EEUC), wireless sensor network (WSN).

I INTRODUCTION

A field-programmable gate array(FPGA)is an integrated circuit designed to be configured by a customer or a designer after manufacturing-hence "field-programmable", The FPGA configuration is generally specified using a hardware description language(HDL).FPGAs contain an array of programmable logic blocks ,and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together " –like many logic gates that can be inter-wired in different configurations. Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, logic blocks also include memory elements, which may be simple flip-flops or mo

With the advancement of FPGAs a new trend of implementing the microprocessors on the FPGAs has emerged in the design community. The ML505 board supports the embedded processor MicroBlaze re complete blocks of memory.

II. EXPERIMENTAL SETUP

The implementation platform is the waxwing FPGA Development board which has a Spartan 6 FPGA. The FPGA development board supports microblaze 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 the other components required by the design are a 100M Ethernet, AC97 codec and DVI-D.Waxwing development board features waxwing Xilinx Spartan 6 FPGA mini module with high density for external IO interface. this development board makes it easy to evaluate and build solutions with waxwing Spartan 6 FPGA module. the base system builder(BSB)wizard inside XPS tool is used for

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(IwIP). Now come to our application requirement, a C-program is written were in the server works on TCP data and listens for the input at the specified port COM4.in our design, the IP address198.168.1.10is connected to my FPGA board.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 5, May 2017

I. Implementation

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 Fig1,2,3. Pinging is a test which confirms the link establishment between the board and the PC using telnet software which is shown in figure.

Figure1:set the protocol and system

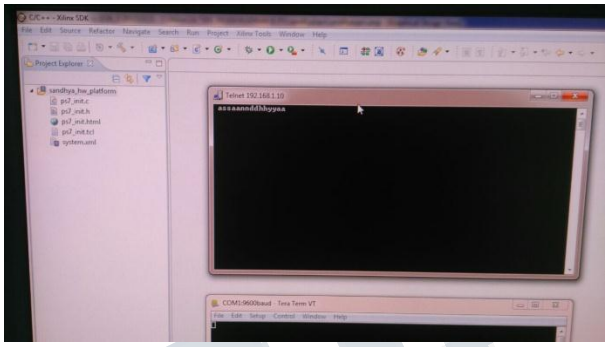

Figure2: Echoback result

II. Data transfer using the telnet from the NVT

- The NVT has a "printer" (or display) and a "keyboard".
- The keyboard produces outgoing data, which is sent over the TELNET connection. The printer receives the incoming data.
- The basic characteristics of an NVT, unless they are modified by mutually agreed options are:
  - The data representation is 7-bit ASCII transmitted in 8-bit bytes.
  - The NVT is a half-duplex device operating in a line-buffered mode.
  - The NVT provides a local echo function. all of these may be negotiated by two hosts

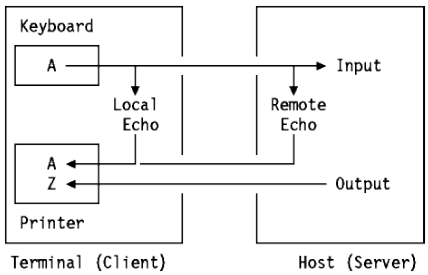

Figure3: Telnet communication model and NVT

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 5, May 2017

Telnet communication model and network virtual terminal

Telnet uses an approach similar to the analogy described above for dealing with its problem of hardware and software compatibility. Rather than having terminals and hosts communicate using their various native "languages", all Telnet clients and servers agree to send data and commands that adhere to a fictional, "virtual" terminal type call the Network Virtual Terminal (NVT). The NVT defines a set of rules for how information is formatted and sent, such as character set, line termination, and how information about the Telnet session itself is sent.

Each Telnet client running on a terminal understands both its native language and NVT. When information is entered by the user on his or her local terminal, it is converted to NVT for transmission over the network in NVT form. When the Telnet server receives this information, it translates it from NVT to the format that the remote host expects to receive it. The identical process is performed for transmissions from the server to the client, in reverse. This is illustrated in figure4.and the addressing setup shown in fig5 and microblaze processor core overview also shown in figure6

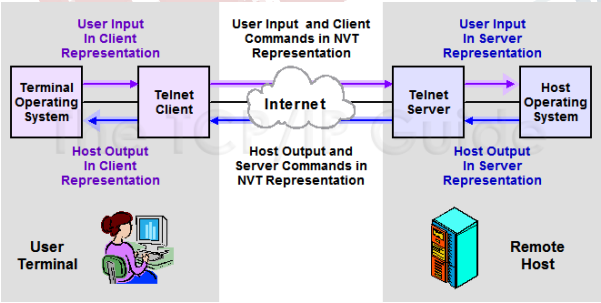

Figure4: Telnet communication and NVT

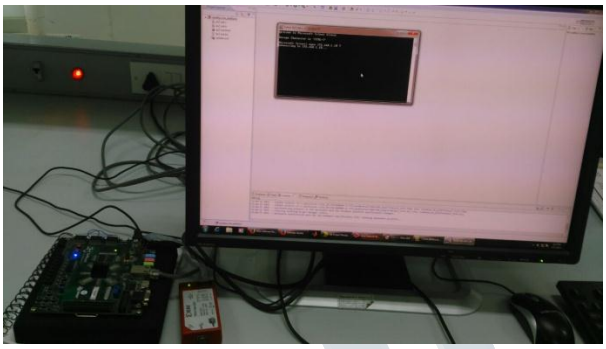

Figure5: addressing setup

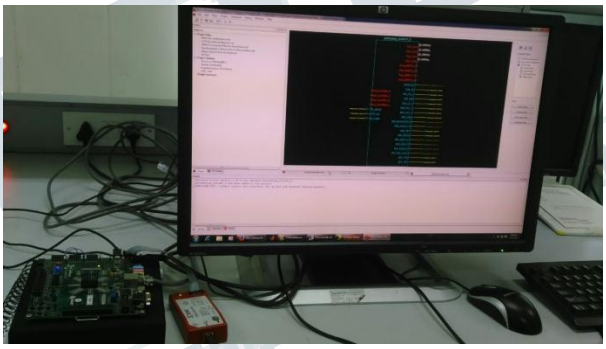

Figure6: Microblaze processor core connection view

CONCLUSION:

Ethernet communication has been established between telnet and development board Subsequently, the commands have communicated between the telnet and Development board using TCP/IP protocol. all the data have been transfer through ethernet using telnet. results are observed in the pc and hardware FPGA tool.

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.