

## Implementation of an EPICS IOC on an Embedded Soft Core Processor Using Field Programmable Gate Arrays<sup>1</sup>

Douglas Curry, Alicia Hofler, Hai Dong, Trent Allison, Curt Hovater, Kelly Mahoney  
*Jefferson Lab, Newport News, USA*

### Abstract

*At Jefferson Lab, we have been evaluating soft core processors running an EPICS IOC over  $\mu$ Linux on our custom hardware. A soft core processor is a flexible CPU architecture that is configured in the FPGA as opposed to a hard core processor which is fixed in silicon. Combined with an on-board Ethernet port, the technology incorporates the IOC and digital control hardware within a single FPGA. By eliminating the general purpose computer IOC, the designer is no longer tied to a specific platform, e.g. PC, VME, or VXI, to serve as the intermediary between the high level controls and the field hardware. This paper will discuss the design and development process as well as specific applications for JLab's next generation low-level RF controls and Machine Protection Systems.*

### Introduction

At present, to be accessible to the accelerator's distributed control system, our field hardware must be designed to operate with a VME or VXI based single board computer Input/Output controller (IOC). Frequently, this requires long cable pulls and in some cases a new VME crate installation. During the initial design cycle, typically we must choose between the VME platform, serial communication, or blind stand-alone units. Stand-alone units provide no feedback to the control room and there is no convenient way to verify that the devices are functioning correctly. Serial devices are severely band limited and require expensive serial interface equipment when the standard IOC ports have been fully utilized. The VME based systems are expensive and often do not accommodate available space constraints. The ability to integrate the IOC directly into our custom hardware provides enormous flexibility and provides significant enhancements to our operating capabilities.

Field Programmable Gate Arrays (FPGAs) are very attractive for implementing digital designs. They have high component counts and are relatively low cost. Their highly configurable platform provides enormous design flexibility, allowing changes to take place during the development process and even well after the board design has been completed. Typically modern digital designs are generated and simulated using a standardized hardware description language such as VHDL or Verilog, synthesized, and then routed for a particular device.

Generally, embedded systems follow the master/slave approach to integrating system designs. However, with the advent of soft core processors, an FPGA can host the application logic and the processor core running a fully featured operating system within a single package. Embedding the processor core within the logic fabric offers tighter integration with the application design logic and greater flexibility in the overall system architecture, and, in many cases, it is expected to have higher performance by avoiding master-slave communication bottlenecks [1].

Occasionally our equipment must be installed and operated in radiation environments in order to support end-station experiments and accelerator operations. Therefore it is reasonable to expect single event upsets (SEUs) in the configuration memory of the FPGA [2]. Modern FPGA families include the ability to perform CRC checks on the configuration memory while the device is in operation [3]. The device can be set up to reconfigure itself after the detection of a soft error if desired. Reconfiguration times are fairly quick on the order of a few milliseconds. It should be noted that device operation is unavailable during this period of time. However, the overall benefit provides significant advantages over manually resetting the device especially when it is located in a controlled environment.

---

<sup>1</sup> This work is supported by DOE contract DE-AC05-84ER40150 Modification No. M175, under which the Southeastern Universities Research Association (SURA) operates the Thomas Jefferson National Accelerator Facility.

## Soft Core Processors

Soft core processors differ from fixed processor cores in the way that they are implemented. Soft cores are configured to meet an application's needs by using an interface provided by the FPGA vendor, adding performance features and peripherals as required by the end user. The "final" core is generated in either VHDL or Verilog and can be targeted towards any modern FPGA family. Once an FPGA is programmed with the core, the architecture itself is fixed. Altera and Xilinx both provide soft core processors that target their FPGA families and take full advantage of a device's enhanced features such as embedded multipliers and memory blocks.

The Altera NiosII soft core processor is a 32-bit RISC architecture that supports separate instruction and data buses, therefore classifying it as a Harvard style architecture [4]. There are 3 different implementations a user can choose from. Starting with the most basic core, the NiosII/e "economy" version is a single staged pipeline with no instruction or data cache. This core executes a single instruction at a time making branch prediction hardware unnecessary and the core therefore much less complicated to implement requiring a minimal amount of FPGA resources. This core trades off FPGA resources at a significant cost in performance. Altera's performance core, the NiosII/f "fast" version is a 6 stage pipeline with dynamic branch prediction, user defined instruction and data cache sizes, and an ALU that can take advantage of the internal multiplier blocks. This core provides the best performance at the cost of additional resources.

It is not necessary to build new cores for each design. The "final core" can be ported to multiple designs and any of the modern Altera FPGA families. After a core is defined, it is easily integrated with the application logic, synthesized, and routed for the target device.

## $\mu$ Clinux

Linux is a fully functional modular operating system which makes it extremely scalable by removing utility programs, tools, and other system services that are not needed in an embedded environment [5]. Running a fully featured operating system over our custom hardware provides multiple advantages over single threaded operating environments. A Linux operating system provides a platform that supports remote logins, the ability to execute user programs and accelerator distributed control system applications, network file system (NFS) access, as well as a simple web server if desired.

Currently, the NiosII architecture does not include a memory management unit (MMU). This requires the use of an operating system such as  $\mu$ Clinux.  $\mu$ Clinux is a port of the Linux kernel that supports embedded processors lacking an MMU. From the application programming perspective, it offers an interface very similar to a standard Linux system.

## EPICS

The Experimental Physics and Industrial Control System (EPICS) is used extensively throughout the Jefferson Lab accelerator and experimental end-stations [6]. A typical low-level device control EPICS application at Jefferson Lab resides on a VME based IOC running vxWorks. With the EPICS IOC environment running directly on our custom hardware, we gain tremendous flexibility in our board designs and their portability. We are no longer fixed to VME or VXI board sizes, so the final printed circuit board can be built to accommodate a custom chassis or frame size that is dictated by the system requirements or available space constraints as opposed to the accelerator control system hardware environment. The details of how to communicate with the device almost becomes a moot point in the design process.

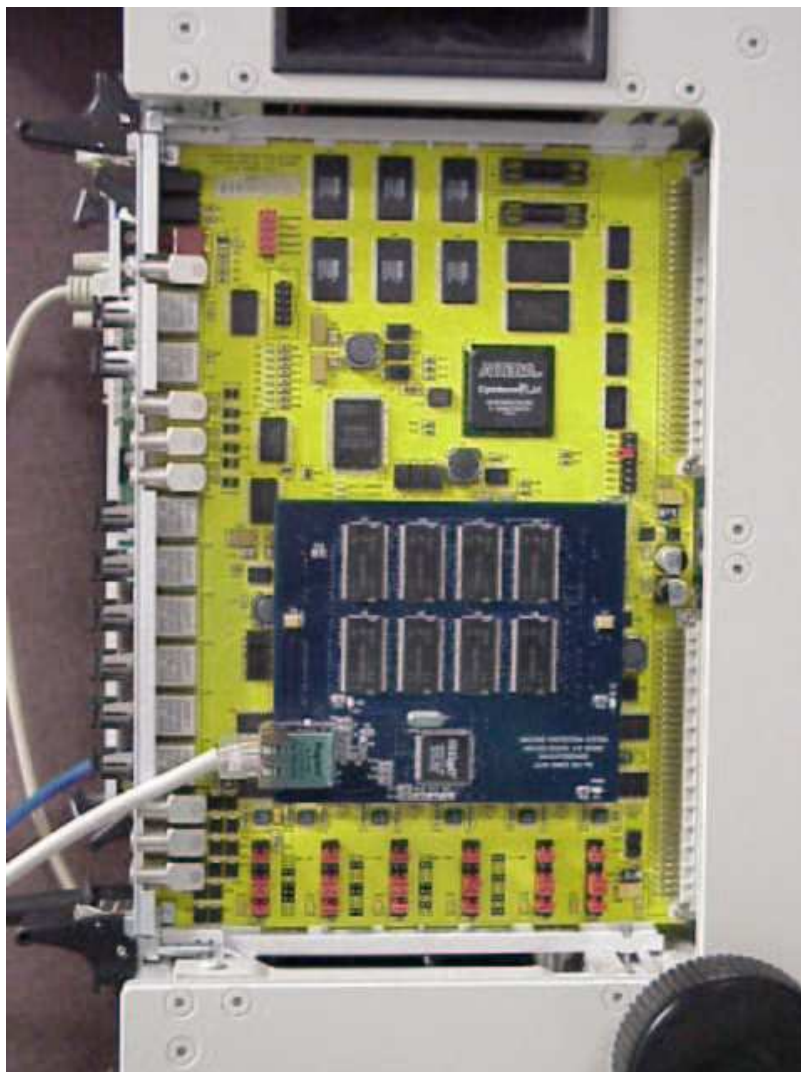
The initial cross-compiling of EPICS for  $\mu$ Clinux at Jefferson Lab was done using the cygwin environment over Windows XP [7]. Cygwin is a port of the GNU development environment for Microsoft Windows. However, Altera's development tools are also supported for Linux, and EPICS builds for  $\mu$ Clinux should also compile in this environment as well.

## Custom Hardware

Setting up a custom hardware design to accommodate the embedded system requires only a minimal amount of board real-estate and components. A fully functioning Nios II processor running on our custom hardware at Jefferson Lab only requires the addition of SDRAM, Flash, and Ethernet controller hardware. With any system, the SDRAM is required to run the operating system, any user programs, the EPICS IOC environment and low level device control applications, and the read/write portion of the file system. The Flash provides permanent storage for the Linux kernel, file system,

and any user applications. During initial power up, the kernel is loaded from the Flash and executes in the SDRAM. The Ethernet controller and RJ-45 connector establish the connection to the local area network. Any EPICS files required to operate the IOC should be loaded from the network file system to ensure proper file revisions are being utilized.

Nios II cores have been built and tested for our new low level RF control module using Altera's EP1S20 Stratix FPGA, and the machine protection analog and digital I/O module using Altera's EP2C35 Cyclone II FPGA. The Nios II cores consume on average about 5 thousand logic elements, which is approximately 15 to 25% of the FPGA's resources [8][9].



**Figure 1. Jefferson Lab's Machine Protection Analog and Digital I/O Module.**

### Summary

Running an EPICS IOC directly over our custom hardware provides tremendous design flexibilities by allowing us to focus our attention on the problem to be solved without adding additional constraints of how to communicate and transmit data to or from the device. With the integrated 10/100 Ethernet controller hardware there is sufficient band-width and it provides a popular platform from which we can communicate with our custom hardware. The use of an embedded IOC within the FPGA fabric simplifies board designs by eliminating the typical master/slave design approach and only requires the addition of a few extra components.

New boards being developed at Jefferson Lab already include the peripherals to accommodate the Nios II architecture. By adding a daughter card populated with Ethernet control hardware, the machine protection analog and digital I/O module shown in figure 1 is fully capable of running an

EPICS IOC. SDRAM and Flash memory is already incorporated as part of the standard configuration of the main board design. The new low-level RF control module currently under development at Jefferson Lab is also built with the necessary hardware to support an embedded EPICS IOC. Both of these designs have been successfully configured with the Nios II processor and have been running  $\mu$ Clinux for some time. The processor core and operating system are proving to be very stable and reliable platforms.

We have also cross-compiled EPICS for  $\mu$ Clinux and are presently in the beginning stages of executing and troubleshooting embedded IOCs on the Nios II architecture.

## References

- [1] J.A. Williams and N.W. Bergmann "Reconfigurable Linux for Spaceflight Applications," MAPLD International Conference, 2004
- [2] D.E. Johnson, K.S. Morgan, M.J. Wirthlin, M.P. Caffrey, P.S. Graham "Detection of Configuration Memory Upsets Causing Persistent Errors in SRAM-based FPGAs," MAPLD International Conference, 2004
- [3] Altera "Error Detection Using CRC in Altera FPGA Devices," Altera Corporation, July 2004, Version 1.0
- [4] Altera "Nios II Processor Reference handbook," Altera Corporation, May 2005, Version 5.0.0
- [5] M. Hennerich, J Hennerich "uClinux as an Embedded OS on an Embedded Processor," Analog Devices Inc. 2004
- [6] L.R. Daliesio (LANL), M.R. Kraimer (ANL), A.J. Kozubal (LANL) "EPICS Architecture," MS H820, Los Alamos National Laboratory, Los Alamos NM, Argonne National Laboratory, Argonne IL, September 1993.
- [7] D. Curry, A. Hofler "Compiling EPICS for uClinux targeting the Nios II architecture," September 2005s
- [8] Altera "Stratix Architecture," Altera Corporation, July 2005
- [9] Altera "Cyclone II Architecture," Altera Corporation, July 2005