IMPLEMENTATION OF RADAR CONTROLLER AND DISPLAY INTERFACE USING NICHESTACK AND NIOS II DEVELOPMENT SYSTEM

[1]MANJUNATH P S, [2]DEBASHIS JENA, [3]HEMANTH S J, [4]CHETAN R, [5]RAHUL LALMANI

[1,2,3,4]Dept. of Telecommunication Engineering, BMS College of Engineering, [5]Scientist, LRDE, DRDO, Bangalore
E-mail: manjunathps.tce@bmsce.ac.in

Abstract- The paper benevolences interfacing application program between Radar controller and Display in Nios II IDE. NIOS II IDE uses NicheStack TCP/IP stack which is to be implemented and to be analyzed for data rate and communication overhead which provides immediate access to a stack for Ethernet connectivity for the Nios II processor. Signal Processor Controller uses Nios II processor for sending and receiving data via Ethernet. The focus of the NicheStack TCP/IP Stack implementation is to reduce resource usage while providing a full-featured TCP/IP stack. Nios II is Altera's Second Generation Soft-Core 32 Bit RISC Microprocessorwhich provides flexibility to add features to enhance performance or conversely eliminate processor features and peripherals to fit the design in a smaller, low-cost device. The implementation of Nichestack is carried out using the software Nios II software build tools for eclipse and Quartus II provided by Altera. Successful transfer of real time radar data is demonstrated between the NIOS II acting as the radar controller (client) and a Host PC acting as display (server) and the data rate is analysed.

Keywords- NIOS II IDE, NicheStack TCP/IP, RISC, Quartus II

I. INTRODUCTION

NIOS IIis a 32-bit embedded-processor architecture designed specifically forAltera family of FPGAs which incorporates many enhancements over the original Nios architecture, making it more suitable for a wider range of embedded computing applications, from DSP to system-control. A Nios II processor system is equivalent to a microcontroller or "computer on a chip" that includes a processor and a combination of peripherals and memory on a single chip. Nios II processor systems use a consistent instruction set and programming model.
Getting started with the Nios II processor is similar to any other microcontroller family. The easiest way to start designing effectively is to purchase a development kit from Altera that includes a ready-made evaluation board and the Nios II Embedded Design Suite (EDS) containing all the software development tools necessary to write Nios II software.

The Nios II EDS includes two closely-related software development tool flows:

- The Nios II SBT
- The Nios II SBT for Eclipse

Nios II SBT provides a command line interface andNios II SBT for Eclipse provides a familiar and established environment for software development.
Using the Nios II hardware, reference designs can prototype an application running on a board before building a custom hardware platform. Figure 1shows an example of a Nios II processor reference design available in an Altera Nios II development kit.

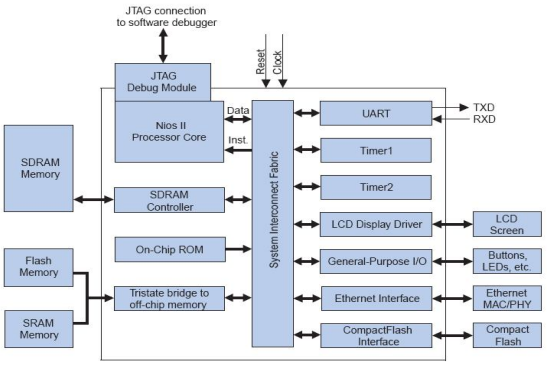

Fig. 1 Nios II Processor System

The processor peripherals can be classified into t Standard peripherals and other Custom peripherals.Standard peripherals are timers, serial communication interfaces, general purpose I/O and other memory interfaces. Custom peripherals are created to integrate with Nios II processor. This approach offers a double performance benefit, the hardware implementation is faster than software; and the processor is free to perform other functions in parallel while the custom peripheral operates on data. Nios II architecture supports a flat register file, consisting of thirty two 32-bit general-purpose integer registers, and up to thirty two 32-bit control registers. Arithmetic Logic Unit (ALU) operates on data stored in general-purpose registers. ALU operations take one or two inputs from registers, and store a result back in a register. The flexible nature of the Nios II memory and I/O organization are the most notable difference between Nios II processor systems and traditional microcontrollers. The Nios II architecture hides the hardware details from the programmer, so programmers can develop Nios II applications

without specific knowledge of the hardware implementation.

II. NIOS II - STARTIX II EDITION

The Stratix® II FPGA family is based on a 1.2-V, 90-nm, all-layer copper SRAM process and features a new logic structure that maximizes performance, and enables device densities approaching 180,000equivalent logic elements (LEs) offering up to 9 Mbits of on-chip, TriMatrix™ memory for demanding, memory intensive applications and has up to 96 DSP blocks with up to 384 (18-bit × 18-bit) multipliers for efficient implementation of high performance filters and other DSP functions. Stratix II devices are also the industry's first FPGAs with the ability to decrypt a configuration bit stream using the Advanced Encryption Standard (AES) algorithm to protect designs. The Nios development board, Stratix II Edition, provides a hardware platform for developing embedded systems based on Altera Stratix II devices.

Basic components of the NIOS II Startix board include Ethernet MAC/PHY (U4):Is a mixed signal analog/digital device that implements protocols at 10 Mbps and 100 Mbps.

Individual LEDs (D0 - D7): connected to general purpose I/O pins on the Stratix II device. When the Stratix II pin drives logic 1, corresponding LED turns on. Seven-Segment LEDs (U8 & U9): U8 and U9 are connected to the Stratix II device so that each segment is individually controlled by a general-purpose I/O pin. When the Stratix II pin drives logic 0, the corresponding LED turns on.

SRAM Memory (U35 & U36): Are IDT IDT71V416S, 512 Kbyte x 16-bit asynchronous SRAM devices used as general-purpose memory. The two 16-bit devices can be used in parallel to implement a 32-bit wide memory subsystem. The SRAM devices share address and data connections with the flash memory and the Ethernet MAC/PHY device.
Flash Memory (U5):is a 16 Mbyte AMD AM29LV128M flash memory device connected to the Stratix II device.

Configuration & Reset Buttons: The Nios development board uses dedicated switches SW8 for reset for CPU, SW9 re configurethe Stratix II device with the factory configuration and SW10 for the power-on reset button.

JTAG Connectors (J24 & J5): Each JTAG header connects to one Altera device and forms a single-device JTAG chain. J24 connects to the Stratix II device (U60), and J5 connects to the EPM7128AE device (U3).

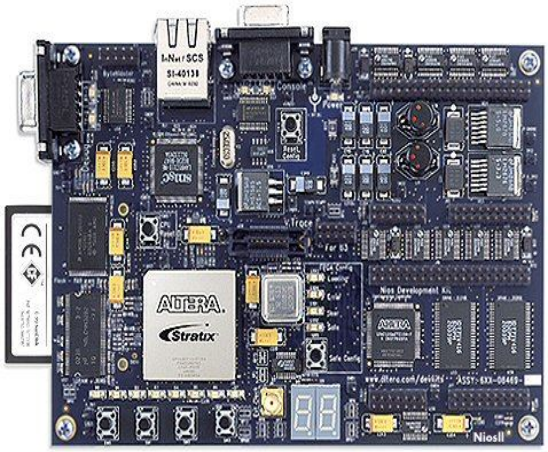

Fig. 2 Niosdevelopment board

III. MICROC/OS-II REAL-TIME OS

MicroC/OS-II is a popular real-time kernel produced by MicriumIncwhich is a portable, ROMable, scalable, pre-emptive, real-time, multitasking kernel which provides services like Tasks (threads), Event flags,Message passing, Memory& Time management, Semaphores.

MicroC/OS-II development for the Nios II processor allows programs to port to other Nios II hardware systems.

3.1 Software Architecture

The onion diagram in Figure 3 shows the architectural layers of a Nios II MicroC/OS-II software application.

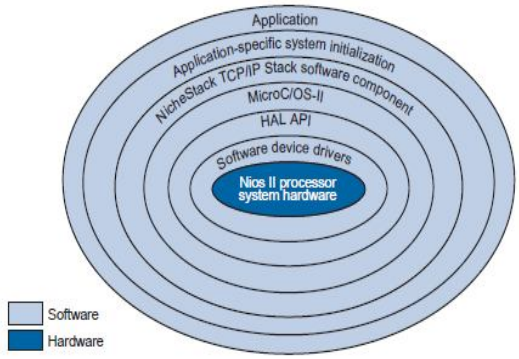

Fig. 3 Layered Software Model

Each layer encapsulates the specific implementation details of that layer, abstracting the data for the next outer layer. The list describes each layer:

Nios II processor system hardware - The core of the onion diagram represents the Nios II soft-core processor and hardware peripherals implemented in the FPGA.

Software device drivers - Contains the software functions that manipulate the Ethernet and hardware peripherals. These drivers know the physical details

of the peripheral devices, abstracting those details from the outer layers.

HAL API - The Altera Hardware Abstraction Layer (HAL) application programming interface (API) provides a standardized interface to the software device drivers, presenting a POSIX-like API to the outer layers.

MicroC/OS-II - The MicroC/OS-II RTOS layer provides multitasking and intertask communication services to the NicheStackTCP/IP Networking Stack and the Nios II Simple Socket Server.

NicheStack TCP/IP Stack software component - provides networking services to the application layer and application-specific system initialization layer via the sockets API. Application-specific system initialization - The application-specific system initialization layer includes the MicroC/OS-II and NicheStack TCP/IP Stack software component initialization functions invoked from main(), as well as creates all application tasks, and all of the semaphores, queue, and event flag RTOS inter-task communication resources.

Application- The outermost application layer contains the Nios II Simple Socket Server task and LED management tasks. Socket: Sockets provide an interface for programming networks at the transport layer. It is an interface between an application process and transport layer. The application process can send/receive messages to/from another application process (local or remote) via a socket.

3.2 Client/Server Communication

Generally, programs running on client machines make requests to a program (often called as server program) running on a server machine. The most widely used programming interfaces for TCP/IP protocols are sockets.Servers and clients have different behaviors therefore the process of creating them is different. What follows is the general model for creating a streaming TCP/IP server and client. The steps in creating (and difference between) server and client sockets is shown in figure 4.

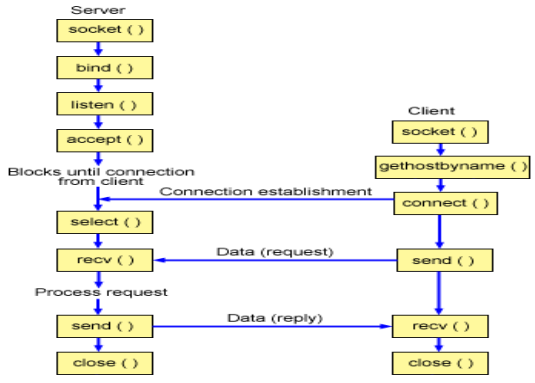

Fig 4. Flow in Client Server Model

IV. NICHESTACKTCP/IP STACK

The NicheStack® TCP/IP Stack - Nios® II Edition is a small-footprint implementation of the TCP/IP suite which is to reduce resource usage while providing a full-featured TCP/IP stack. Altera provides the NicheStack TCP/IP Stack as a software package that is added to board support package (BSP), available through the Nios II Software Build Tools (SBT).

The NicheStack TCP/IP Stack includes many features like Internet Protocol (IP) including packet forwarding over multiple network interfacesInternet control message protocol (ICMP) for network maintenance and debugging User datagram protocol (UDP)Transmission Control Protocol (TCP) with congestion control, round trip time(RTT) estimation, and fast recovery and retransmitDynamic host configuration protocol (DHCP)Address resolution protocol (ARP) for EthernetStandard sockets application program interface (API) The NicheStack TCP/IP Stack provides immediate access to a stack for Ethernet connectivity for the Nios II processor and uses the MicroC/OS-II RTOS multithreaded environment.

V. NIOS II SOFTWARE DEVELOPMENT

Before we can start building a software project in NIOS II SBT for Eclipse we create the hardware for the system using the Quartus II and SOPC Builder software.

The main output produced by generating the hardware for the system is the SRAM Object File (.sof), which is the hardware image of the system, and the SOPC Information File (.sopcinfo), which describes the hardware components and connections. This software generation tools use the (.sopcinfo) file to create a BSP project.

The BSP project is a collection of C source, header and initialization files, and a makefile for building a custom library for the hardware in the system. The NIOS IIsoftware development flow is shown in figure 5.

6.2 The NIOS II Software Build Tool (SBT) for Eclipse: Is an easy-to-use GUI (Graphical User Interface) that automates, build and make file management and integrates Nios II flash programmer and the Quartus II Programmer.

We use the Nios II SBT for Eclipse to compile a simple C language program for Client to run on the Nios II standard system configured in the FPGA on your development board. We create a new software project, build it, and run it on the target hardware and also edit the project, re-build it, and set up a debugging session.

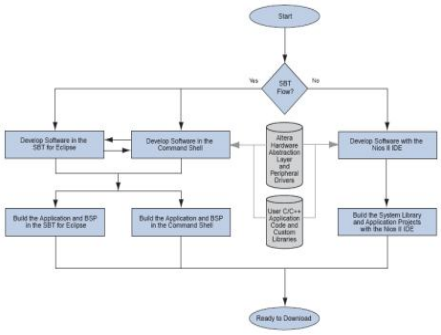

Fig. 5 NIOS II Software Development Flow

VI. IMPLEMENTATION

The implementation of the project is considered as a block diagram as shown in figure 6.

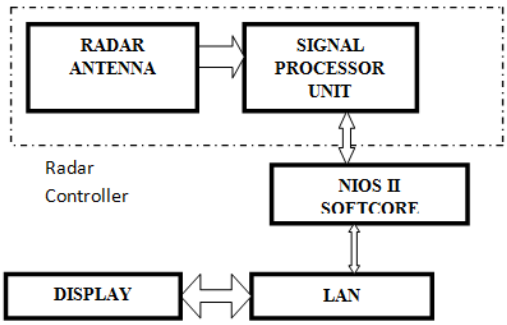

Fig. 6 Project block diagram

6.1 Nios II SBT for Eclipse Flow

Creating Project *Client*

Open the application software NIOS II SBT for Eclipse from the installed directory.
On the blank SBT window click File > New >Nios II Application and BSP from Template.
The Nios II Application and BSP from Template wizard appears. For SOPC Information File name, browse to <project directory>and open the SOPC Information File (.sopcinfo) for the design.

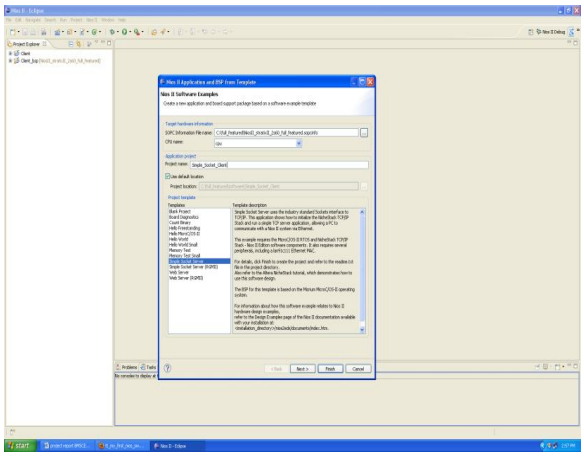

Fig. 7. Creating Project Client

In the Project name box, type Client.
Click Finish. The Nios II SBT for Eclipse creates the project and returns to the Nios II perspective.

After creating a new project, the Nios II SBT for Eclipse creates the following new projects in the Project Explorer view:

Client is the application project. Client_bspis the board support package (BSP) for our Nios II system hardware. The BSP includes following details:

Component device drivers for the Nios II hardware system, Nios II hardware abstraction layer (HAL), NicheStack TCP/IP Network Stack, Nios II Edition, Nios II host file system, Nios II read-only zip files system, Micrium's MicroC/OS-II real-time operating system (RTOS), system.h which is a header file that encapsulates your hardware system alt_sys_init.c which is an initialization file that initializes the devices in the system, linker.h which is a header file that contains information about the linker memory layout.

6.2 Building and Running the Nios II Simple Socket Client Project

In this section we build the application, configure the development board with a hardware design, and download the executable software file to the FPGA on the board. To build and run the application, perform the following steps:

Configure the FPGA on the development board by performing the following steps:

On the Nios II menu, click Quartus II Programmer.
In the Quartus II Programmer dialog box, Click on Add File and Browse .**sof** file. Information for the file appears in the Quartus II Programmer dialog box. Verify Program/Configure is on. Click **Start** to configure the FPGA on the development board.

On the File menu, click **Exit** to close the Quartus II Programmer and return to the Nios II SBT for Eclipse. If you receive a message that asks if you want to save the changes to the **chain1.cdf** file, click **No.**

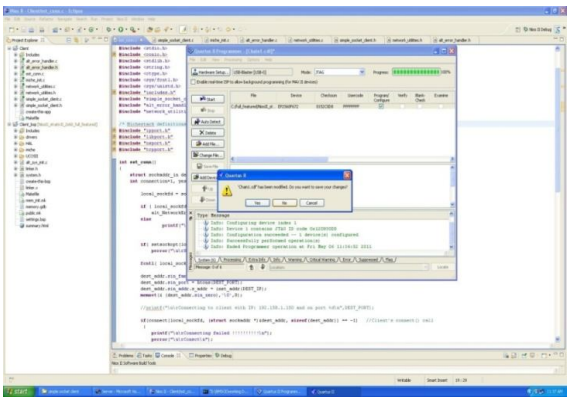

Fig. 8 Verify Program/Configure is on

To build the program, right-click the *Client* project in the Project Explorer view, and click Build Project.

The Build Project dialog box appears and the Nios II SBT for Eclipse begins compiling the project. When compilation completes, the message "Build completed" appears in the Console view. The completion time varies depending on your system.

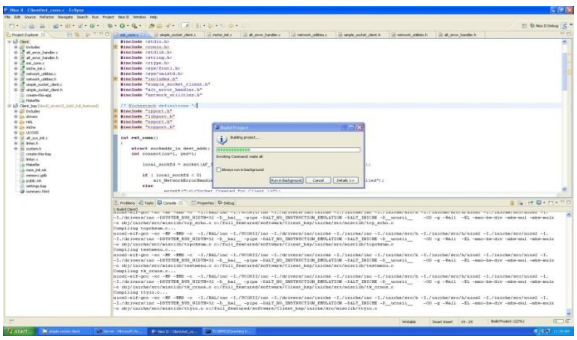

Fig. 9 To build a program and complete compilation

To run the program, right-click *Client*, point to Run As, and click Nios II Hardware. The Nios II SBT for Eclipse downloads the program to the FPGA on the target board and executes the code.

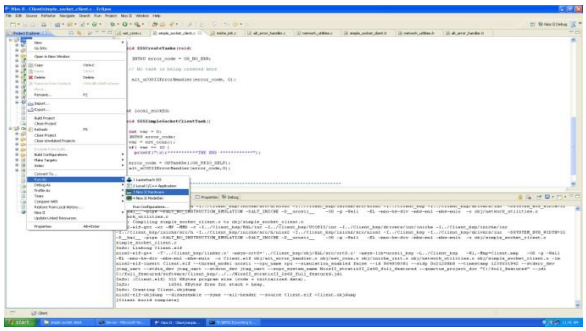

Fig. 10. To run the program and execute the code

The work is carried out to implement interfacing application program between Radar controller and Display in Nios II IDE that uses NicheStack TCP/IP. The NicheStack TCP/IP Stack provides immediate access to a stack for Ethernet connectivity for the Nios II processor. Signal Processor Controller uses Nios II processor for sending and receiving data via Ethernet. NicheStack is used for Ethernet connectivity. We have successfully demonstrated the transfer of data from a client which is run on the NIOS II processor and a server run on Microsoft Visual C++ on a host PC. Figure 11 shows the basic working of our project.

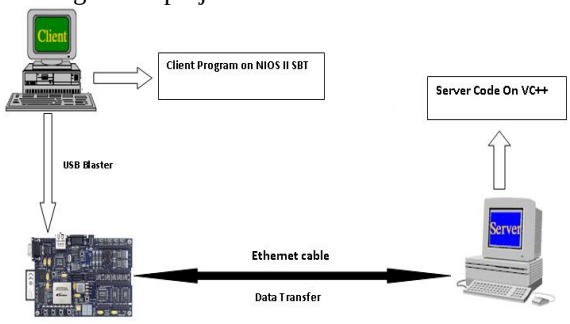

Fig. 11. Working Model

VII. RESULTS

After running both the client and server programs data could be successfully transferred. Figure 12 and Figure 13 below shows the output windows on compilation and running of client and server programs.

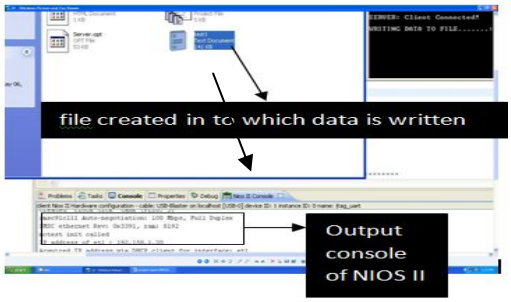

Fig. 12. Output window on compilation

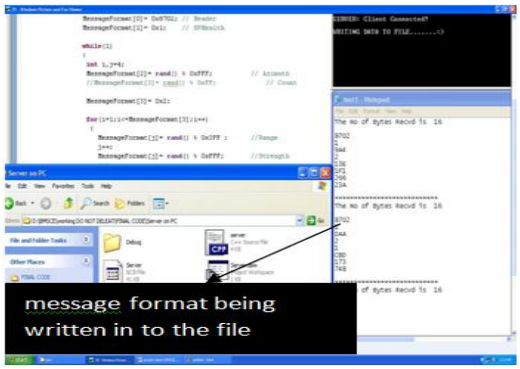

Fig. 13. Output window running of client and server programs.

CONCLUSION AND FUTURE SCOPE

The interface developed provides full duplex communication at high data rate between client and server. There is seamless communication between client and server. The processor used provides flexibility to its users. The protocol used provides greater advantages to embedded systems with minimal resources. The use of Nichestack reduces resource usage of the embedded system. The protocol facilitates embedded systems to use the Ethernet. The interface developed is used presently in electronic war-fare systems and in other general applications of embedded systems. Mainly it is implemented in sending and receiving data to and from radar controller to the display. Many challenges have been investigated in the past few years to make network communications more reliable, scalable, standardized and higher performance. As it is new concept there is a lot of scope in future to enhance network performance by using Nichestack TCP/IP protocol.

REFERENCES

[1] Altera's Nios II Processor Reference Handbook.

[2] Altera's Nios II software Developer's Handbook.

[3] MicroC/OS-II, The Real-Time Kernel textbook by Jean J. Labrosse.

[4] Beej's Guide to Network Programming. Let Us C by YashwantKanetkar.

[5] www.altera.com

[6] www.tenouk.com/Winsock

[7] Key Engineering Materials ;Vol 464,20Jan2011;PP: 237-240Xiao Fei Wang; Bo Quan Li;

[8] Hai Bin PanDesign of Signal Processing System for Doppler Speed Radar Based on Sopc.

★★★