ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUMPL03

NEW EPICS/RTEMS IOC BASED ON ALTERA SOC AT JEFFERSON LAB

J. Yan, C. Seaton, T. Allison, B. Bevins, A. Cuffe
Thomas Jefferson National Accelerator Facility, Newport News, VA 23606, U.S.A.

Abstract

A new EPICS/RTEMS IOC based on the Altera System-on-Chip (SoC) FPGA is being designed at Jefferson Lab. The Altera SoC FPGA integrates a dual ARM Cortex-A9 Hard Processor System (HPS) consisting of processor, peripherals and memory interfaces tied seamlessly with the FPGA fabric using a high-bandwidth interconnect backbone. The embedded Altera SoC IOC has features of remote network boot via U-Boot from SD card or QSPI Flash, 1Gig Ethernet, 1GB DDR3 SDRAM on HPS, UART serial ports, and ISA bus interface. RTEMS for the ARM processor BSP were built with CEXP shell, which will dynamically load the EPICS applications at runtime. U-Boot is the primary bootloader to remotely load the kernel image into local memory from a DHCP/TFTP server over Ethernet, and automatically run RTEMS and EPICS. The first design of the SoC IOC will be compatible with Jefferson Lab's current PC104 IOCs, which have been running in CEBAF 10 years. The next design would be mounting in a chassis and connected to a daughter card via standard HSMC connectors. This standard SoC IOC will become the next generation of low-level IOC for the accelerator controls at Jefferson Lab.

INTRODUCTION

The accelerator control system at Jefferson Lab is based on the Experimental Physics and Industrial Control System (EPICS) which uses a client/server model. Currently, the servers, or IOCs consist of VME IOCs, PC104 IOCs and Soft IOCs. PC104 IOCs running Real Time Executive for Multiprocessor Systems (RTEMS) and EPICS have been successfully used in the accelerator control system for the past 10 years with more than 280 in service [1]. However, the PC104 manufacturer, Kontron, has discontinued the PC104 board, so we have to decide whether to source another vendor to continue using the PC104 or seek a new embedded IOC platform that meets our long term goals. The PC104 is a stackable Single Board Computer (SBC) with an ISA bus connecting a carrier board. The data transfer baud rate between the processor and the carrier board is limited by the ISA bus. This limitation pushes us to seek a new IOC which has high bandwidth, especially for those applications which require fast data transfer. Since most of our designs are using a Field-Programmable Gate Array (FPGA) and a separate microprocessor, a System-on-Chip (SoC) FPGA should definitely be considered. The SoC FPGA is likely to provide comparable, if not superior functionality and performance, but at a lower board space, lower power, and lower system cost. The integration of thousands of internal connections between the processor and the FPGA leads to substantially higher bandwidth and lower latency compared to a two-chip solution. Some SoC FPGAs have been designed as EPICS IOCs for other accelerator control systems [2,3]. The Intel (formally Altera) Cyclone V SoC [4] was chosen as our standard IOC platform. In this paper, we will describe the design of the SoC IOC board, the booting of the SoC, and the real-time operating system development.

HARDWARE DESIGN

The Terasic SoCKit and DE0-Nano-SoC Development Kit were chosen as our hardware design reference platforms [4,5]. Both of the kits were built around the Intel/Altera Cyclone V SoC FPGA, which integrates an ARM-based HPS consisting of a ARM Cortex-A9 processor, peripherals and memory interfaces tied seamlessly with FPGA fabric using a high-bandwidth interconnect backbone. On our first design of the SoC motherboard, we keep most features of the SoC kits and add an ISA bus to be compatible with the PC104 IOC. Figure 1 shows the block diagram of the new SoC motherboard. It has the following feature devices:

- Altera Cyclone® V 5CSEBA2U19C8N device.
- Serial configuration device – EPCS.
- USB-Blaster II onboard for programming; JTAG Mode.
- 25MHz clock sources from the clock generator.
- One ISA Bus header.
- 600MHz Dual-core ARM Cortex-A9 processor.
- 1GB DDR3 SDRAM (32-bit data bus).
- One EEPROM for MAC address.
- 1 Gigabit Ethernet PHY with RJ45 connector.
- Micro SD card socket.
- Two UART to RS-232 10-pin connectors.
- One 64 MB QSPI Flash.
- One header for Reset, Led and Speaker.
- One user button and one user LED.
- Switches for boot selection.

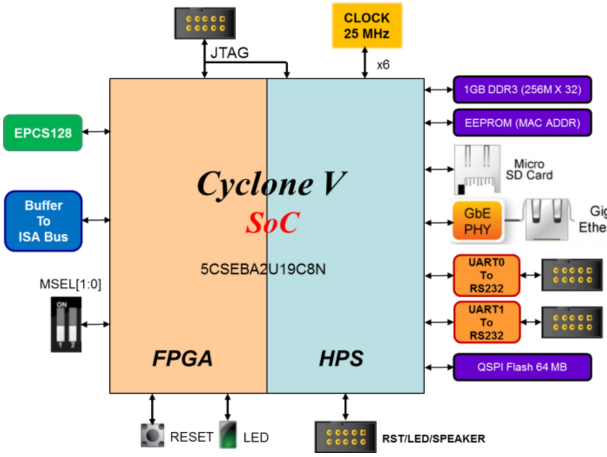

Figure 1: Block Diagram of the SoC IOC.

The SoC board is a 10 layer Printed Circuit Board (PCB) designed with Altium Designer software. Figure 2 shows the picture of the SoC IOC motherboard. It has the same standard size as our existing PC104 SBC and will be a direct replacement.

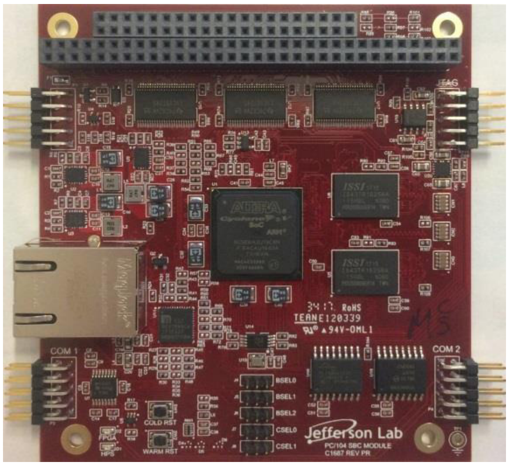

Figure 2: The Picture of the SoC IOC Board.

SOFTWARE DEVELOPMENT

The control system standard at Jefferson Lab requires new IOCs to remotely boot the operating system, dynamically load the objects, database, and EPICS, just like the present VME and PC/104 IOCs. The software development of the SoC IOC will meet this standard. In this section, We introduce the SoC IOC's booting, the real-time operating system RTEMS, the building of the RTEMS kernel with a generic "shell" CEXP and generic system application (GeSys), and finally EPICS and its applications.

U-Boot

The Universal Boot Loader (U-Boot) is an open source, primary boot loader used in embedded devices to package the instructions to boot the device's operating system kernel. U-Boot is both a first-stage and second-stage bootloader. It is loaded by the system's ROM or BIOS from a supported boot device, such as SD card, SATA driver, NOR flash, or NAND flash. The first stage is configuring memory controllers and SDRAM, and the second-stage is performing multiple steps to load an operating system from a variety of devices that must be configured, presenting a menu for users to interact with and control the boot process. Figure 3 shows the boot stages of the Golden System Reference Design (GSRD) boot flow [6]. Upon power on, the processor executes the BootROM code which resides in on-chip ROM. It initializes all required hardware components and fetches the Preloader binaries form the boot device. The Preloader configures clocking, IOCSR, pin-muxing, SDRAM and loads U-Boot. It is based on the SPL (Secondary Program Loader) which is a component of U-Boot. U-Boot configures the FPGA, and finally loads an Operating System (OS), such as Linux, RTEMS, VxWoks, etc.

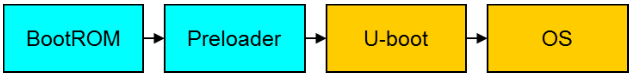

Figure 3:The GSRD Boot Flow.

The SD card is a very convenient method to boot the SoC board, and can be easily re-created and updated. Figure 4 presents the layout of the SD card that is used by the GRSD. The Master Boot Record (MBR) partition contains descriptions of the partitions on the card. Partition 1 contains the Device Tree Blob files (soc_system.dtb), compressed FPGA configuration file (soc_system.rbf), compressed Linux kernel image file (zImage), and U-Boot script for configuring the FPGA (u-boot.scr). Partition 2 is the Linux root file system. Partition 3 contains the Preloader image and U-Boot image. It is a custom partition with type=0xa2, and is required by the boot ROM which loads the Preloader from the beginning of it. In order to run RTEMS, the RTEMS image file (rtems.bin) is copied in the partition 1, and the file u-boot.scr is updated to load RTEMS instead of Linux. When it is booting, U-Boot will read the MAC address from the EEPROM chip and assign it to the environment parameter 'ethaddr', and run a Dynamic Host Configuration Protocol (DHCP) to obtain an IP address from the HDCP server and request a Trivial File Transfer Protocol (TFTP) server to download the RTEMS kernel over the Ethernet. U-Boot is a powerful and user-friendly boot loader for the embedded SoC.

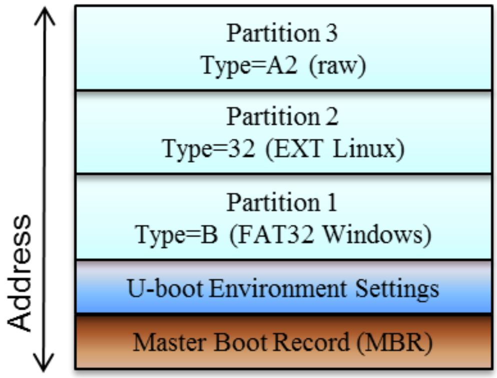

Figure 4: SD Card Layout.

RTEMS

RTEMS is an open-source real-time operating system designed for embedded systems [7]. It has been ported to various target processor architectures and provided a variety of Board Support Packages (BSPs) including commercial-off-the-shelf (COTS) hardware and custom hardware. RTEMS version 4.11 and 4.12 were built for our target processor Cyclone V SoC which has the BSP of altcycv_devkit or altcycv_devkit_smp. Many example applications have been built and tested to check the network, I2C bus and I/O interfaces. RTEMS GeSys from SLAC [8] was chosen as the system application which simply fires up the system and starts CEXP. CEXP is a C expression interpreter command line tool. It can be used as a general purpose shell for RTEMS, but most importantly it provides run-time dynamic loading of code, such as EPICS applications. The RTEMS GeSys executable file was built with two stages of compiling and linking. First of all, compiling and linking all the application source codes, including the linkcmds file of the specific BSP, to generate all the object files. Then, using the tool 'ldep' to create the system symbol table file 'allsysms.c' which includes symbols of all parts of the core libraries (libc, libgcc, libstdc++, RTEMS managers, CPU/BSP support, network, etc.), all object files, and other additional libraries which will possible be used by modules loaded into to the running system. Second, compiling the allsyms.c file and linking it with all object files and libraries to create the system executable file 'rtems.exe'. Finally, using 'objcopy' to generate the system image file 'rtems.bin' which can be loaded by U-Boot.

After the RTEMS image is loaded in the target memory, the system initializes RTEMS, starts the network, mounts the NFS client, and runs the CEXP shell. The CEXP shell dynamically loads the executable file from a remote NFS server, and then runs the applications. We are now on the development stage of porting EPICS to RTEMS and the Cyclone V SoC.

CONCLUSION

The new IOC board based on the Altera Cyclone V SoC FPGA is being designed and built. The first design is a PC104 compatible board which can replace current PC104 IOCs in our control system. It can be booted with a SD U-Boot and run the RTEMS kernel. The RTEMS system application built with CEXP will dynamically load the EPICS application at the run-time. However, there is a lot of work to be done to optimize the RTEMS primary application and port EPICS to RTEMS and the SoC BSP. Eventually, the standard SoC IOC will be design to be mounted in a chassis and connected to a daughter card via standard HSMC connectors. And, this SoC IOC will be the new low-level IOC for accelerator controls at Jefferson Lab.

ACKNOWLEDGMENT

Thanks to Till Straumann from SLAC for providing the CEXP source code and technical support.

Notice: Authored by Jefferson Science Associates, LLC under U.S. DOE Contract No. DE-AC05-06OR23177. The U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce this manuscript for U.S. Government purposes.

REFERENCES

[1] J. Yan, etc. "PC/104 Embedded IOCS at Jefferson Lab", Proceedings of ICALEPCS'09, Kobe, Japan, page 230-232.

[2] Y.S. Cheng, etc. "Embedded Environment with EPICS Support for Control Applications", Proceedings of ICALEPCS'15, Melbourne, Australia, page 107-109.

[3] T. Xue, etc. "Design of EPICS IOC Based on RAIN1000Z1 ZYNQ Module", Proceedings of *ICALEPCS'15*, Melbourne, Australia, page 905-906.

[4] SoCKit – the Development Kit for New SoC Device, https://www.terasic.com.tw/cgibin/page/archive.pl?CategoryNo=167&No=816.

[5] DE0-Nano-SoC Kit/Atlas-SoC Kit, http://www.terasic.com.tw/cgibin/page/archive.pl?Language=English&No=941

[6] RocketBoards – Documntation Portal, https://rocketboards.org/foswiki/view/Documentation /GSRDBootFlow

[7] RTEMS Real Time Operating System website, http://www.rtems.org/.

[8] RTEMS Software, http://www.slac.stanford.edu/~strauman/rtems/software.html