slac-pub-12053

An Integrated Front-End Readout and Feature Extraction System for the BABAR Drift Chamber

Jinlong Zhang, *Member, IEEE*

Abstract— **The BABAR experiment has been operating at SLAC's PEP-II asymmetric *B*-Factory since 1999. The accelerator has achieved more than three times its original design luminosity of $3 \times 10^{33}\mathrm{cm}^{-2}\mathrm{s}^{-1}$, with plans for an additional factor of three in the next two years. To meet the experiment's performance requirements in the face of significantly higher trigger and background rates, the drift chamber's front-end readout system has been redesigned around the Xilinx Spartan 3 FPGA. The new system implements analysis and feature-extraction of digitized waveforms in the front-end, reducing the data bandwidth required by a factor of four.**

Index Terms— **BABAR, Drift Chamber, FEE, FPGA.**

I. BABAR Detector

THE BABAR experiment was designed to study CP-violation in *B*-meson decays with asymmetric e^+e^- beams. The detector consists of six main subsystems: the Silicon Vertex Tracker (SVT), which has been designed to measure angles and positions of charged particles just outside the beam pipe; the Drift Chamber (DCH), which provides the momentum measurement for charged particles and also supplies the measurement of dE/dx for particle identification; the Detector of Internally Reflected Cherenkov light (DIRC), which is a device that provides information on particle type and hence separation of pions and kaons in the momentum range from about 500 MeV/c to 4.5 GeV/c; the Electromagnetic Calorimeter (EMC), which is designed to detect electromagnetic showers with excellent energy and angular resolution over the energy range from 20 MeV to 4 GeV; the superconducting solenoid, which provides 1.5 T magnetic field; and the Instrumented Flux Return (IFR), which is designed to identify muons and neutral hadrons. The complete detector is described in detail elsewhere [1].

The DCH, a multi-wire proportional chamber, consists of a 280 cm long cylinder with aluminum end-plates, whose inner and outer radii are 23.6 cm and 80.9 cm, respectively. The wires, including tungsten-rhenium sense wires, low-mass aluminum field wires, guard wires and clearing wires, are strung in the holes on the endplates between the inner wall and the outer shell. All wires are organized into 40 layers of small hexagonal cells, total 7104, subdivided into 10 superlayers in an alternating axial and stereo pattern. The DCH gas is an 80:20 mixture of helium:isobutane and the operating voltage is ~2000 V.

II. DCH Readout System

The front-end electronics (FEE) system of the DCH measures the drift time of ionization onto the sense wire and the total charge collected, and also sends a signal for every hit wire to the trigger system. The whole system is organized into 16 wedges azimuthally. The major components of one wedge are shown in Fig. 1.

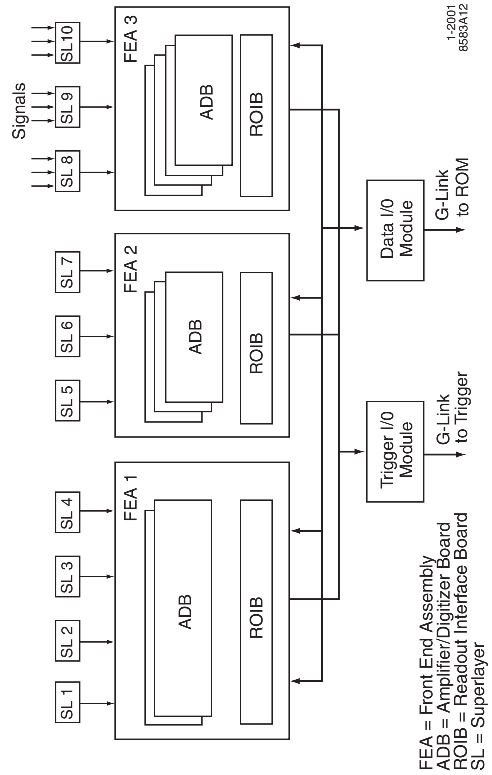

Fig. 1. One wedge of the DCH electronics showing three Front End Assemblies and the I/O modules.

The HV Assembly distributes high voltage to the sense wires and routes the sense wire signals up to the Front End Assemblies (FEAs).

The FEAs are constructed as wedge-shaped aluminum boxes. One FEA holds a few Amplifier/Digitizer boards (ADB) and one Readout Interface Board (ROIB). The number of ADBs in one FEA depends on the radial position of the FEA in a wedge as Fig. 1 indicates.

The ADB receives the signals from the sense wires through a connector on the HV assembly. Its basic building block in-

Manuscript received November 11, 2005.

For BABAR Collaboration. J. Zhang is with the University of Colorado at Boulder.

Submitted to IEEE Nucl.Sci.Symp.Conf.Rec.

Work supported in part by Department of Energy contract DE-AC02-76SF00515
Stanford Linear Accelerator Center, Stanford, CA 94025

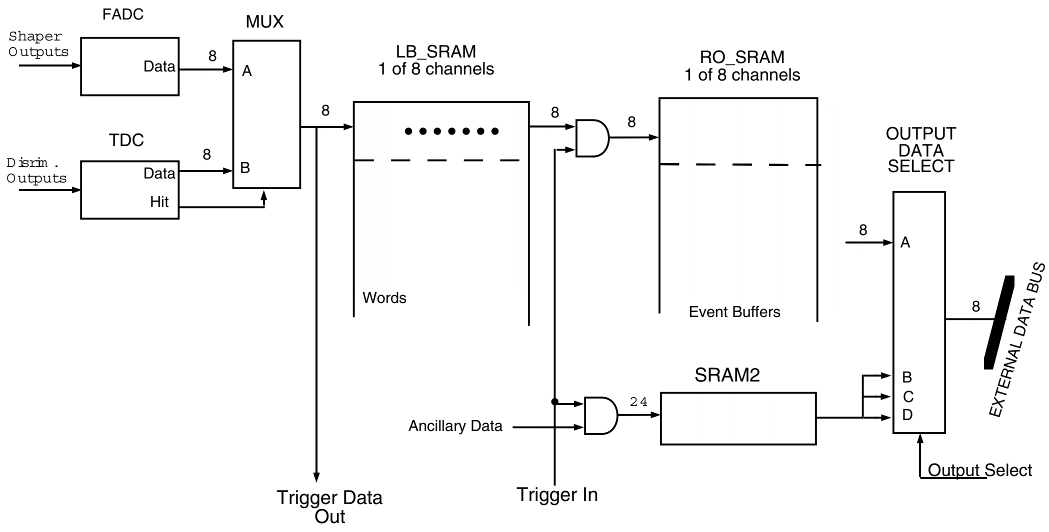

Fig. 2. The block diagram of the ELEFANT chip showing the major functional components.

cludes two amplifier integrated circuits (DCAC) containing four channels each providing input to a single 8-channel digitizer IC (ELEFANT chip). The total number of channels per ADB board again depends on the radial position of the FEA which holds the ADB. A block digram of the ELEFANT chip is shown in Fig. 2. The details of its functionality are described in the next section.

The ROIB has four major functional components. The command decode logic receives commands from the Fast Control system, decodes them and performs the appropriate operation. Commands include the standard BABAR data taking commands as well DCH-specific commands that configure the DCAC or the ELEFANT chip or control calibration. The ELEFANT readout controllers (slave controller) read the data from the ELEFANT chips into FIFOs on the ROIB. The master readout controller coordinates the readout of FIFO data onto the 2-bit 30 MHz bus to the Data I/O Module (DIOM). The trigger interface multiplexes the serialized trigger output data from ELEFANT chips onto 1-bit 60 MHz data links to the Trigger I/O module (TIOM).

The DIOM converts 16 sets of 60 MHz serial links from the FEAs to 1 Giga-bit fiber links that carry the data to the BABAR Readout Module (ROM). It also receives the Fast Control commands from the ROM via the fiber link and fans those signals out to 12 FEAs. Four DIOMs, each driving one data link and receiving one control link, are sufficient for the expected rates in the DCH at design luminosity of the accelerator.

Each TIOM multiplexes the trigger data from six FEAs and transfers it to the BABAR trigger system. There are unused channels that are deployed to send timing signals to the trigger system to monitor synchronization.

The ROM receives the formatted and digitized signal amplitude (FADC) and time information (TDC) from the on-detector electronics, then performs feature extraction (FEX) in software. The ROM also receives fast control clocks and commands from the Fast Control system and sends them down to the on-detector electronics.

III. Data Flow

During BABAR data taking, the ELEFANT chip continuously digitizes analog waveforms from eight channels at 15 MHz into 32 bytes data per channel per trigger signal. If the waveform crosses a discriminator threshold in one of the 32 samples, the FADC hit is replaced by a TDC hit. The data is stored into a circular latency buffer (LB_SRAM) where up to 12 μs of data can be stored. If the trigger system decides that the event should be read out, a trigger signal is sent down to the FEE and it causes 32 byte data to be transferred from the LB_SRAM to one of a set of four readout buffers (RO_SRAM). While the transfer is proceeding the ELEFANT chip marks whether each wire was hit or not, using some settings set by the ROIB. The trigger module in the ELEFANT chip derives the trigger primitives from the FADC or the TDC and sends out one byte trigger data which indicates hits of each of the eight channels.

Some time after the trigger signal, an event read signal is sent down and it causes the slave controllers on the ROIB to transfer the data from all RO_SRAMs to the FIFOs on the ROIB. One slave controller is responsible for all ELEFANT chips in one ADB and the data from those ELEFANT chips is transfered into one FIFO on the ROIB serially via the 8-bit 7.5 MHz bus.

About 15 μs after the event read signal, the DIOM sends down another signal, causing the master controller on the ROIB to transfer the data from the FIFOs to the DIOM, and then up to the ROM. The data from all FIFOs on one ROIB are serialized on the 8-bit 7.5 MHz bus before multiplexed onto 2-bit 30 MHz bus to the DIOM. The data is multiplexed further onto 1-bit 60 MHz bus to the ROM.

IV. System Bottleneck and Upgrade solutions

The PEP-II luminosity has been increasing successfully in the last few years. With the increasing trigger rate and background rate, a readout bottleneck arises in the DCH readout system. It routinely limited data taking to 3 to 3.5 kHz trigger rate. The projection performed in 2003 for dead time as a function of accelerator luminosity is shown in Fig. 3. The dead time contribution in 2003 was ∼1% and it would have reached 30% if the system had remained unchanged. The bottleneck arises because there are too many hits in the DCH, and the time to read out 32 bytes per hit is too long.

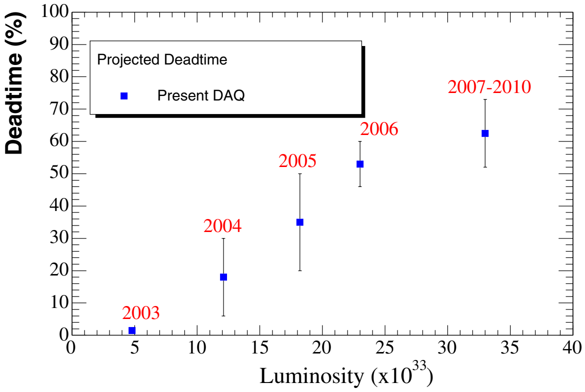

Fig. 3. Projected DCH dead time versus accelerator luminosity.

To lower the readout time either the data must be read out faster, or the data size must be reduced. One solution is to decimate the waveform by sending out only half of the waveform, from 32 bytes down to 16 bytes, so called half sampling. This decimation could be done by modifying the slave controller firmware on the existing ROIB boards. The difficulty of this upgrade is to increase the ELEFANT chip readout rate to 15 MHz, which had been 7.5 MHz, to avoid changes in other functional components. Another solution is to run FEX algorithm earlier, in firmware on ROIBs instead of in software on ROMs. This upgrade requires a new ROIB with a larger FPGA and a Xilinx Spartan 3 series FPGA, XC3S1500, is chosen. In addition to implementing the FEX algorithm for the new FPGA with VDHL, all the functionalities of the master controller, the slave controller, and the trigger interface, which are performed by separate Xilinx 3000 series FPGAs, also need be reimplemented with VHDL and integrated into the new FPGA. All I/Os on the ROIB remain the same so no other boards need to be changed.

V. Half Sampling

The goal of half sampling is to obtain a factor of two in the data flow speed without significantly impacting physics results. In all 32 samples, for a pair, the algorithm keeps TDC hit (if presents) or second half of pair of FADCs. As a first upgrade, the algorithm has been implemented in the slave controller firmware and the EPROMs on the ROIBs were changed in the summer of 2004. The track-by-track comparisons of the charged track information and the overall yield comparisons of physics events show no significant effect on the quality of the data. This upgrade yields a factor of two as expected.

VI. Feature Extraction Implementation

Before the second upgrade, the FEX software ran on the 300 MHz PowerPC processors in the ROMs. The algorithm extracts a list of TDC hits from the 32 sample waveform and identifies one of them as the leading edge. The total charge is determined from the integral of FADC samples following the leading edge and FADC information is inferred for samples with TDC hits by interpolation. A correction for saturated FADC samples and measured pedestal drift is also applied to the charge calculation before a final gain correction factor. The process converts a 32 byte waveform to a two byte status word, a two byte charge, and a list of two byte TDC hits. This leads to a 75% data size reduction on average. By moving the FEX algorithm from ROMs into larger XC3S1500 FPGAs on the new ROIBs, a factor of four will be obtained in the data flow speed compared to the original system.

When the algorithm is implemented in the FPGA, further data reduction is achieved by eliminating channels with no TDC hits, typically 15% of channels read. An additional factor of 25% in size reduction can be obtained by applying the Huffman encoding [2] or other encoding algorithms to the status word, charge, and TDC hits.

A block digram (Fig. 4) shows the FEX process in the FPGA which reads data from three ADBs. The FEX algorithm requires channel dependent constants as well as global correction factors. These numbers will be transfered down to the buffer space in the FPGA before the FEX engines start. Because of the size reduction, it could be insufficient for the three 15 MHz ADB readouts to supply data to the effectively faster 2-bit 30 MHz output stream, and an underrun condition is possible. In order to make best utilization of the output bandwidth, the data multiplexer may switch between ADBs after completing any ELFANT chip's output, but in the case when none is available, the output will be padded. The padding is in a distinguishable pattern from the meaningful data. To reduce the underrun occurrences the ADB readout controllers can be started sooner and their output can be buffered. It's impossible for the old implementation because the old ROIB buffer (FIFOs) has a depth of only one event and all slave controllers must wait for the completion of the slowest associated with the same ROM before proceeding. Having ROIB buffer (block RAMs in the FPGA) depth equal to that of the ELEFANTs allows the ADB readout controller to read the next unread triggered event promptly after the FEX output of the previous triggered event gets buffered but not necessarily read by the data multiplexer yet.

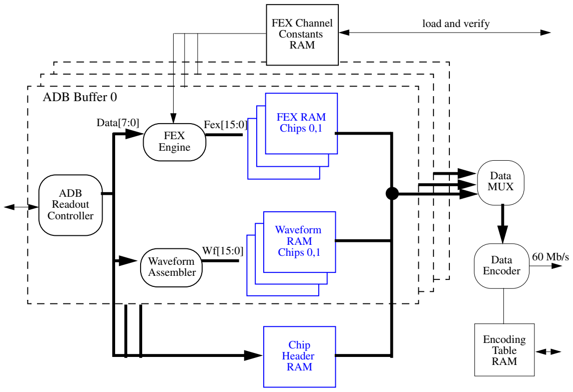

Fig. 4. The FEX implementation in the FPGA.

VII. Image Switch and Firmware Upload

The larger XC3S1500 FPGA on the new ROIB allows new features to be added. It is useful to implement a mechanism to upload a new image to the FPGA from the ROM without shutting down the detector and accessing the electronics hardware directly. In case of errors during uploading the system must be able to revert to a known stable state so the upload process can be retried. To support this the new ROIB contains two flash PROMs (Fig. 5). The primary PROM, PROM 0, holds a stable image. The second PROM, PROM 1, can be programmed through the JTAG interface. The JTAG interface of PROM 1 are connected to both a JTAG header and the user I/O pins of the FPGA. This allows a block of logic in the FPGA to serve as a JTAG programmer.

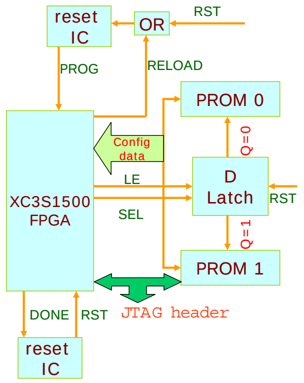

Fig. 5. The design to perform image switch and firmware upload.

The program signal PROG of the FPGA is connected to an IC reset device which can hold the PROG signal in an active state for at least 100 ms when the PROG signal is asserted. It will be asserted when the reset signal (RST) is asserted or when the FPGA asserts its reload (RELOAD) signal. So with a power recycle or a RST signal, the FPGA will be forced into the program state. A D-latch is used to control which of the two PROMs is selected when programming the FPGA. The latch enable signal (LE) is asserted by either the FPGA or the RST signal. The latch input (SEL) is pulled low on the board. The RST signal forces the D-latch to latch the low SEL signal then select PROM 0. After the RST is de-asserted, the de-assertion of the PROG signal allows the FPGA to start loading the image from the selected PROM 0. Once the image is stable a command can be issued from the ROM to assert the SEL signal high following by a high assertion of the LE signal. Thus PROM 1 will be selected. Another command from the the ROM can cause the FPGA to assert the RELOAD signal. As a result the PROG signal will be asserted and the FPGA will reload the image from PROM 1.

The VHDL based firmware are compiled into the configuration stream in Serial Vector Format (SVF) [3], which is a standardized format to interact with the JTAG interface. A C++ program converts the SVF file into chunks in the format of BABAR DAQ communication protocol. The chunks are transfered down to the block RAMs in the FPGA. A "PROM-Programmer" controller has been developed with VHDL and integrated into the firmware. It runs in the FPGA and controls the JTAG interface to program the PROM when chunks are received.

The C++ program reads the SVF commands and their payload in the SVF file, then encodes into binary chunks which can be sent down from ROMs as DAQ commands. Each chunk includes an integer number of SVF commands and the chunk size is determined by the size of memory available in the FPGA and the maximum size allowed of a DAQ command. In the SVF file, the basic SVF commands, SIR, SDR, STATE and RUNTEST, are converted while ENDIR, ENDDR commands are integrated along with the end states of the SIR, SDR commands. Other commands used to program the chained PROMs are discarded.

All JTAG operations are performed through the Test Access Port (TAP) of PROM 1. The four signals of the TAP, TMS, TDI, TDO and TCK, are received on the JTAG pins from I/O pins of the FPGA. The "PROM-Programmer" controller, which is 16-state finite state machine [4], generates these signals based on the SVF commands in the chunks. The TMS signal controls state transitions. Instructions and data are shifted into PROM 1 on the TDI pin and are shifted out on the TDO pin. All state transitions and activity on the TDI and TDO signals are synchronous to the clock signal TCK. The ROM keeps polling a status register to check if the state machine has finished the programming of the current chunk successfully, and decides if to send down the next chunk or to restart.

VIII. Radiation Damage and Configuration Check

The Spartan 3 XC3S1500 chip is an SRAM based FPGA. Single event upsets (SEU) can corrupt either the configuration SRAM, or a programmable logic element that normally changes during running. The FPGA configuration corruption is the main source of concern because it can potentially cause all data to be corrupted until the FPGA is reconfigured. Furthermore, it causes additional dead time to reconfigure the system.

At the location of the DCH electronics, the low energy neutrons are the dominant source of SEUs. Given that the low energy neutrons are emitted isotropically if there is a point source, the intensity scales as $1/r^2$. With the counting rates from several neutron detectors, the position of the point source was estimated. Subsequently a neutron rate of $\sim 2\ \mathrm{kHz/cm^2}$ at the location of the DCH electronics was predicted. The SEU rate was estimated with the predicted neutron rate and the published SEU cross section [5].

A configuration check mechanism has been developed and integrated into the firmware. It runs continually and takes two seconds to check all configuration data. SEUs are detected by bit for bit comparison and CRC verification between the configuration data read from FPGA configuration memory via JTAG interface driven by FPGA user I/O signals, and the original configuration data read from PROMs via the serial data line used as a user I/O after configuration. A register is used to count the occurrences and it can be read back. A status bit is inserted into the event data stream.

SEUs are observed at about the predicted rate (20-80 per chip in 80 days) on the three test chips installed on the detector. Not all SEUs matter and only three SEUs on this three chips in six months clearly affected data taking, projecting a total of 48 in the whole DCH electronics system for a same time duration. Double Module Redundancy (DMR) is being implemented to determine the serious problems. Triple Module Redundancy (TMR) and the Scrubbing mechanism are under investigating but both require a different FPGA.

IX. Conclusion

After the upgrade the DCH readout system is more capable, flexible and robust. The factor of four in the data flow speed will accommodate the incoming increasing luminosity. The new system has been successfully installed and the BABAR experiment will certainly produce more physics results with this system.

References

[1] BABAR Collaboration, B. Aubert *et al.*, Nucl. Instr. and Methods **A479**, 1 (2002).

[2] D.A. Huffman, *A method for the construction of minimum-redundancy codes*, Proceedings of the I.R.E., Sept 1952.

[3] http://www.asset-intertech.com/support/svf.pdf

[4] http://www.xilinx.com/bvdocs/appnotes/xapp139.pdf

[5] For example, http://www.xilinx.com/esp/mil_aero/collateral/RadiationEffects /NSEU_response.pdf