APPLICATIONS AND UPGRADE OF FLEXIBLE AND LOGIC-RECONFIGURABLE VME BOARD

T. Hirono[#], T. Ohata, T. Kudo, JASRI/SPring-8, Sayo, Hyogo 679-5198, Japan

Abstract

A flexible logic-reconfigurable VME board [1] was designed to be used in a wide range of applications, for which fast real-time control must be achieved. The abovementioned VME board can be used in tag generation systems and as controller in pulse selectors. The board has field-programmable gate array (FPGA) chips, and system sequences are run on one of the FPGA chips. We have created a novel environment for developing the FPGA firmware. We adopt C as the implementation language instead of a hardware description language. With this novel environment, the development time is considerably reduced. We also upgrade the board so that it can be used in more types of applications. The upgraded board has a PCI bus and a gigabit Ethernet port for data transfer. The new board can be used in control systems in which mass data acquisition and fast real-time control are required.

INTRODUCTION

The flexible logic-reconfigurable VME board [1] has two components: a base board with FPGA chips and simple I/O mezzanine cards. Development of the FPGA firmware and customization of the I/Os are necessary for using the board in control systems. The architecture of the VME board allows customization of the I/Os to be carried out within a short time. Hence, users only need to mount the appropriate I/O mezzanine cards on the base board using connectors and screws. Further, users can easily develop a new, simple mezzanine card if the specifications of the existing I/O mezzanine card are not suitable for their system. The time required for hardware development can be significantly reduced when using the flexible logic-reconfigurable VME board.

One of the FPGA chips on the base board can be reconfigured by users and used for implementing system sequences. However, users who are not familiar with FPGA take a long time for firmware development. This limits the applicability of the developed board. In this study, we have designed a new development environment for increasing the applicability of our board.

A VME bus is used as the external bus for this board. The low data transmission rate of this bus (40 MB/s) prevents the board from being used in data acquisition systems. Hence, we upgrade the base board so that it can be equipped with interfaces for fast data transfer in this sutdy.

REQUIREMENTS FOR DEVELOPMENT ENVIRONMENT

The main purpose for creating a new development environment is to reduce the time taken for developing the firmware for the FPGA. The following are the three requirements for the novel development environment: (1) an implementation language familiar to users, (2) ability to support several types of I/O mezzanine cards, and (3) separation of codes that are unrelated to the user algorithm.

Implementation Language

For reducing the development time, it is necessary to use a well-known implementation language that most users are familiar with. The implementation languages used in the development environments provided by FPGA vendors are hardware description languages such as VHDL and Verilog. However, we do not adopt these languages because most users are not familiar with them. We adopt C as the implementation language because many of the major control systems in SPring-8 [2] are developed using this language, and we have many C experts. Besides, C can be easily learnt from several books and websites.

Ability to Support I/O Mezzanine Cards

Currently, there are four types of I/O mezzanine cards available, and several more will be developed in the future. Our development environment is required to support these I/O mezzanine cards. We adopt a commercial tool ImpulseC [3] as the C-to-VHDL converter. Impulse C generates a register transfer level (RTL) module with standardized data interfaces. With the help of Tcl scripts in InpulseC, these interfaces can be customized to suit specific hardware or the existing VHDL RTL modules. By appropriate modification of the Tcl scripts, our development environment can be made to support I/O mezzanine cards. ImpulseC is one of the few tools that can support custom-made hardware such as our VME board.

Separation of Codes

The major difficulty in the development of FPGA firmware is coding of data transfer sequences between devices, such as writting data from the FPGA to the double data rate (DDR) memory. Howerever, data transfer sequences are not directly related to the user argorithm. We have developed data transfer sequences in VHDL so that they can be well distinguished from the user argorithms implemented in C. The development

[#] hirono@spring8.or.jp

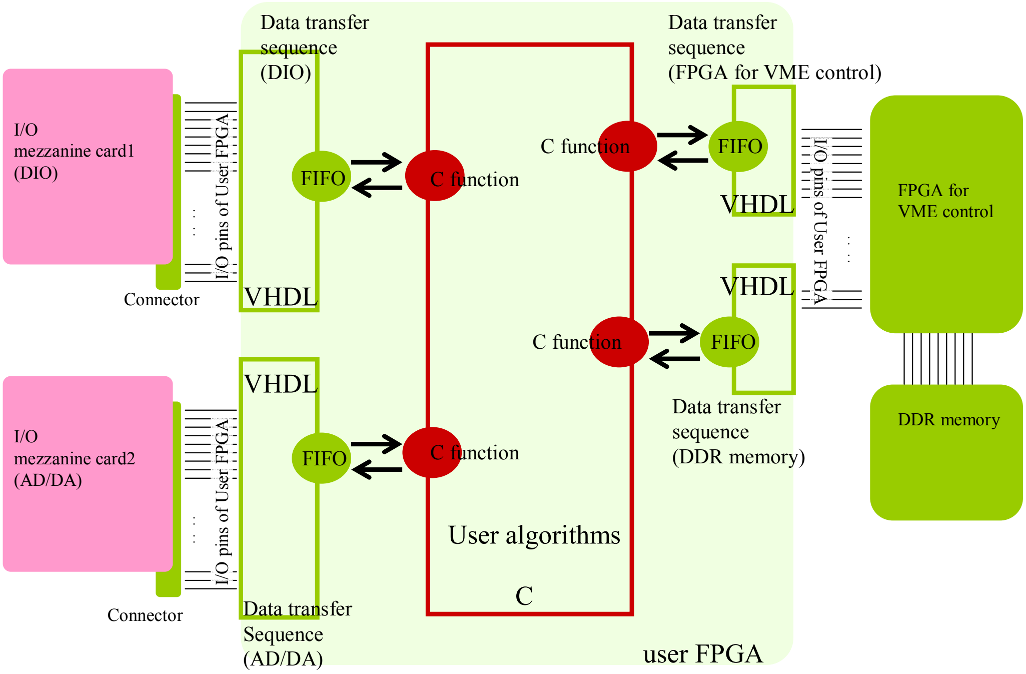
Data transfer sequence (DIO)
Data transfer sequence (FPGA for VME control)
I/O mezzanine card1 (DIO)
I/O pins of User FPGA
FIFO
C function
VHDL
Connector
FPGA for VME control
I/O mezzanine card2 (AD/DA)
Data transfer sequence (DDR memory)
DDR memory
User algorithms
C
Data transfer Sequence (AD/DA)
user FPGA

Figure 1: Block diagram of the flexible logic-reconfigurable VME board. Green blocks represent elements on the base board, and pink blocks represent I/O mezzanine cards. Algorithms in the user's control system are implemented in C. The development environment automatically produces data transfer sequences and generates the FPGA firmware.

environment automatically merges data transfer sequences and C-based argorithms and generates the FPGA firmware. The other components of the base board, such as DDR memory, can be accessed by calling a C function. A block diagram of the board and the architecture inside of the FPGA firmware is shown in Fig. 1.

We built the development enviroment. The development enviroment designed in this study satisfies the three requirements mentioned in this section and can be installed in Windows PCs.

APPLICATIONS

We used the VME board in several applications by using the abovementioned development environment. In one of the applications, the board was used as the controller in a pulse selector for free-electron lasers [3]. The pulse selector was driven by a stepping motor, which was controlled by transistor-transistor logic (TTL) pulses. The main requirement to the controller was to output trapezoidal driving pattern in TTL, of which maximum pulse rate was 86000 pps. Additional functions such as an emergency stop function were added to the controller during the development stage. These functions could be supported because with the new development environment, work was completed within a short time.

We also used this board in a tagging system, which counts until a 60-Hz trigger is applied and outputs the counter value as a tag, and a pulse-train generator system, which outputs a 1-kHz digital pulse train.

UPGRADE OF THE BASE BOARD

Upgrade of the base board was necessary because in the previously developed base board [1], only a VME bus was used as the data transfer interface. The data transmission rate of the VME bus was not sufficiently high for it to be used in data acquisition systems. The new base board was developed with cooperation from ARKUS Inc. [4]. Figure 2 shows the photograph of the new board. Table 1 shows a comparison of the specifications of the new base board and those of the previously developed basic base board.

The upgraded base board has a gigabit Ethernet port, which is controlled by Linux OS that runs on the FPGA. Data collected by the I/O mezzanine cards can be stored in the DDR memory and transferred to the external system via a network by using Linux. Advanced network protocols such as NFS can be used because they are supported by applications bundled with Linux distributions. The user FPGA is upgraded to Xilinx Virtex 5 [5] with a PowerPC hard macro.

The number of I/O mezzanine card interfaces on the basic and upgraded base boards was changed: two interfaces were included in the basic base board and one in the upgraded board. The same I/O cards could be mounted on either base board. Instead of a second mezzanine card, a PCI mezzanine card (PMC) was mounted on the upgraded base board. We developed a PCI bridge driver for making a transparent connection between a PCI connection and any of the PMC.

Consequently, a generic PCI device driver provided by PMC vendors could be used.

The gigabit Ethernet port and the PMC interface enable fast transfer of massive data. The flexible logic-reconfigurable VME board with the upgraded base board can be used in data acquisition systems such as image recorder systems with external data storage facility.

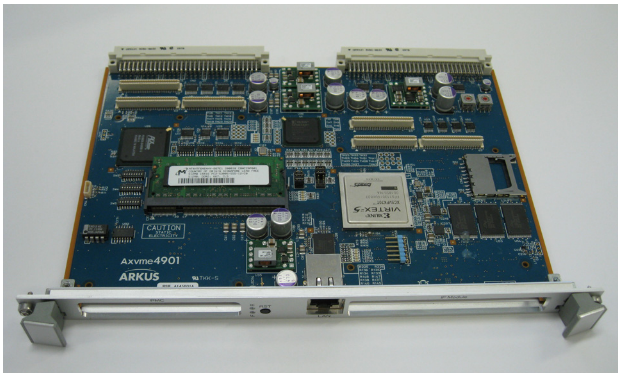

Figure 2: Photograph of upgraded base board of the flexible logic-reconfigurable VME board. PCI bus interface (upper left) and gigabit Ethernet port (bottom) facilitate fast data transfer.

Table 1: Comparison of the Specifications of the Upgraded Base Board of the Flexible Logic-Reconfigurable VME Board and Those of the Basic Base Board

	Upgraded base board	Basic base board
User FPGA	Xilinx Vertex 5 XC5VFX70T	Xilinx Spartan 3 XC3S1500
Bus	VME revision C.3	VME revision C.3
Memory	512-MB DDR 8-MB Flash socket for SD card	256-MB DDR 8-MB Flash
Number of I/O card(s)	1	2
Clock frequency of user FPGA	50 MHz	50 MHz
Data transfer interface	gigabit Ethernet port PCI bus	--
CPU on FPGA (OS)	PowerPC 440 (Timesys Linux)	--

CONCLUSION

We have designed a new development environment of a flexible logic-reconfigurable VME board. With this development environment, the time taken for firmware development can be reduced considerably. We have used our board in various control systems using the development environment.

We also developed a new base board that enables fast data transfer. The development environment and upgraded base board help increase the capability of the flexible logic-reconfigurable VME board.

REFERENCES

[1] T. Hirono, T. Ohata, and T. Kudo, “Developments of the flexible logic-reconfigurable VME board,” Proc. of ICALEPCS 2007, Knoxville, U.S.A., (2007).

[2] http://www.impulseaccelerated.com/index.htm.

[3] T. Ohata, T. Fukui, Y. Furukawa, M. Ishii, M. Kodera, T. Masuda, T. Matsushita, T. Nakatani, M. Takeuchi, K. Tamasaku, R. Tanaka, and A. Yamashita, “SPring-8 Beamline Control System,” Proc. of ICALEPCS'99, Trieste, Italy, (1999) 666.

[4] http://www.arkus.co.jp/index.html.

[5] T. Kudo, T. Hirono, M. Nagasono, and M. Yabashi, “Vacuum-compatible pulse selector for free-electron laser,” Review of Scientific Instruments, Vol. 80, (2009) 093301.

[6] http://www.xilinx.com/.