REALIZATION OF A CUSTOM DESIGNED FPGA BASED EMBEDDED CONTROLLER*

F. Severino†, M. Harvey, T. Hayes, L. Hoff, R. Lee, P. Oddo, K. S. Smith
BNL, Upton, NY 11973, U.S.A.

Abstract

As part of the Low Level RF (LLRF) upgrade project at Brookhaven National Laboratory's Collider-Accelerator Department (BNL C-AD), we have recently developed and tested a prototype high performance embedded controller. This controller is a custom designed PMC module employing a Xilinx V4FX60 FPGA with a PowerPC405 embedded processor, and a wide variety of on board peripherals (DDR2 SDRAM, FLASH, Ethernet, PCI, multi-gigabit serial transceivers, etc.). The controller is capable of running either an embedded version of LINUX or VxWorks, the standard operating system for RHIC front end computers (FECs). We have successfully demonstrated functionality of this controller as a standard RHIC FEC and tested all on board peripherals. We now have the ability to develop complex, custom digital controllers within the framework of the standard RHIC control system infrastructure. This paper will describe various aspects of this development effort, including the basic hardware, functional capabilities, the development environment, kernel and system integration, and plans for further development.

INTRODUCTION

The primary goal of the C-AD LLRF upgrade is to develop a stand alone, general purpose, digital signal processing platform which can be used to implement a common LLRF control architecture across the C-AD complex. The standard controller will comprise a 19" rack mount carrier board hosting several daughter modules (ADC, DAC, DSP) providing the specific controller functionality. Moving away from our historical use of VME, the controllers are designed as stand alone to provide for easy and flexible deployment as needed and to give us greater control over the electronic noise environment, a particularly acute issue for RHIC RF systems. The internal carrier-daughter architecture provides us with flexibility of controller configuration options (system feedback controller for RHIC, cavity controller, BPM processor, etc.) depending on daughter configuration and system firmware implementation. It also helps to stave off early obsolescence problems by modularizing the system, and allows for easy integration of follow on modules as needed.

The key feature of the new controller is the implementation of an FPGA based embedded PowerPC processor running VxWorks, and configured as a standard RHIC FEC (equivalent to an EPICS IOC). The requirement to function transparently as a standard RHIC FEC was adopted early, in order that the new controllers integrate seamlessly into the existing C-AD controls network. In support of the processor and daughter modules, the carrier will also feature a number of peripherals, including: 10/100 Ethernet on Cu and gigabit Ethernet on fiber, 128MB SDR SDRAM, up to 2GB DDR2 SDRAM, FLASH memory, and a cross-point switched array of 3.125 Gb/s serial communication links capable of either custom protocols, or standard protocols such as PCIe. Remote carrier and daughter FPGA reconfiguration via Ethernet will also be supported.

Because so many new technologies were being integrated into the new LLRF controller, a prototype controller was designed and fabricated in a PCI mezzanine card (PMC) physical format to provide us with proof of principle testing. With the exception of gigabit Ethernet and the remote FPGA reconfiguration function, this prototype included all of the major peripherals mentioned above. For over one year now, the prototype has been operating as a RHIC FEC, essentially indistinguishable from the standard RHIC MVME-2100 based FECs, and has served as the test and development platform for our continuing embedded development work. The experience gained has provided us with a great deal of confidence in the overall concept for our new LLRF controller, and we are now moving forward with development of the first complete carrier prototype and the RF ADC and DAC daughter modules.

BASIC HARDWARE

The major hardware elements on the board (Figure 1) are the FPGA device, a 100Mb/s Ethernet port, a RS-232 serial port, a DDR2 memory module, discrete SDR SDRAM, one on-board flash device, one 32-bit/33 MHz PMC connector set and a complete set of SMA and fiber optic connectors to support the FPGA-based serial multi-gigabit transceivers (MGTs).

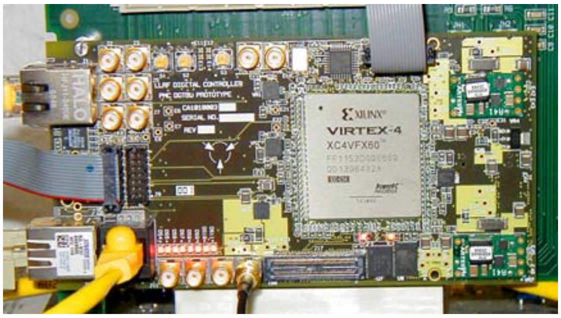

Figure 1: Custom design PMC module using ML310's 3.3V PCI slot to obtain power.

*Work performed under Contract Number DE-AC02-98CH10886 under the auspices of the US Department of Energy.

† severino@bnl.gov

The FPGA device is a feature rich Xilinx Virtex-4 FX60, with support for a long list of hard and soft IP cores. This includes, 2 PowerPC405 700+ DMIPS RISC 32-bit hard processor cores that can operate at up to 450MHz, 4 Ethernet media access controller (MAC) hard IP blocks, 16 MGTs, 128 XtremeDSP Slices (dedicated hardware multiply-accumulate units used for digital signal processing applications), and 4Mbit of on chip Block RAM [3].

EMBEDDED HARDWARE

Xilinx Embedded Development Kit (EDK) provides a Base System Builder (BSB) wizard that simplified the process of creating a base design for our board. The wizard only required us to specify the FPGA device we are using and external memories and I/O devices on our board. It generated a Xilinx Platform Studio (XPS) project that only required adding the FPGA pin location constraints in the generated user constraints file. The wizard also automatically created two standalone software applications (no OS required for running) for testing the memory and all the peripherals on the generated system's project.

We used Xilinx ML410 evaluation board online documentation and the experience acquired with a ML310 evaluation board to assign the proper user constraints to all the components. This process concluded with a full functional embedded hardware design that includes a PowerPC405 running at 300 MHz, a RS232 port running at 9600bit/s, one Ethernet port configured for 10/100 Mb/s, support for 256 MB of 64 bit wide DDR2 SDRAM running at 400 MT/s and access to the flash device.

A complete block diagram of the embedded design showing standard IP cores at the top of the diagram and custom ones at the button is shown in Figure 2.

Custom Modules

Since the BSB wizard only has support for standard Xilinx IP cores, custom developed cores had to be added manually to the XPS project. We added a total of four custom designed VHDL blocks to our embedded design.

The first three cores developed and inserted into our project are: a RHIC event link (REL) receiver, a real-time data link (RTDL) decoder, and a RHIC reset system link decoder. These cores are a FPGA based substitute for the standard RHIC V108 VME utility module. The V108 module provides RHIC FECs with synchronized ring equipment parameters, synchronous software event execution, and FEC remote reset capability via the VME bus [2, 4].

The RTDL link decoder connects to the OPB bus through a 32 bit wide, dual port on chip BRAM block as shown in Figure 2. The dual port block RAM allows the PowerPC to read parameters from one of the ports while at the same time the decoder logic writes to the other port without the possibility of data corruption.

The REL decoder connects to the OPB bus using a Xilinx OPB to IP interface (IPIF) controller. An HDL controller template was generated using the Create and Import Peripheral wizard. The template provides a user logic section for adding the body of the decoder logic. It also includes user logic interrupt support needed for the event decoder.

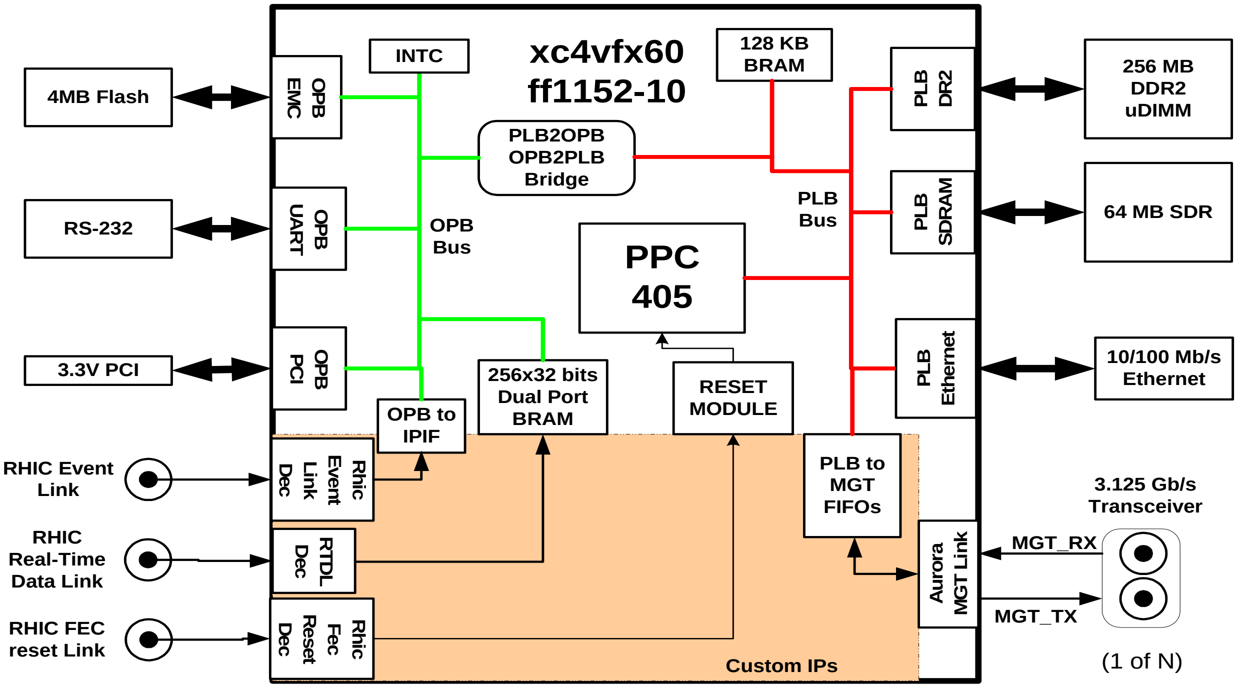

Figure 2: Block diagram of the embedded design.

The fourth core provides a high bandwidth connection between the PowerPC's PLB bus and an MGT Aurora link. Aurora is a point to point, low latency, low overhead, Xilinx protocol for high speed serial communication. The goal is to use an Aurora link to connect the FEC to other RF hardware elements such as ADCs, and DACs at a speed of 2.5 Gb/s per link. Due to the lack of ADC and DAC hardware at this time, we have only used the link to send data between microprocessors.

Such high speed links provide us with a tremendous capability in data throughput. As configured, we will have a peak transfer capability of 1.5GByte/s between the distributed daughter modules and the processor DDR2 memory. Such throughput vastly exceeds any capability we currently have in standard RHIC hardware, and as a result we envision much improved availability, quantity and resolution in the diagnostic data from our LLRF systems.

EMBEDDED SOFTWARE

The development software is capable of generating a Board Support Package (BSP) for a number of real time operating systems including VxWorks, LINUX and xilkernel. Since the RHIC control system runs mainly on VxWorks [1], we pay exclusive attention to this operating system.

Board Support Package

The BSP contains all the necessary support software for a system, including boot code, device drivers, and Real Time Operating System (RTOS) initialization code.

We selected the RS232 device as the standard I/O console for our board and proceeded to generate a VxWorks 5.5 BSP from within the XPS project. The resulting BSP required very minor modifications, since the XPS tool uses the hardware design to pre-configure most of the drivers. Manual configuration of the BSP involved changing the serial device baud rate, memory range of OS, and start address of the ROM device.

VxWorks Kernel and Bootrom

Once the BSP was properly configured, we used Tornado 2.2 IDE to build a kernel that contains most of the functionality of a RHIC FEC. It includes support for network services such as NFS and FTP required to connect with the RHIC control system software.

We also generated an uncompressed bootrom image of the kernel using the BSP. The bootrom is a scaled down VxWorks kernel whose primary job is to locate and boot a full VxWorks image from a remote file server. We copied this bootrom into the flash device which is used during system startup. This permits the system to perform system startup in the same way the standard RHIC style FEC does.

In order to get an idea of the performance of the Xilinx PPC405, we created a set of simple, qualitative benchmark functions that measured the speed of the system in four areas. We found that Xilinx PPC405 running at 300MHz performed better than a MVME2100 PPC603 that runs at 200MHz in two of the four tests. It performed better accessing the memory and doing integer computations. It performed worse when doing floating point computations and about the same on the network throughput test. The lack of performance when doing float operations comes from the fact that Xilinx processor does not have hardware support for floating point operations.

FUTURE DEVELOPMENTS

The most immediate development effort going forward involves transitioning from the Virtex-4 to the Virtex-5 FPGA. The Virtex-5 offers improved performance and features in a number of areas. These include higher clock speed, lower power dissipation, an improved processor core (PPC440) more block ram, and improved interfacing between the PLB and peripherals. The Virtex-5 is also offered with MGTs (now designated as GTPs) across all FPGAs in the family, not just the FX PPC hardcore series, allowing us to implement either a hard or soft processor core as appropriate, without losing the high speed serial connectivity. The Virtex-5 MGTs are also of an improved design, alleviating a number of practical implementation issues associated with the Virtex-4 MGTs.

Embedded firmware development continues using both the PMC prototype module and newer Xilinx ML50x series Virtex-5 evaluation platforms. In particular, we continue to expand and improve support for standard RHIC controls functionality – adding kernel support for the REL, and developing new HDL peripheral modules to replace the functionality of other existing RHIC controls VME hardware, such as the V202 delay generator.

CONCLUSION

We have developed an FPGA based embedded system that functions as a standard RHIC front-end computer. This controller has the potential for being used in a broad number of applications. It has all the necessary components of any commercially available embedded controller with the added power and flexibility provided by the FPGA resources.

REFERENCES

[1] L.T. Hoff, “Experience with FPGA-based processor as Front-end Computer, ICALEPCS'05, Lucerne, Oct 2005, http://www.jacow.org.

[2] D.S. Barton et al, “RHIC Control System for the Brookhaven Relativistic Heavy Ion Collider”, RHIC (EPAC 1998, Stockholm, Sweden)

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

[4] H. Hartmann, “The RHIC real time data link system, Proceedings of the 1997 IEEE Particle Accelerator Conference”, Vancouver, p. 2499.