A MODULAR ARCHITECTURE FOR ACCELERATOR INSTRUMENTATION*

J. DeLong,# Brookhaven National Laboratory, Upton, NY, U.S.A.

Abstract

With accelerated schedules and finite resources, the development of a common open source platform for accelerator instrumentation is required. This effort has led to the development of a flexible architecture with clearly defined interfaces. The resulting platform is currently used to implement fast orbit feedback as well as the Beam Position monitors for NSLS-II. The design includes an embedded processor, digital signal processing resources and communications interfaces to controls, the timing system and other devices distributed throughout the accelerator complex. This new architecture promotes customization and design re-use and is presented as an Open Source Hardware development project.

ARCHITECTURE

Centered on a Xilinx Virtex 6 field programmable gate array (FPGA), the Digital Front End (DFE) electronics for the NSLS-II Cell Controller and Beam Position Monitors [1] was designed to provide a cutting edge platform for attaching a variety of back end processing modules. The interfaces are designed to be flexible and provide the necessary communication, data storage and math processing blocks used in many accelerator designs. This architecture (Figure 1) lends itself to the fast development of devices such as low level RF, beam instruments, and communications hubs.

FPGA RESOURCES

The current design uses a Virtex 6 LX240T FPGA in an 1156 pin ball grid array package. This FPGA offers 720 user configurable IO pins (360 differential pairs), 24 Low power GTX transceivers [2] (6 connected to SFP modules), 14Mbits of Ram, 4 Ethernet Media Access controllers (1 connected to a Micrel PHY) and 768 DSP48E1 [3] slices.

The DFE uses the soft core Micro Blaze processor and Xilinx Kernel to control on-board peripherals as well as exchange data with the control system via a hard core Ethernet MAC and external PHY. The PCB has a 256MB DDR-3 Dynamic Ram module for program and data memory and is accessible from the Micro Blaze processor and the FPGA fabric.

The GTX transceivers are high speed (6.25 Gb/s) serializer / deserializers (serdes) with the necessary synchronization hardware for connecting multiple boards. These transceivers are used to implement a reflective memory among the 31 cell controllers in the NSLS-II. The transceivers provide a simple 32-bit interface to the FPGA fabric. The design of the reflective memory is discussed in detail.

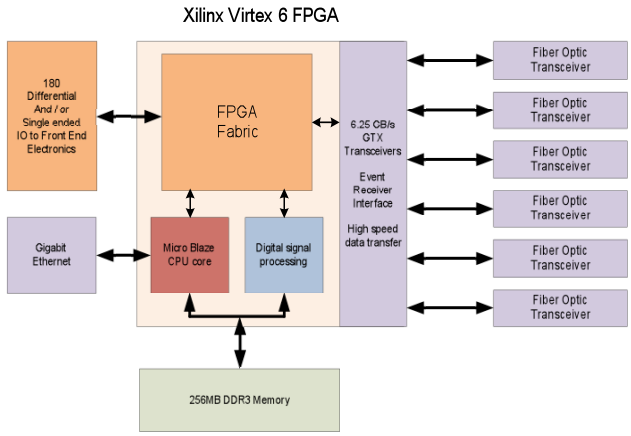

Figure 1: DFE block diagram.

INTERFACES

Timing

The timing system for the NSLS-II is based on the commercial off the shelf products provided by Micro Research Finland [4]. Timing is sourced by an event generator, which locks to the RF reference clock and delivers beam synchronous events at ¼ of the RF frequency (~125 MHz). Events are transmitted around the machine over fiber optic cables and received in hardware designed to produce triggers with ~8 ns resolution and timing jitter of less than 15 ps RMS.

The DFE board uses one of the GTX transceivers to receive and decode events locally. Once decoded triggers are generated and distributed to the fabric of the FPGA. All machine synchronous timing on the DFE utilizes this embedded event receiver. The transmitter logic of two GTX transceivers is used to provide enhanced resolution and jitter performance.

The timing system clock is recovered and cleaned in a phase lock loop then applied to the GTX transmit clock. The serial bit stream from the transmitter is then manipulated to produce arbitrary trigger patterns with a resolution of 200 ps and jitter of less than 5 ps RMS with respect to the RF reference clock.

* Work performed under the auspices of the U.S. Department of Energy.

jdelong@bnl.gov

High Speed Data Communications

Data from all BPMs in the NSLS-II must be transferred to the 31 cell controllers, where the Fast Orbit Feedback algorithm is executed. This must be accomplished in less than 25 us. A high speed serial communication link has been designed to implement this.

The topology of the link is shown in Figure 2. Local to each cell controller is a group of BPMs producing data at the 10 kHz rate. Data from each BPM is transferred to each adjacent node in a loop. Since the same data is circulated in both directions the link is redundant and when fully functional data transfer occurs in half the time a single loop requires. As the local data is received at the cell controller it is passed to the adjacent nodes on the cell controller loop. After local data is transferred data packets from the remote nodes are transferred until all data is reflected in all nodes.

The data link uses the Xilinx GTX transceivers operating at 6.25 Gb/s. During bench testing data transfers between cell controllers take approximately 400ns per node after an initial 600ns for local data transfer to complete. Each node produced 8 packets with two data words (simulating vertical and horizontal BPM data). The cell controllers under test were separated by 30 meters of fiber optic cable which approximates the final installation. The fiber contributes approximately 150ns of delay to the system. Extrapolating to 31 Cell controllers the total transit time is less than 10us for all BPM data to be transferred to all nodes with two operational links and less than 20us if one link is interrupted. The interface to the data producing device (BPM) is a 32 bit parallel data stream with a data valid strobe. Prior to transmission the packet length and target address are known and stored as parameters. Data from all nodes on the link is stored in dual port memory as it is accumulated. When all data is received a strobe triggers the calculation unit to read the new data package.

Back End Processing Interface

Five Tyco Z-PACK HM-Zd connectors are available with 180 connections to the FPGA for Back end processing boards designed to interface with the DFE. These connections can be differential or single ended as required. Two differential pairs are connected to the transmit side of GTX transceivers. This allows for using the high speed low jitter CML outputs of the GTX to drive clocks or timing on the back end.

Micro Blaze Processor and Control System Interface

The DFE uses the Xilinx MicroBlaze processor core running at 100MHz and runs the Xilinx Micro Kernel. A multi-port memory controller is used to interface the DDR3 memory to the processor, Gigabit Ethernet DMA channel, and data streams originating in the FPGA fabric. The memory interface runs at 400MHz allowing for data multiplexing between the different data sources.
The DFE uses the Xilinx Light Weight Internet Protocol (LWIP) stack for Ethernet communications. Performance measurements show a data transfer rate of approximately 4MB/s with the LWIP stack.

The BPM has been integrated with the control system using EPICS channel access. Transfer of large data blocks consisting of one million points of BPM data and raw ADC samples has been tested and proven robust both at BNL and at Lawrence Berkeley National Lab.

SUMMARY

The DFE is designed to provide the required interfaces and processing power to support many accelerator devices. This design is currently used in the NSLS-II Beam Position Monitors and Cell Controller where it provides high speed communications, digital signal processing and the control system interface. This design is available as Open Source Hardware for further development at BNL as well as other institutions where this design may help reduce hardware development cycle time [5].

ACKNOWLEGEMENTS

This board was designed by Al Della Penna, Kiman Ha, Joe Mead, Yuke Tian, Kurt Vetter, and others at the Brookhaven National Lab.

REFERENCES

[1] K. Vetter et al., “NSLS-II RF Beam Position Monitors”, these Proceedings: PAC’11, New York, March 2011, paper MOP211; to be published on jacow.org.
[2] Xilinx Virtex-6 FPGA GTX Transceivers, User Guide, UG366 (v2.5) January 17, 2011
[3] Xilinx Virtex-6 FPGA DSP48E1 Slice, User Guide, UG369 (v1.3) February 14, 2011, www.xilinx.com
[4] Micro Research Finland, www.mrf.fi
[5] J. DeLong, “Moving Toward Open Source Hardware Development”, PAC’11 New York, March 2011, MOP195; to be published on jacow.org.

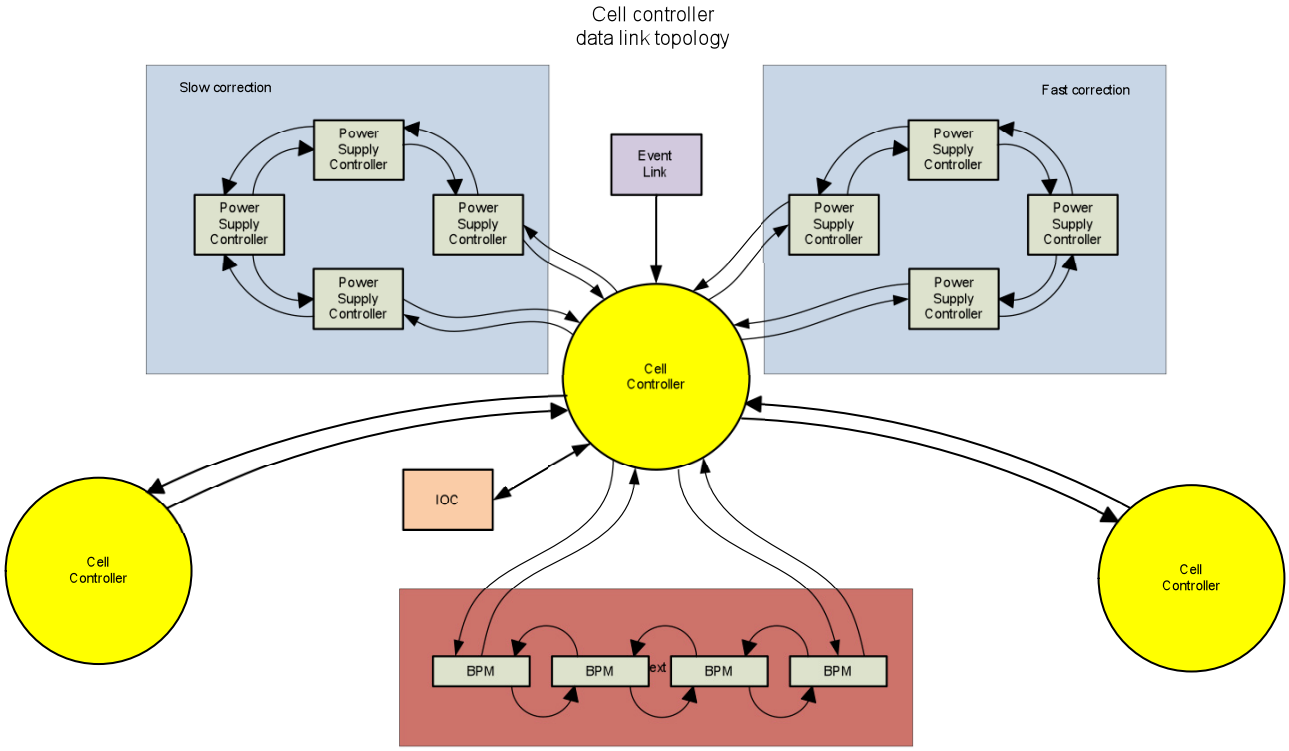

Figure 2: Cell controller communications topology.