APPLICATION USING TIMING SYSTEM OF RAON ACCELERATOR

S. Lee[*], CW. Son[†], HJ. Jang[‡], IBS, Daejeon, South Korea

Abstract

RAON is a particle accelerator to research the interaction between the nucleus forming a rare isotope as Korean heavy-ion accelerator. RAON accelerator consists of a number of facilities and equipments as a large-scaled experimental device operating under the distributed environment. For synchronization control between these experimental devices, timing system of the RAON uses the VME-based EVG/EVR system. This paper is intended to test high-speed device control with timing event signal. To test the high-speed performance of the control logic with the minimized event signal delay, we are planing to establish the step motor controller testbed applying the FPGA chip. The testbed controller will be configured with Zynq 7000 series of Xilinx FPGA chip. Zynq as SoC (System on Chip) is divided into PS (Processing System) and PL (Programmable Logic). PS with the dual-core ARM cpu is performing the high-level control logic at run-time on linux operating system. PL with the low-level FPGA I/O signal interfaces with the step motor controller directly with the event signal received from timing system.

This paper describes the content and performance evaluation obtaining from the step motor control through the various synchronized event signal received from the timing system.

INTRODUCTION

The RAON [1] is a new heavy ion accelerator under construction in South Korea, which is to produce a variety of stable ion and rare isotope beams to support various researches for the basic science and applied research applications. To produce the isotopes to fulfill the requirements we have planed the several modes of operation scheme which require fine-tuned synchronous controls, asynchronous controls, or both among the accelerator complexes. RAON, which is a large experimental machine, consists of many experimental devices and additional facilities. For synchronized control under the distributed environment, timing system of the RAON uses the VME-based Event Generator (EVG)/Event Receiver (EVR) [2] system.

TIMING SYSTEM

The timing system uses the EVG/EVR system of the Micro-Research Finland Oy company in VME form factor. The timing system consists of an EVG which converts timing events to optical signals, Fan-Out unit that repeats the signals, an array of EVRs and VME-controller. The EVRs decode the optical signal and produce hardware signal or software interrupt for the timing event. It uses the MVME6100 or 3100 of Emerson company as VME-controller. The controller also uses real-time operating system, VxWorks 6.9 for a fast response. Finally, EPICS IOC (mrfioc2) on VxWorks controls the FPGA-based EVG/EVR boards.

The characteristics of MRF timing system are :

- Event driven system, 256 event codes
- Event generation using external RF reference clock
- 50 ~ 125MHz event clock rate
- Events generated
  - From external HW inputs
  - Two sequencers (up to 2048 events/sequencer)
  - Multi counters
- Cascaded Event Generators
- Different Clock Synchronization

Hardware Configuration

Hardware configuration of the timing system consist of:

- XLI GPS Time System
- Rubidium Frequency Standard Clock Source (FS725)
- Event Trigger System (EVG/EVR,Fanout Repeater/Concentrator)
- MVME 6100/MVME 3100 controller board
- VME Crate (Wiener)
- SMA 100a RF Signal Generator

Figure 1 shows hardware configuration to test timing system.

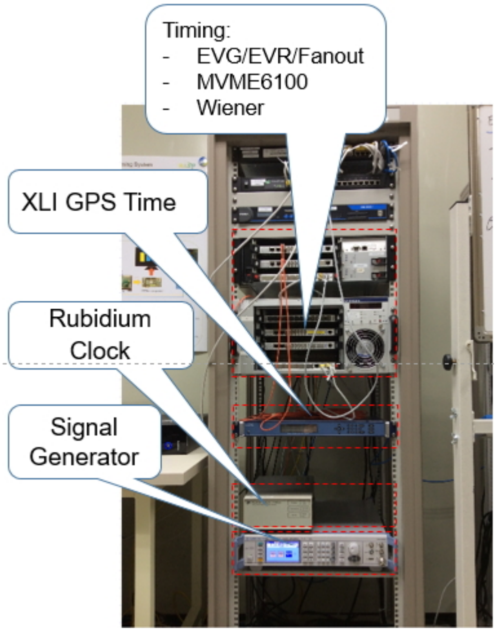

Figure 1: Timing hardware configuration.

[*] silee7103@ibs.re.kr

[†] scwook@ibs.re.kr

[‡] lkcom@ibs.re.kr

Referring to the configuration of Fig. 1, GPS receiver synchronizes with rubidium clock and EVG to 1PPS. Signal generator generates RAON reference clock (81.25 MHz) to EVG. The two input signals of EVG are synchronized by locking the phase. The generated signals from EVG according to event codes registered by mrfioc2 are distributed to EVRs through Fan-out repeater.

Software Configuration

Software modules to operate the timing system consist of:

- Workbench 3.3, vxWorks IDE
- VxWorks 6.9 Realtime Operating System or RTEMS for MVME3100
- EPICS Framework (R3.14.12.5)
- MRFIOC2/SRSIOC
- Network Time Protocol (NTP)

FAST STEPPER MOTOR TESTBED USING TIMING SIGNAL

Configuration of Fig. 2 shows the overall of a fast stepper motor testbed operating by the input of the timing signal. As described already before, the EVG of the timing system receives two input signals, the external event source which is locked to GPS and RF reference clock. The synchronized events of EVG are distributed through optical fiber links to EVRs. The EVRs received some events from EVG decode those events and trigger TTL level signals to ZC706 evaluation board. The PL of zynq which is received from external TTL signal gives the signal to motor controller according to the motor's parameters. The setting of the parameter values for the motor control is performed on the EPICS interface of the zynq PS. PS is equipped with the EPICS software module on the cross-compiled linux OS of the ARM processor. The interface between PS and PL uses the AXI interface of the ARM Advanced Microcontroller Bus Architecture (AMBA). The AMBA [3] is an open-standard, on-chip interconnect specification for the connection and management of functional blocks in system-on-a-chip (SoC) designs. Software module for AXI interface between both area should be also developed in the area of the linux device driver. The types of motor which can be supported by a low voltage drive board of the Analog Device [4] are brushless DC, PMSM, brushed DC or stepper motor.

System on Chip: FPGA - Zynq

Zynq [5, 6] as SoC is divided into Processing System (PS,dual-core ARM cpu) and Programmable Logic (PL, FPGA). Communication between PS and PL is through the AXI interface of ABMA.

ZC706 Evaluation Board of Xilinx

The ZC706 evaluation board for All Programmable SoC (AP SoC) provides a hardware environment for developing and evaluating designs targeting the Zynq®-7000 AP SoC. [7] The ZC706 board has the SMA port which is the programmable user clock for the TTL out signal of the EVR timing board. The reason which selected the board is to have the user clock port to receive input from the external trigger.

Controller for Motor Drive

Figure 3 shows the controller and low voltage drive board of the Analog Devices to drive the motor. Controller board communicates with ZC706 via FPGA Mezzanine Card (FMC) connector. FMC connector of the controller board is connected to XADC interface, digital I/O and sensors, two gigabit ethernet modules, power line and so on. XADC and digital I/O signals among those connections are delivered to the low voltage drive board via the drive board connector. The low voltage drive board received those signals generates Pulse Width Modulation (PWM) signal through the MOSFET gate driver and drives the stepper motor. The low voltage drive board is operated in 12~24V DC power from the external power source.

Linux on Zynq PS (ARM Processor)

To operate linux OS on the ARM processor of the zynq PS there consists of:

- ARM Cross Compile Tool Chain (arm-linux-gnueabihf)
- Linux Kernel Source (Linaro)
- Bootloader (BOOT.BIN)
- Board Support Package (Linux Device Tree)
- Root File System (Busybox)

Linaro [8] is a not-for-profit engineering organization that works on free and open-source software such as the Linux kernel, the GNU Compiler Collection (GCC), power management, graphics and multimedia interfaces for the ARM family of instruction sets and implementations thereof as well as for the Heterogeneous System Architecture. The linaro kernel source is the ubuntu-based kernel source including a lot of device driver such as zynq device tree. The bootloader of PS uses U-Boot [9] supporting for the zynq device of Xilinx. To boot the zynq device, it should be made up the boot.bin file created by Vivado [10] tool. The boot.bin [11] file consists of fsbl.elf (First Stage BootLoader), u-boot.elf (Second BootLoader), uImage (kernel zImage for uboot), zynq.bif (Boot Image Format file) and user.bit (bitstream file of FPGA). The linux kernel is using the root file system generated by the Busybox [12] tool.

Linux Device Driver

Linux device driver for zynq PS uses Industrial IO (IIO) module of the Analog Devices. Libiio [13] is a library that has been developed by Analog Devices for easy interface with IIO devices. Localbackend module of Fig. 2 goes into kernel mode from user mode through the system call and accesses the iio device driver via "struct file_operations". Basically the iio device driver is a character device type.

Software Interface

EPICS IOC is being implemented using the libiio library that abstracted the low-level details of the hardware, as shown in Fig. 2. EPICS is a set of open source software tools,

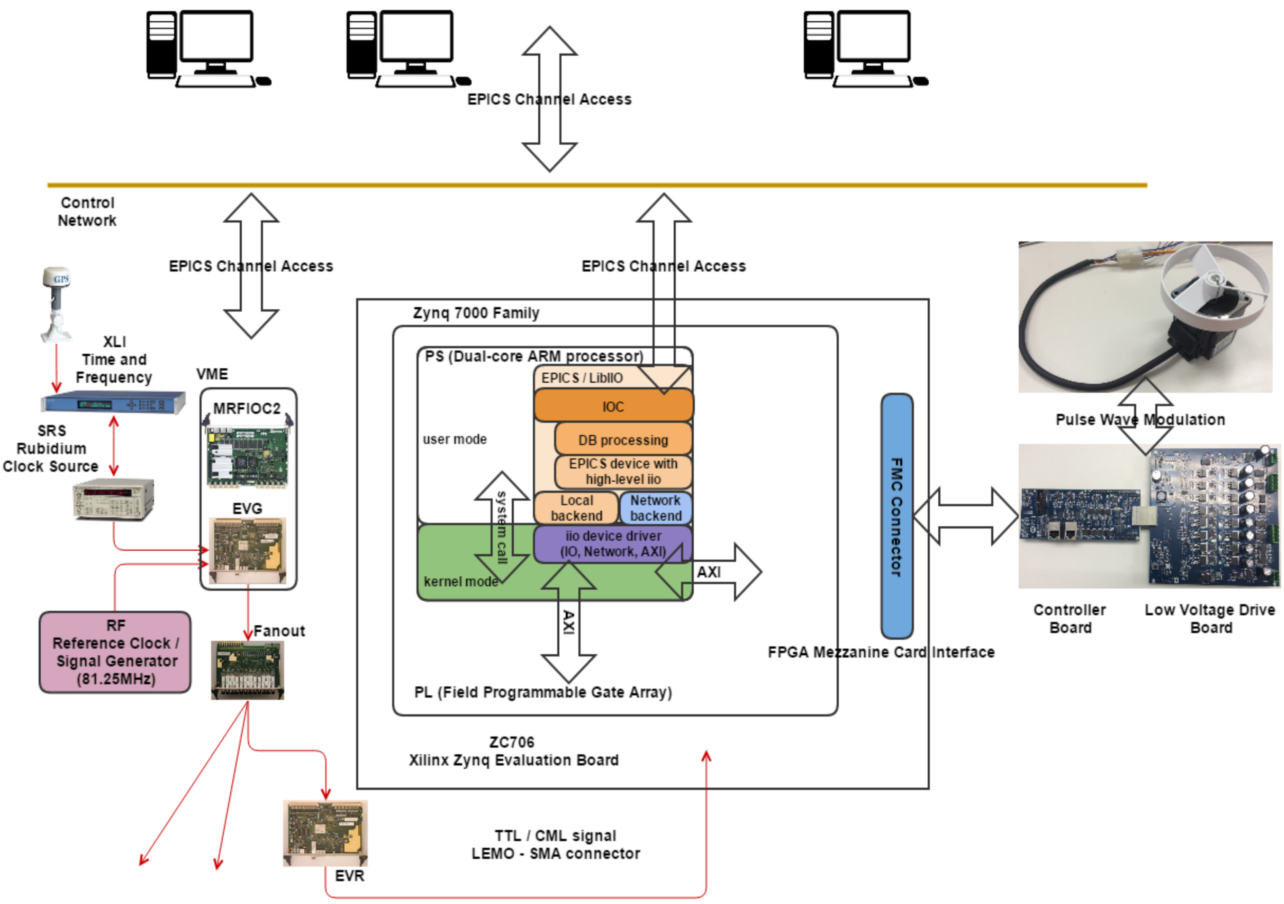

Figure 2: Fast motor testbed adopted zynq device using timing event signal.

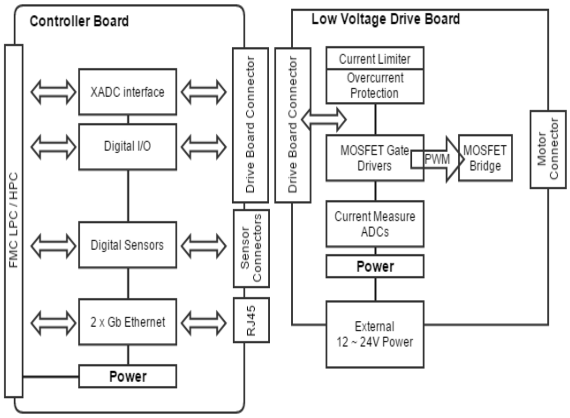

Figure 3: Controller and low voltage drive board.

libraries and applications developed collaboratively and used worldwide to create distributed soft real-time control systems for scientific instruments such as a particle accelerators, telescopes and other large scientific experimental facilities [14]. The device support routine of EPICS is implemented using the libiio library to read or write the parameter values for the motor control. The motor control parameters become the Process Variables (PVs) of EPICS and are carried out the EPICS database processing according to EPICS scan rate.

Also, it is going to develop additional EPICS waveform PV to monitor the PWM signal of the low voltage drive board.

FPGA Interface reuses the Hardware Description Lanaguage (HDL) code which is provided by the Analog Devices. FPGA HDL code is develeped by VerilogHDL using Vivado of Xilinx. When the motor controller was purchased from Analog Devices, it operated on ZedBoard. It should be changed to operate the ZC706 board as well as ZedBoard. It is going to transplant the code of ZedBoard into ZC706 board. In addition, It is being implemented FPGA code to drive the motor receiving the input signals (EVR TTL signals) via user clock port of ZC706 board.

SUMMARY

Table 1, 2 summarize the configuration of hardware and software as shown in Fig. 2.

Table 1: Hardware Configuration for Stepper Motor Testbed

Hardware	Contents	Company
GPS	GPS antena	Symmetricom
	XLi time and frequency	
Clock Sync.	Rubidium frequency	Standford
	standard model FS725	Research
Ref. Clock	Signal Generator or RF	
	Reference (81.25 MHz)	
EVG	Event Generator	Micro
EVR	Event Receiver	Research
FanOut	Event Repeater	Finland Oy
MVME6100	VME Controller	
	Zynq 7000 family Eval. kit	Xilinx
ZC706	Processing System	
	(dual-core ARM processor)	
	Progrmmable Logic	
Controller	Motor controller board	Analog
Low Volt.	Low voltage drive board	Devices
Motor	Stepper or BLDC	

Table 2: Software Configuration for Stepper Motor Testbed

Software	Contents	Module
EPICS	Base R3.14.15.2	PS of Zynq
Libiio	Industrial I/O library	
	supplied Analog Devices	
Kernel	Linaro kernel source	
	including iio device driver	
Bootloader	U-Boot for zynq	
IOC	In-house using Libiio	
FPGA	Analog Devices and	PL of Zynq
	In-house code	
VxWorks	Real-time OS for VME	Timing
	including BSP	
MRFIOC2	EPICS IOC for timing	
SRSIOC	IOC for rubidium clock	
Workbench	Workbench3.3 for VxWorks	Development
Vivado	Vivado14.4 including SDK	Tool
	with node lock license	
Busybox	for rootfs system (free)	
Toolchain	for ARM cross-compile (GNU)	

CONCLUSION

Timing system can distribute the fine synchronized event signal at a high speed. The objective of the stepper motor control testbed is to know how to operate the timing system and how to apply it to the high speed controller. The overall implementation is still underway, however if the distributed control system using EPICS makes a connection with the high speed parallel processing of FPGA, it is possible to improve the performance and efficiency of the control system. Zynq SoC can be considered as an ideal device to implement this high-speed control system.

ACKNOWLEDGMENT

This work is supported by the Rare Isotope Science Project funded by Ministry of Science, ICT and Future Planning(**MSIP**) and National Research Foundation(**NRF**) of Korea(Project No. 2011-0032011).

REFERENCES

[1] Y. K. Kwon, *et. al*,“Status of Rare Isotope Science Project in Korea”, Few-Body Syst 54, 961-966, (2013).

[2] Micro-Research Finland Oy website: http://www.mrf.fi

[3] Advanced Microcontroller Bus Architecture: http://en.wikipedia.org/wiki/Advanced_Microcontroller_Bus_Architecture

[4] Analog Devices website: http://www.analog.com

[5] Xilinx-FPGA website: http://www.xilinx.com

[6] Zynq-Book Document website: http://www.zynqbook.com

[7] ZC706 Evaluation Board Document: http://www.xilinx.com/support/documentation/boards_and_kits/zc706/ug954-zc706-eval-board-xc7z045-ap-soc.pdf

[8] Linaro Document website : https://en.wikipedia.org/wiki/Linaro

[9] U-Boot Document website : http://www.denx.de/wiki/U-Boot

[10] Xilinx Vivado Document website : http://www.xilinx.com/products/design-tools/vivado.html

[11] Boot Image Document website : http://www.wiki.xilinx.com/Prepare+boot+image

[12] Busybox Document website : http://www.busybox.net/

[13] Industrial I/O Document website : https://wiki.analog.com/resources/tools-software/linux-software/libiio

[14] EPICS website : http://www.aps.anl.gov/epics/