International Journal of Engineering Research and General Science Volume 3, Issue 1, January-February, 2015
ISSN 2091-2730

FPGA BASED CONTROLLER CARD FOR DATA ACQUISITION SYSTEM

Augustine Abraham, Supriya Unnikrishnan K, Sudheesh Madhavan, Hema M

PG Scholar, Dept. of ECE, Toc H Institute of Science and Technology, augustinepanayil@hotmail.com

Abstract— Data acquisition is the process of measuring real world physical conditions and converting it into digital numeric values. A typical data acquisition system consists of (i) Sensors that convert physical parameters to electrical signals, (ii) Signal conditioning circuitry, (iii) Analog to Digital Convertors [1]. For Sound Navigation and Ranging (SONAR) front end application specific data acquisition systems, a controller card is required which controls the transmission of data that is received. In this paper, the focus is for an efficient controller card development by using the Virtex – 5 FPGA (Field Programmable Gate Array). FPGAs have soft IPs (Intellectual Property) and hard IPs that can be used on demand. Here the controller card is based on a soft processor microblaze. The use of soft processor reduces the complexity of developing the controller card with an FPGA and an external processor as a controller since the soft processor exists on the same FPGA. This creates a highly robust and reliable system.

Keywords— FPGA; Microblaze; Intellectual Property (IP); SONAR; Human Machine Interface (HMI); Xilinx Platform Studio (XPS); Software Development Kit (SDK)

INTRODUCTION

The data acquisition system receives the data from sensors and is converted to digital form by the Analog to Digital (ADC) cards. The data in digital form is fed from ADC cards to the controller card. Controller card has two regions, an embedded processor and a transceiver hardware section. Embedded processor communicates with the HMI through which necessary control data is taken. Depending on the instructions given by the HMI to the embedded processor, data received from ADC cards are routed to required destination. Virtex5LXT FPGAs have microblaze soft processors and peripherals like UART (Universal Asynchronous Receiver/Transmitter), Ethernet, DDR2 SDRAM (Double Data Rate Synchronous Dynamic Random-Access Memory), Flash etc. as soft IPs. Microblaze is a 32 bit Reduced Instruction Set Computing (RISC) processor [5]. The peripherals can be instantiated based on the requirement to create an embedded system inside the FPGA. Here, an embedded soft processor based controller is developed for SONAR front end application.

METHODOLOGY

ML505 evaluation board based on Virtex 5 FPGA of Xilinx is used for the system development. Block diagram of controller card is shown in Figure -1. The transceiver takes the configuration data from the block RAM (Random Access Memory) which is written by the soft processor. This paper focuses on development of microblaze soft processor based controller.

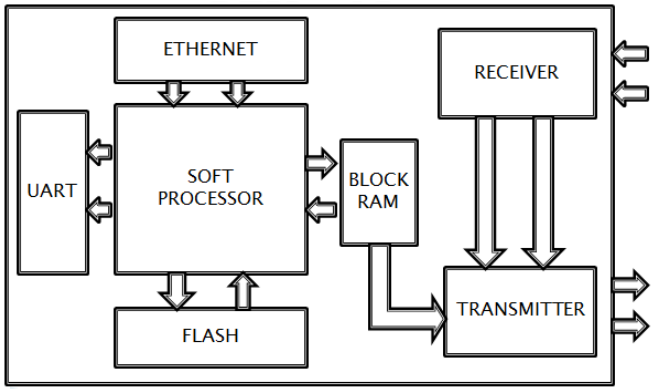

Figure – 1: Block diagram of controller card

Hardware Design

The embedded controller hardware is developed in XPS by instantiating the microblaze soft processor with peripherals UART, Ethernet, Flash and a custom memory peripheral generated from core generator. This custom memory peripheral is the bridge between the transceiver and the embedded processor.

The microblaze processor is instantiated with clock 125MHz and local memory of 64KB. UART and Ethernet forms the Human Machine Interface. The configuration details of the system and any debugging information are shown on the PC through the Tera Term/ Hyper Terminal using the UART 16550 through the RS232 cable. The Ethernet configured was a hard Ethernet MAC which could support speed up to 1Gps and is configured in Gigabit Media Independent Interface (GMII) mode with required jumper settings [6]. The custom Block RAM was made with 32 bit width and 1024 bit depth from the IP Core and the custom Block RAM (BRAM) was added by using the IP – IC (Intellectual Property – Inter-Connect) interface. It was done so because the BRAM must be common to hardware transceiver and the embedded controller. The control data was written to BRAM through the Ethernet and based upon the control data the transceiver would work.

The microblaze soft processor accesses its peripherals using Processor Local Bus [8]. Hence a Processor Local Bus top module is written over the custom memory in order to import the memory peripheral to microblaze processor using the IP – IC interface technology. Intel JS28F256P30T95 BPI flash is used to take back up of last configuration data in BRAM [4].

Software Design

The hardware design is exported on to the SDK to start with the embedded programming. Upon this hardware, a Board Support Package is made. The BSP was made such that the microblaze processor was loaded with Xilkernel OS [3]. Xilkernel is a small, robust, and modular kernel. It is highly integrated with the Platform Studio framework and is a free software library that is available with the Xilinx Embedded Development Kit (EDK) [9]. It allows a very high degree of customization, letting users tailor the kernel to an optimal level both in terms of size and functionality. To this BSP lwIP and xilflash are added for the Ethernet and flash support [11]. Since the design is based on Xilkernel OS, the lwIP is configured in socket mode [7]. The main advantage of Xilkernel OS is multithreading functionality.

Software design can be divided into three sections. The first section includes the startup procedure to initialize the UART along with a programming to assign specific IP address and MAC address to the board. The second section does the reception of packet and extracting the data to write into the BRAM. The third section does the initialization of flash and unlocking the specific memory areas to write the same data from BRAM to flash.

IMPLEMENTATION

The hardware implemented is shown in Figure – 2. The microblaze processor is instantiated with peripherals UART, Ethernet, DDR2, Flash and custom memory core. It could be seen that the microblaze processor communicates with the peripherals using the Processor Local Bus (PLB) and Local Memory Bus (LMB) is used to access the local memory Tools used is Xilinx 13.3 ISE/EDK, Lab VIEW, Wire shark and Tera Term.

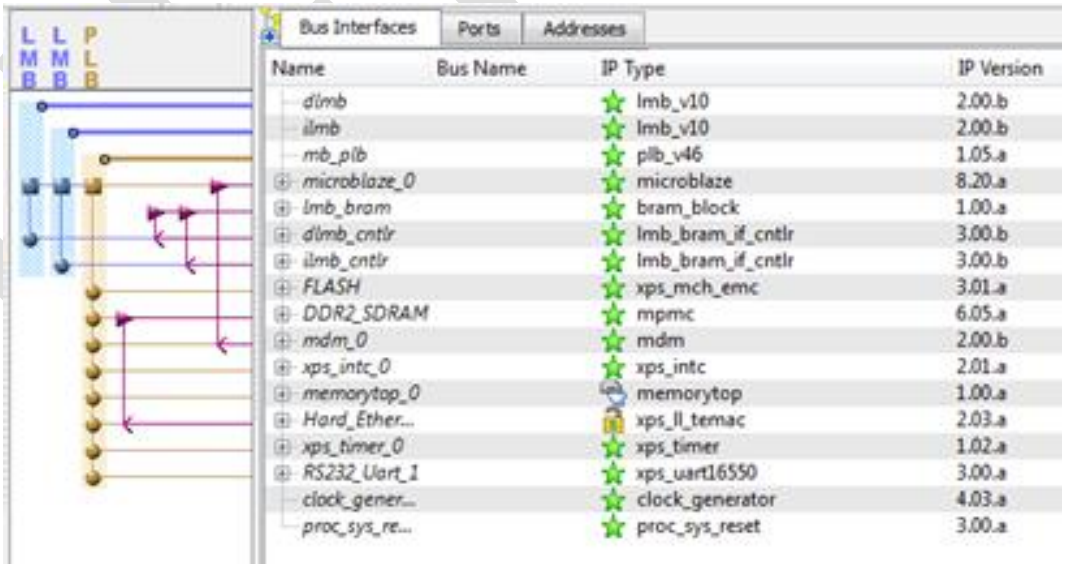

Figure – 2: XPS view of the hardware

Figure – 3 shows the addresses mapped to each of the microblaze peripherals. The memory added is assigned an address space of 1K and the starting address is 0x89840000. The Flash address starts at 0x8C000000 and has a size of 32M.

Bus Interfaces | Ports | Addresses

Instance	Base Name	Base Address	High Address	Size	Bus Interface(s)	Bus Name
microblaze_0's Address Map						
dlmb_cntlr	C_BASEADDR	0x00000000	0x0000FFFF	64K	SLMB	dlmb
ilmb_cntlr	C_BASEADDR	0x00000000	0x0000FFFF	64K	SLMB	ilmb
xps_intc_0	C_BASEADDR	0x81800000	0x8180FFFF	64K	SPLB	mb_plb
xps_timer_0	C_BASEADDR	0x83C00000	0x83C0FFFF	64K	SPLB	mb_plb
RS232_Uart_1	C_BASEADDR	0x83E00000	0x83E0FFFF	64K	SPLB	mb_plb
mdm_0	C_BASEADDR	0x84400000	0x8440FFFF	64K	SPLB	mb_plb
DDR2_SDRAM	C_SDMA_CTRL_BASEADDR	0x84600000	0x8460FFFF	64K	SDMA_CTRL1	mb_plb
Hard_Ethernet_MAC	C_BASEADDR	0x87000000	0x8707FFFF	512K	SPLB	mb_plb
memorytop_0	C_BASEADDR	0x89840000	0x898403FF	1K	SPLB	mb_plb
FLASH	C_MEM0_BASEADDR	0x8C000000	0x8DFFFFFF	32M	SPLB	mb_plb
DDR2_SDRAM	C_MPMC_BASEADDR	0x90000000	0x9FFFFFFF	256M	SPLB0:SDMA_LL1	mb_plb:Hard_Et...

Figure – 3: Address mapped to each peripherals

This implemented hardware is ported to SDK tool for writing the embedded code. Figure – 4 shows the SDK view of the system developed.

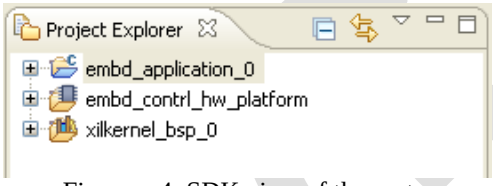

Figure – 4: SDK view of the system

RESULTS

An embedded soft processor controller was developed on ML505 board for SONAR front end application and the coding was done in embedded C [12]. The protocol supports provided include Internet Protocol (IP), Internet Control Message Protocol (ICMP), User Datagram Protocol (UDP), Transmission Control Protocol (TCP), and Internet Group Message Protocol (IGMP) [10]. The system was configured with IP address 192.168.1.100 [2]. The ping command was used to verify the ICMP protocol. TCP protocol was verified by checking the correctness of data by sending it back to PC. A UDP packet was send from lab view and that data was extracted and written on to the custom memory peripheral.

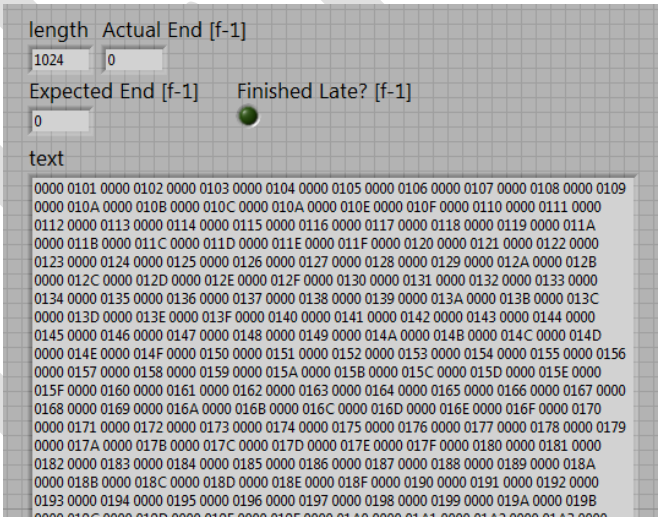

Figure – 5: Lab view providing inputs

The UART output in Tera Term shows the board configuration details initially. Tera Term screen shot is shown in Figure – 6. Once the UDP packet is received by the system, the received number of bytes and the data written to each memory location is displayed which validates the input data. After the completion of data writing to the custom memory, back up of same data is taken by writing the same to flash memory.

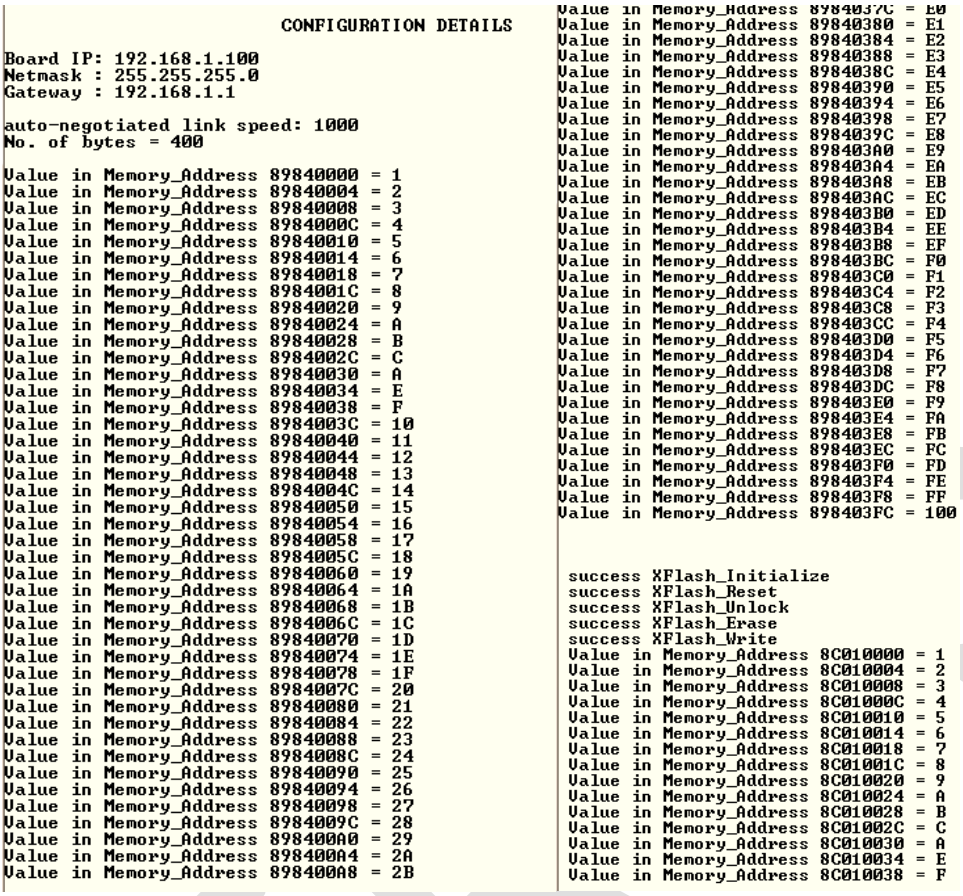

Figure – 6: Display in Tera Term

Wire shark was used as a monitoring tool which showed the data sent and the protocol followed by the data. It is shown in Figure – 7.

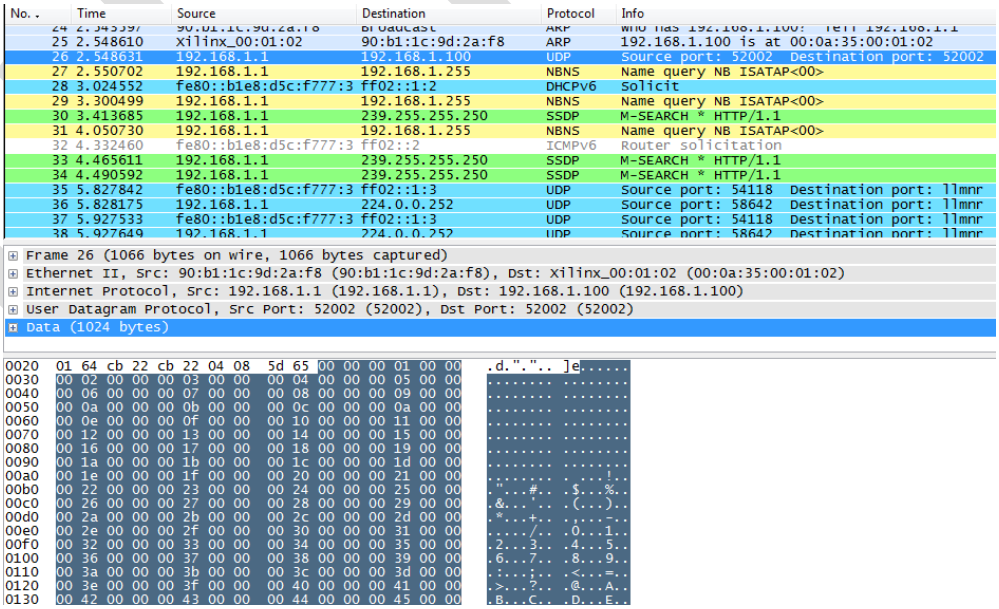

Figure – 7: Wire shark view

ACKNOWLEDGEMENT

The authors wish to thank Director, NPOL for permitting to carry out this project. Authors also wish to thank Mr. Suresh M., Scientist

G, NPOL, Mrs. Jayamma T. M., Scientist F, NPOL, Ralph D Kappithan and Scientist D, NPOL for their valuable guidance, help and insightful comments.

CONCLUSION

Soft processor based FPGA controller card is a promising reliable and robust system for SONAR front end applications. The hardware was built by instantiating microblaze soft processor with UART, Ethernet, Flash and custom memory as peripherals. On this hardware an embedded system was developed which receives data in UDP packet and extract the data to write to the custom memory and a backup of data was taken in flash simultaneously. The system was implemented on Xilinx ML505 board and the required functionalities were verified. Here a highly efficient system is suggested which reduces the handshake unreliability between the hardware and the processor that becomes a big bottle neck while designing controller cards.

REFERENCES:

[1] "Data Acquisition Fundamentals", National Instruments, August, 1999

[2] "Embedded System Example: Web Server Design Using MicroBlaze Soft Processor", Xilinx, October 13, 2006

[3] "Xilkernel", Xilinx, December 12, 2006

[4] "StrataFlash Embedded Memory (P30)", Numonyx, November, 2007

[5] "MicroBlaze Processor Reference Guide", Xilinx, 2008

[6] "ML505/506/507 Overview and Setup", Xilinx, June, 2008

[7] "LightWeight IP (lwIP) Application Examples. June 15", Xilinx, 2009

[8] "PLBV46 Slave Single (v1.01a)", Xilinx, June 22, 2010

[9] "Using EDK to Run Xilkernel on a MicroBlaze Processor", Xilinx, August 11, 2010

[10] "lwIP 1.3.0 Library (v3.01.a)", Xilinx, July 6, 2011

[11] "LibXil Flash (v3.01.a)", Xilinx, April 24, 2012

[12] "OS and Libraries Document Collection", Xilinx, April 24, 2012