Experience with FPGA-based processor core as Front-end Computer

L.T. Hoff, BNL, Upton, NY, USA*

Abstract

The RHIC control system architecture follows the familiar “standard model”. LINUX workstations are used as operator consoles. Front-end computers are distributed around the accelerator, close to equipment being controlled or monitored. These computers are generally based on VMEbus CPU modules running the VxWorks operating system. I/O is typically performed via the VMEbus, or via PMC daughter cards (via an internal PCI bus), or via on-board I/O interfaces (Ethernet or serial). Advances in FPGA size and sophistication now permit running virtual processor “cores” within the FPGA logic, including “cores” with advanced features such as memory management. Such systems offer certain advantages over traditional VMEbus Front-end computers. Advantages include tighter coupling with FPGA logic, and therefore higher I/O bandwidth, and flexibility in packaging, possibly resulting in a lower noise environment and/or lower cost. This paper presents the experience acquired while porting the RHIC control system to a PowerPC 405 core within a Xilinx FPGA for use in low-level RF control.

Introduction

The popularity of Software Defined Radio (SDR) for military and other applications has led to a standardization of LLRF control architectures. The SDR architecture consists of three main elements. Wideband analog to digital converters (ADCs) are used for signal capture. Wideband digital to analog converters (DACs) are used for signal generation. The third element is some sort of discrete component such as a programmable logic devices (PLD), digital signal processor (DSP), or general purpose microprocessor. These devices are used to perform signal extraction, down-conversion, demodulation, filtering, and to run feedback and other control algorithms. Within the general SDR architecture there are many options for selecting specific components, packaging those components into a complete system, and interfacing with the accelerator control system.

The accelerator control system is generally used to control and monitor performance of the feedback algorithms and other aspects of the signal processing. In addition to providing transparency into system operations, the accelerator control system supports standard techniques for archiving performance data in such a way that it can be correlated with other archived accelerator data, as well as standard techniques for saving and restoring sets of system configurations. The need to interface with the accelerator control system, through a supported communication channel, may influence the packaging of the LLRF control system.

History

The RHIC accelerator control system standardized on VME enclosures for distributed, embedded front-end computer systems. [1] Adhering to this standard, the RHIC

*Work performed under the auspices of the US Department of Energy

LLRF system was composed of VME modules, packaged within VME enclosures, and interfaced to the accelerator control system via the VME bus. [2] The RHIC LLRF control system is based around a commercially available 4-DSP VME module. The accelerator control system initially supported controlling and monitoring a few dozen scalar quantities, at speeds up to 720Hz, through a proprietary communication protocol between the accelerator control system processor and the LLRF DSP using the VME bus. During system commissioning it became clear that 720Hz time resolution data was not sufficient in all cases. To address this issue within VME bus bandwidth constraints, a technique was developed to support periods during which, in lieu of standard data reporting, data collected at bursts of higher speeds is transmitted.

Despite the bandwidth limitation, the RHIC LLRF system architecture has proved successful, and is still in use largely unchanged. Concerns have been raised regarding parts obsolescence in this rapidly evolving field of technology, and the potential for noise to be introduced by the switching power supplies and VME bus activity with the VME enclosure.

The LLRF control system for the SNS accumulator ring, developed at BNL, is architecturally similar to the RHIC LLRF control system. [3] In the time between the design of the two control systems, technology evolved considerably, and the SNS LLRF control system reflects this evolution. The SNS LLRF system is based around a 4-DSP PMC “daughter card”. This “daughter card” attaches directly to the accelerator control system processor. It is interfaced via a PCI bus, rather than the VME bus, also using a proprietary communication protocol. The increased processing power of the SNS DSPs, and the greater bandwidth afforded by the PCI bus allows the transmission of several arrays of turn-by-turn (1Msample/second over a few milliseconds) data on every SNS cycle (up to 60Hz).

The increased performance of the SNS LLRF system has set the standard for future LLRF systems developed at BNL. The need for several new LLRF systems is foreseen in order to support a variety of accelerator upgrades and new accelerator initiatives. An attempt is underway to adopt a single architecture, using common components for all anticipated LLRF systems.

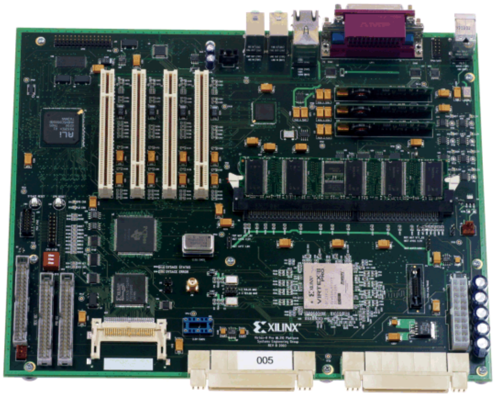
Technology continues to evolve, affording new possibilities. Advances in field-programmable gate array (FPGA) size and sophistication now permit running virtual processor “cores” within the FPGA logic. The virtual “core” can be used to run the accelerator control system software. Tight coupling with the FPGA logic permits high communication bandwidth. Small size and power consumption of the complete system affords flexibility in packaging, possibly reducing cost and improving the noise environment.

The RHIC accelerator control system runs on a number of different processor platforms. The control system requires a 32-bit processor, Ethernet, and a serial port. The most common platform is a PowerPC 603 processor on a VME module.

In order to gain experience with a FPGA-based system, an evaluation kit was procured from an FPGA vendor (Xilinx). [4] The evaluation kit consisted of an evaluation board (model ML310), and development software. The development software included support for developing FPGA logic, and for configuring and programming a PowerPC 405 processor “core”. This evaluation kit was chosen for its similarity to common RHIC accelerator control system platforms.

ML310

The ML310 is full-featured, including 256Mbytes of RAM, a 100Mbit Ethernet port, a serial port, a CompactFlash card, 4 PCI slots, and two proprietary (RocketIO) I/O connectors.



The development software supports several embedded operating systems, including VxWorks, LINUX, and QNX. The RHIC control system typically runs on VxWorks. Therefore efforts focused on evaluating the possibility of running RHIC controls software on the ML310 using VxWorks.

Progress to date

The results to date have been mixed. The evaluation kit supplied by Xilinx came sensibly pre-configured and fully documented. Online documentation includes tutorials and user guides, board schematics, “Gerber files”, and component data sheets. “Quick Start” documentation permitted rapid initial progress. Subsequently, a number of issues have arisen which have somewhat hindered continued progress.

The ML310 can boot from a CompactFlash card. The evaluation kit includes a CompactFlash card containing a VxWorks 5.5 kernel, configured with Ethernet and serial port support. The serial port provides access to the VxWorks shell. The shell can be used to set system parameters, such as network address. Network services include FTP, allowing network file transfer through the ML310 to the CompactFlash card. This service can be used to load executable files and even new boot images to the ML310. The CompactFlash card is formatted to contain two “scratch areas” where users are free to load new boot images without overwriting default boot images.

Although the ML310 could boot and run VxWorks, several issues needed to be addressed before RHIC control system software could run successfully. The RHIC software was compiled for the PowerPC 603, not the PowerPC 405, using VxWorks 5.4, not VxWorks 5.5. The RHIC control system software relies on network services, such as Network File System (NFS) client software, which are absent in the default boot image.

Some setbacks

Xilinx FPGA development tools are capable of generating VxWorks Board Support Packages (BSPs) for custom system designs. Xilinx provides extensive on-line resources, including pre-configured VxWorks 5.5 BSPs. Despite minor issues with these files, such as a misconfigured serial baud rate, and the need for Ethernet device driver patches, these files proved to be more convenient than generating new ones.

The VxWorks vendor (Wind River Systems) treats the PowerPC 405 processor as a separate development target than the PowerPC 603. This meant that an additional development license and additional software installation were necessary before the VxWorks 5.5 upgrade could be installed.

Once VxWorks 5.5 for the PowerPC 405 was installed, and the necessary BSP and patch files were downloaded from the Xilinx web site, configuring and building the VxWorks kernel proceeded smoothly.

Creating a bootable image requires combining the VxWorks kernel with the FPGA logic into a Xilinx “ACE file”. This process can be performed from within the FPGA development environment. It can also be performed from a command-line, by executing the tcl script file “genace.tcl”. The “genace.tcl” script provided in the FPGA development environment does not support the ML310. An enhanced “genace.tcl” must be downloaded from the ML310 section of the Xilinx web site. Learning this subtlety proved to be difficult partially because of somewhat unresponsive customer support services.

Once the appropriate “genace.tcl” was installed, a simple shell script was created to create the “ACE file”. The “ACE file” was then loaded into a “scratch area” on the Compact Flash card via FTP, and the ML310 successfully booted the newly created VxWorks kernel.

Up to that point in time, ML310 kernel development was isolated from the rest of the RHIC controls system VxWorks kernel development. For completely independent reasons, a project began to upgrade the RHIC VxWorks release level to 5.5 during the current shutdown period. Therefore further development has been put on hold while the ML310 software is integrated into the mainstream RHIC VxWorks installation.

Conclusion

As of this time, the system is not fully operational, and the details of interfacing with FPGA logic and the communication bandwidth are still untested. These issues are expected to be successfully addressed shortly.

A challenge that is likely to remain involves managing the VxWorks kernel configuration. FPGA logic development, and updates to the FPGA development software may impact the kernel configuration and required operating system release level. This may also result in tighter integration between RF and controls group development efforts.

Despite setbacks in progress to date, the BNL outlook remains optimistic. There is continued belief that embedded controller architectures similar to ML310 have a role to play in control system architectures in general and BNL LLRF systems in particular.

Xilinx support for VxWorks and extensive on-line resources were immensely helpful. Xilinx provides similar resources for other embedded operating systems, including LINUX and QNX. Enterprising developers could use these examples to provide their own support for additional operating systems, such as RTEMS.

References

[1] D.S.Barton et al, *RHIC control system* (Nucl. Instr. and Meth. A 499 (2003) 367-371)
[2] J.M. Brennan, et al, *RF Beam Control System for the Broookhaven Relativistic Heavy Ion Collider, RHIC* (EPAC 1998, Stockholm, Sweden)
[3] K. Smith, et al, *Progress on the SNS Ring LLRF Control System* (PAC 2003, Portland, OR)
[4] K. Smith et al, *RF Developments at the BNL Collider-Accelerator Department* (Workshop on Low Level RF CERN, Geneva, 10-13 October 2005)
[5] http://www.xilinx.com/ml310