A Java-Based Embedded Digital Signal Processing System

ZHE LOU, STEFAAN VERSCHUERE, JO PLENTICKX, XING-ZHI QIU, JAN VANDEWEGE
INTEC_design, Information and Technology
Ghent University
St. Pietersnieuwstraat 41, B-9000, Gent
BELGIUM

Abstract: - This paper presents a java-based embedded digital signal processing system, which consists of DSP processors, FPGA chips and friendly graphical user interface made of java. The DSP processors are very powerful at calculation and control functions and the FPGA chips are responsible for fast data exchange and some real-time operations. The use of the java programming language makes this system platform independent and easy to use. The system can be accessed via Internet. Only a web-browser is needed to control and monitor the system on a computer. A current monitoring application is depicted in the paper and the experiment results show the java-based embedded digital signal processing system is a very flexible, portable and easy-to-use digital signal processing platform.

Key-Words: - Digital Signal Processing, Embedded System, Real-Time Operation, FPGA, DSP.

1 Introduction

Since the first Digital Signal Processor (DSP) appeared in 1980's, DSP chips have been more and more widely used in the electronic engineering field because of its special architecture for digital signal processing. DSP chips are capable of carrying out millions of floating point operations per second, which are not only good at digital signal processing, but also very powerful in the industrial control systems[1]. However, DSP chips don't have the ability of reading/writing high speed I/O, e.g. giga bit rate input/output, which hinders its high speed application. The introduction of the Field Programmable Gate Array (FPGA) is being the best solution to this problem[2]. The I/O performance of some advanced FPGAs is increased to 3Gbps by using source synchronous data transmission architecture. Since both DSP and FPGA are programmable, a combination of these two chips results in a very general and powerful digital signal processing system, which not only has very good calculation and control functionality but also has the ability to handle high-speed digital signals.

Current advances in networking technology and software engineering techniques make it possible to build open, interoperable, modular and extensible digital signal processing systems which, for example, can be dedicated to monitoring and control requirements of a physical environment through the Internet[3]. This paper presents an architecture of a Java-Based Embedded Digital Signal Processing System (JEDSPS). The JEDSPS becomes a very powerful digital signal processing system by incorporating both DSPs and FPGAs. And it is a platform independent embedded system because the Java language is used as the program language for the Graphical User Interface (GUI). The reason why java is selected as the programming language for the GUI is its platform independence and tight security. The Java language from SUN, is becoming more and more popular because of its powerful web-programming[4]. It also enforces tight security--a java program cannot access anything to which the client computer does not specifically give it access. The user can control the JEDSPS via a computer in the Internet and the GUI will be automatically downloaded to the computer so that nothing but a web-browser is needed in the computer.

The rest of the paper is organized as follows: section 2 depicts the system architecture of the JEDSPS and section 3 describes its software architecture. A JEDSPS application is introduced in section 4. Finally the conclusion is given.

2 The System Architecture

The system structure of the JEDSPS is given in Fig.1, from which one can see that the JEDSPS is comprised of 4 parts, the FPGA circuitry, the DSP block, the data storage part and the Ethernet control part. The FPGA block is directly connected to the instruments/devices to transmit/receive data. Some real-time operations, e.g. encoding/decoding, can be done within the FPGA chip before the data is transferred to the DSP block or temporarily saved in the memory if the data transmission speed is beyond the DSP's capability. The non-time critical operations are performed in the DSP block and the data will be sent to the GUI via the Ethernet control block afterwards.

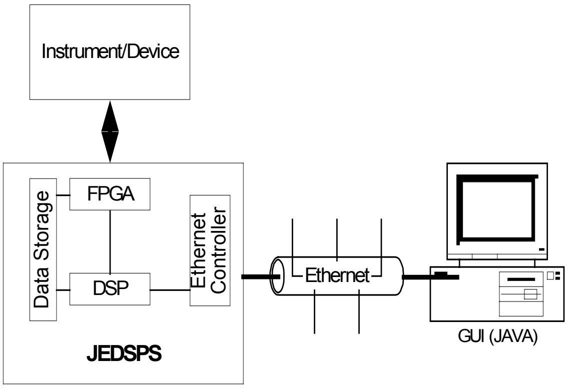


Fig.1. The system architecture of the JEDSPS.

2.1 The DSP Block

The DSP block is one of the key parts of the JEDSPS, which interacts with the FPGA for data communication, with the memory for data storage and with the Ethernet controller to establish the physical layer of the network. The DSP chip performs most of non-time critical digital signal processing, data calculations and controls the whole system. The interface between the DSP processor and the FPGA chips is mainly for two purposes, configuration and communication. At power-up, the FPGA chips need to be programmed to implements their functionalities. And during operation, the DSP is capable of reprogramming FPGA chips individually without changing the other FPGA chips' functionalities. The DSP chip offers the possibility to master other standard logical interfaces, e.g. i2c and uart, which can be used to control some components or monitor the system environment. The DSP has the ability to change its own firmware that is saved in the non-volatile memory when it is working and the new core will take effect on next boot up.

2.2 The FPGA Circuitry

The FPGA circuitry on the JEDSPS is mainly used for high-speed data communication and real-time operations. The I/O performance of some advanced FPGAs is increased to around 3Gbps by using source synchronous data transmission architecture. It supports many I/O standards, including TTL, LVDS, LVPECL and CMOS. The FPGA mainly has 3 functionalities, data transmission, data reception and real-time operation. It transmits/receives data to/from instruments, memories and the DSP according to various logic interfaces. Several internal FIFO structures are made to temporarily save data before sending out. Some real-time operations, which cannot be done by the DSP chip, will be implemented here. Since the FPGA chip is a programmable logic device, it is very flexible to change its functionalities to offer different device-to-device interfaces, I/O standards, timing and control options.

3 The Software Structure

The proposed software architecture which is the core of the JEDSPS, is shown in Fig.2. This architecture is modular in order to isolate and make independent the main parts of the system. It is possible to configure four modules: the FPGA communication manager, the memory manager, the task manager, the Internet communication manager and the GUI.

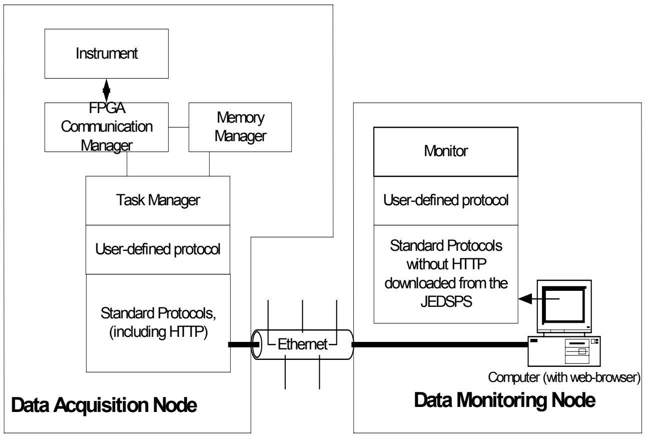

Instrument
FPGA Communication Manager
Memory Manager
Task Manager
User-defined protocol
Standard Protocols, (including HTTP)
Data Acquisition Node
Ethernet
Monitor
User-defined protocol
Standard Protocols without HTTP downloaded from the JEDSPS
Computer (with web-browser)
Data Monitoring Node

Fig.2. The software architecture of the JEDSPS

The FPGA communication manager executes a data acquisition loop, which interacts with i) the instruments to transmit/receive data and commands, ii) the memory manager to read or save data information, iii) the task manager to receive user's commands or send data back to data monitoring node. The data acquisition loop running on the DSP processor is mainly used to control the FPGA chips to transmit or receive data from the instruments. It is also responsible for programming the FPGA chips, which makes the FPGA implement different functionalities in different environments.

The memory manager is responsible for saving data. According to the DSP's commands, it saves the data either into the volatile memory or into non-volatile memory. Some non-volatile memories are reserved to store the firmware of the JEDSPS.

The task manager is a bit similar with the "task manager" in the windows operating system, which is responsible for generating, assigning and executing tasks. Several task queues are implemented with different priorities. After the DSP receives a command, it will generate a few tasks with varied priorities and every task will be put into a queue according to its priority.

The Internet communication manager is designed for establishing communication between data acquisition node and data monitoring node. It includes implementing standard Internet protocols, Internet Protocol (IP), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), HyperText Transfer Protocol (HTTP), Internet Control Message Protocol (ICMP) and Address Resolution Protocol (ARP) and user defined protocols designed for specific applications. The main problem with remote control requests through the network is to ensure that data and commands are transmitted within deterministic delay bounds. In the software architecture of the embedded control platform, attention has been focused on the Internet communication manager, which is responsible for encapsulating the protocols between two nodes. Since all the standard Internet protocols are fixed, the design of the user-defined protocol is the key to guarantee the efficiency and reliability of the transmission. The design of the user-defined protocol should consider the protocol that is used in the transport layer. If the UDP is used in transport layer, consideration of designing the user-defined protocol should be put on how to make the transmission more reliable. If the TCP is applied, then how to lighten the rest of the payload will be the most important thing.

The GUI consists of 2 parts, a simple web page and the java applet. The web page is the carrier of the java applet that is the core of the GUI. Instead of being put in a computer, the GUI is saved in the non-volatile memories of the JEDSPS. The JEDSPS has its own IP address, which makes it act as a “server” when connected to the Internet. Because the HTTP is implemented, only a web-browser is needed on the computer on the data-monitoring node. During the communication setup, a web-page is sent to the computer based on the HTTP. The computer with the Java virtual machine on it will automatically start to download the java applet, the GUI, after finishing downloading the web-page. During the communication, the HTTP is not in use any more because a UDP socket or a TCP socket will be used to directly establish the communication link between the computer and the JEDSPS.

4 An Example of JEDSPS

A setup of the JEDSPS was built in the INTEC_design lab to monitor the current of a printed circuitry board. This setup is mainly made of 2 DSP processors, ADSP-21065L from the Analog Devices, 2 FPGA chips, XCV200E and XC2S200 from Xilinx, and 1 Ethernet controller, CS8900A from Cirrus Logic and some sdram and flash components, as shown in Fig.3.

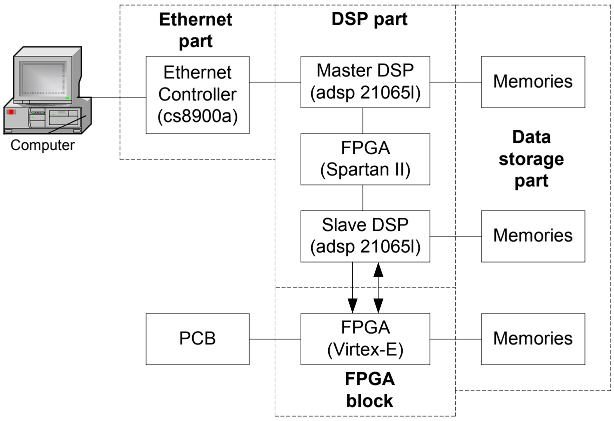
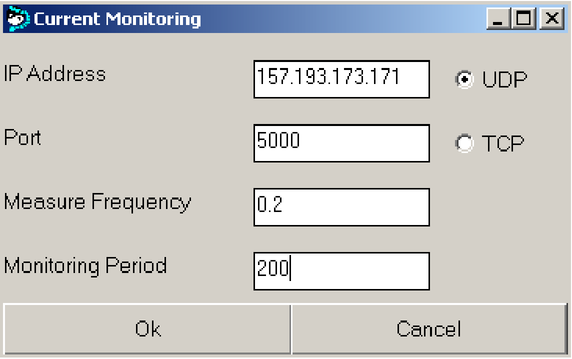


Fig.3. An application example of JEDSPS

One of the DSP processor, the master DSP, is responsible for managing and controlling the whole system. Except the FPGA communication module, the other three software modules are running on the master DSP. The other DSP, the slave DSP, is mainly responsible for program/re-program and communication with the FPGA chip (Virtex-E). A small FPGA, Spartan II, in-between acts as a FIFO and offers link between the DSP and other logical components. The Virtex-E is presented to directly connect to the printed circuitry board to get the current information via SPI interface.



IP Address 157.193.173.171 ● UDP

Port 5000 ○ TCP

Measure Frequency 0.2

Monitoring Period 200

Fig.4. The graphical user interface of current monitoring

A simple GUI is shown in Fig.4. From this GUI, one can see that the JEDSPS has an IP address and either the communication protocol or the port is configurable. After giving the measure frequency and the monitoring period, this information will be sent to the JEDSPS and a popup message box will show whether the transmission succeeds. Based on the measure frequency and monitoring period, the DSP will generate several tasks to meet the requirement. One of the monitoring results is shown in Fig.5.

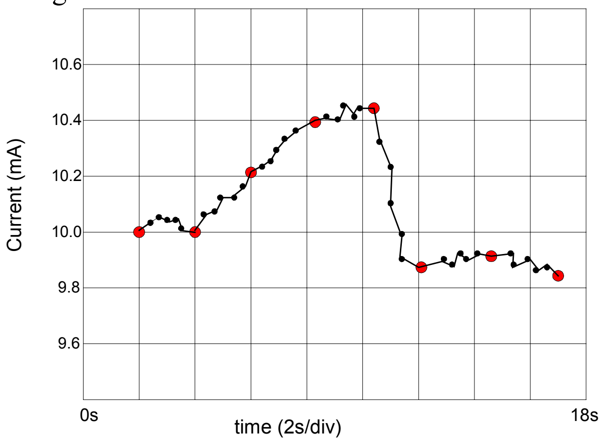


Fig.5. The monitoring result

5 Conclusion

In this paper, a java-based general embedded digital signal processing system is presented. It is a very flexible and powerful system due to its programmable core. Its built-in Internet connection makes it possible to be controlled via network and the customer only require a web browser to establish the link to the embedded system because of the usage of the object-oriented programming language, Java. The experiment results show that it is suitable for remote data controlling, monitoring and measurement.

References:

[1] R. H. Hosking, High Speed Storage Data Links for Data Acquisition and DSP, ICSPAT97.

[2] Peter Schulz, Embedding DSPs into FPGA-based Structures of Interconnection. ICSPAT96.

[3] B.R. Kumar, K. Sridharan and K. Srinivasan, The Design and Development of a Web-Based Data Acquisition System. IEEE trans. on instrumentation and measurement, vol 51, NO. 3, Jun. 2002

[4] D. Frimaldi, L.Nigro and F. Pupo, Java-Based Distributed Measurement Systems. IEEE trans. on instrumentation and measurement, vol 47 No. 1, Feb. 1998