The BABAR Data Acquisition System

R.T. Hamilton[1], R. Claus[2], P. Grosso[3], M.E. Huffer[2], C. O'Grady[2], I. Scott[4], J.J. Russell[2]

[1]The University of Iowa, Department of Physics and Astronomy, Iowa City, IA 52242

[2]Stanford Linear Accelerator Center, 2575 Sand Hill Rd., Menlo Park, CA 94025

[3]INFN Torino, Italy

[4]Royal Holloway & Bedford New College, University of London, Egham, Surrey, TW20 0EX, UK

Abstract

The BABAR experiment at the Stanford Linear Accelerator Center is designed to perform a search for CP violation by analyzing the decays of a very large sample of B and $\overline{B}$ mesons produced at the high luminosity PEP-II accelerator [1]. The data acquisition system must cope with a sustained high event rate, while supporting real time feature extraction and data compression with minimal dead time.

The BABAR data acquisition system is based around a common VME interface to the electronics read-out of the separate detector subsystems. Data from the front end electronics is read into commercial VME processors via a custom "Personality Card" and PCI interface. The commercial CPUs run the Tornado operating system to provide a platform for detector subsystem code to perform the necessary data processing. The data is read out via a non-blocking network switch to a farm of commercial UNIX processors.

The current implementation of the BABAR data acquisition system has been shown to sustain a Level 1 trigger rate of 1.3 kHz at an event size of 25 kbytes and with negligible deadtime. Upgrades currently in development will permit the system to support the design Level 1 rate of 2 kHz with negligible deadtime.

I. OVERVIEW

The BABAR data acquisition system is an intimate partnership between both the hardware and software systems. This partnership may be viewed as either a software system *exporting* the functionality of the hardware system, or the hardware system augmenting or *leveraging* the functionality of the software system. If the latter view is taken, the hardware system may be considered the *platform* on which the software system resides. Within this document, this view prevails. The hardware system is called the *DataFlow Platform* (DFP) and the software system is called *DataFlow.* Together they form a *Data Acquisition Platform* (DAP). In turn, DAP augments other systems; for example, Online Event Processing (OEP) or Calibration to form a complete *Data Acquisition Systems* (DAS).

A system of this form has the attractive feature or capability of being replicated in whole or part. This allows subsystems the luxury of developing, testing, and/or calibrating their own hardware/software systems in parallel. There is therefore potentially more than one DAS in existence at any point in time.

The discussion here will form an overview of the DAS, with emphasis on the DAP. Details on the challenges found and overcome in the development of the BABAR DAS can be found in [2], [3], [4].

The BABAR DAP is responsible for reliable transportation of data from the detector to the Level 3 trigger farm with minimal losses. In order to achieve this it must be able to sustain data transfer rates which are consistent with the design rate of 2.0 kHz for Level 1 triggers.

The input data volume from the detector is 1.2 Mbytes/event. This results in a data rate of 2.35 Gbytes/sec into the collision hall DAP. After processing in the DAP, this results in 65 Mbytes/sec (33 Kbytes/event) being transported from the DAP to the Level 3 trigger farm.

The BABAR DAP must also act as a vehicle for the reduction and/or compression of the data from the front ends, for the calibration of all the detector subsystems, and for the development, commissioning, and diagnosis and repair of the subsystem detector electronics.

II. THE FEE MODEL

The DAP treats all of the on detector electronics in a uniform manner using the Front End Element (FEE) model illustrated in Figure 1. Within this model, analog data is generated by the detector elements. This data stream is continuously digitally sampled and stored in a circular buffer. Upon receipt of a Level 1 trigger, data is transferred from this buffer and placed in an event queue. When the DAP issues a read event command, the oldest event in the queue is transferred to the DAP. Communication to and from the FEEs is achieved through G-links implemented over fiber optic lines. The communicated data structures follow an agreed upon protocol. The implementation of this model is subdetector dependent.

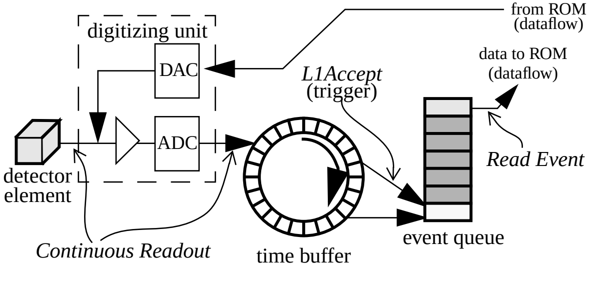

Figure 1: The FEE Model

III. THE DATAFLOW PLATFORM

Any instance of a DataFlow Platform (DFP) comprises a networked set of components. A platform's components are derived from a collection of embedded processors, workstations, crates, and modules. DataFlow and its applications execute under the supervision of a real-time operating system (VxWorks [6]) on the embedded processors and under the supervision of UNIX on the workstations.

There are a variety of different VME modules within a DFP. Due to the custom J3 backplane, these are constrained to a particular kind of crate and slot number. The different flavors of module are:

Fast Control Timing Module (*FCTM*) Phases BABAR timing to PEP-II timing. Resides in a Master Crate.

Fast Control Gate Module (*FCGM*) Receives, distributes, and scales Trigger System signals, auxiliary trigger signals, and common calibration start signals. Resides in a Master Crate.

Fast Control Partition Master (*FCPM*) Uses input from the Trigger System and the FCDMs to synthesize *L1Accepts*. Resides in either a Master Crate or a Slave Crate.

Fast Control Partition Router (*FCPR*) Fans-out Fast Control and Timing System (FCTS) command packets and *sysclk* to and fans-in *full/available* from FCDMs residing in Slave Crates. Resides in either a Master Crate or a Slave Crate.

Fast Control Distribution Module (*FCDM*) Resides in a DAQ Slave Crate, receiving FCTS command packets and *sysclk*; redistributing them to its ROMs. In addition, it gathers the *full/available* lines from its ROMs and exports the summed value back to the FCTS.

Read-Out Module (*ROM*) Provides a platform from which applications may control FEEs and access event data for the purpose of reduction, transformation, or monitoring. In addition, redistributes the FCTS command packets and *sysclk* to its FEEs.

The modules are assembled into crates. With the platform these come in two varieties:

DAQ Slave Crate Houses a set of up to seventeen Read-Out Modules (*ROMs*) and one Fast Control Distribution Module (*FCDM*). Support for the ROMs and FCDM normally requires additional Fast Control and Timing System (*FCTS*) modules. These additional modules can be contained either in a Slave Crate or in a Master Crate (as described below). The ability of a Slave Crate to house these supplemental FCTS modules allows a Platform to be assembled without the additional cost of a Master Crate.

DAQ Master Crate Houses a set of FCTS modules that allows the simultaneous control of up to 32 Slave Crates as a single Platform. In addition, the crate contains an FCTM which interfaces BABAR to the accelerator. A Platform may have at most one Master Crate. This type of crate is only of interest as part of the platform in the collision hall.

Communication within the platform is via one of three methods:

Dedicated Serial Line The FCTS multicasts control messages from the master crate to all slave crates using dedicated serial lines.

VME Event data from ROMs within a crate are forwarded across the VME backplane to the *Slot 1* ROM. The *Slot 1* ROM then performs a partial event build for the data from its crate before forwarding the results to the UNIX processor farm.

Ethernet Communication from the crates to the processor farm is achieved over ethernet using a 100 base-T connection.

A. Partitioning

It is important to note that the Fast Control and Timing system (FCTS) is capable of sending control messages to a subset of the crates within a platform. This is the enabling technology which allows the platform to be partitioned. This is a mechanism which allows the platform to be shared in such a way as to maintain the fiction of multiple systems operating simultaneously within a single platform. This ability enables the optimal use of the platform during commissioning and calibration.

IV. THE READ-OUT MODULE

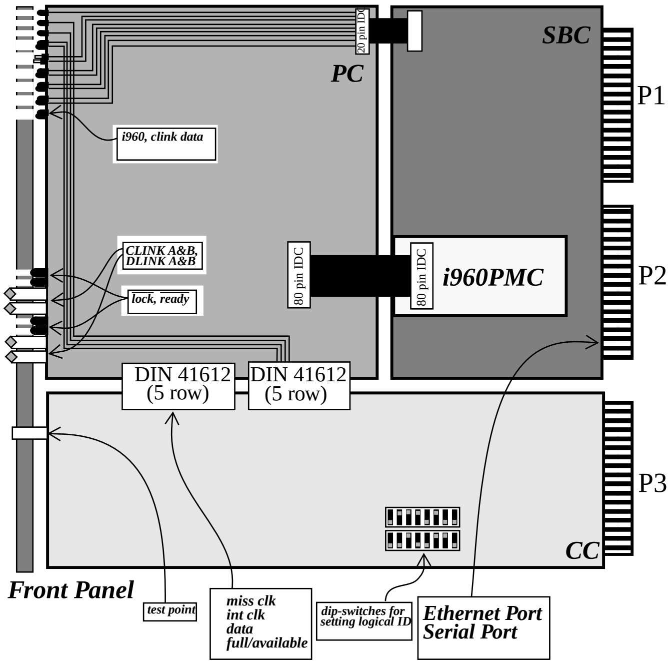

Figure 2: Schematic view of a ROM

In concert with the Fast Control and Timing System (FCTS), the Read-Out-Module (ROM) gathers event oriented data from its managed Front-End-Electronics (FEEs). Once the data have been gathered into the ROM, a general purpose processor working under the supervision of VxWorks [6] is available to extract, transform, reduce and/or monitor this data. This operation is called feature extraction. The results of feature extraction are forwarded out of the module (via its VME interface) and contribute to the overall content of a DataFlow event.

The ROM views its electronics in logical units called elements (FEEs). Each ROM is capable of managing up to thirty-two elements. Typically, an element represents a collection of detector channels, where the channel count per element and the channel definition vary as a function of subsystem.

A schematic view of a ROM is shown in Figure 2.

A. ROM Components

The ROM is a composite of four distinct boards assembled into a single 9U VME module. The purpose of these components is detailed below.

Single Board Computer (SBC) The DataFlow processor is a stripped down version of a VME standard, commercially available Single Board Computer, the Motorola MVME2306 Power PC Module[5]. This utilizes a 300 Mhz processor with 32 Mbytes of RAM. This board is responsible for performing the subsystem specific feature extraction and for controlling communication with the rest of the platform, either over the VME backplane or on ethernet as previously discussed.

i960RP based PMC Card (i960PMC) The i960PMC contains an i960RP integrated circuit, one mega-byte of local memory, and dedicated microcode. From an architectural point of view, the i960PMC bridges from the PCI Bus on the SBC to the i960 bus on the CC and PC. However, from the perspective of DataFlow, the i960PMC's principal function is to manage both the movement of event data from the PC's intermediate store to the SBC, and the movement of FCTS command packets from the CC to the SBC.

Personality Card (PC) Using a BABAR standard communication protocol, the PC manages a group of up to 32 Front-End-Elements (FEE). It reads data from the Front-Ends' event queues into an intermediate store. The CC initiates the read-out and forwards the result to the i960RP. Fluctuations in the movement of data from the Front-Ends to the SBC are smoothed out by the intermediate store. The intermediate store is partitioned into a set of fixed size buffers whose size and number are determined by the flavor of personality card.

The PC comes in two flavors: Triggered (TPC) and Untriggered (UPC). The TPC moves data from the FEEs to the intermediate store upon receipt of a level 1 trigger. The UPC is used to read out the electromagnetic calorimeter and takes a continuous stream of input data from the FEEs, in effect implementing the FEEs event queue within the PC.

Controller Card (CC) The CC provides an interface to the FCTS through which it receives *sysclk* and command packets. The CC extracts and then forwards a subset of each packet to its FEEs. In addition, the entire packet is sent to DataFlow through the i960PMC.

The Controller Card, on the receipt of a Level 1 trigger command (*L1Accept*), initiates and directs the collection of event data from its associated PC. Each CC in a Platform contributes to flow control. Flow control is modeled within the Platform by back pressure coupled to trigger regulation. Trigger regulation is accomplished by asserting *full/available* back to the FCTS. The CC maintains the buffer management model for its FEEs.

B. ROM Performance

The ROM can be thought of as a pipelined series of stages through which data must be transferred. The time averaged throughput of the ROM is therefore limited to the throughput of the slowest of these stages. Figure 3 illustrates the maximum event rates that can be achieved for various event sizes. Separate curves are shown for: data transfer from the FEE to the PC for front end clock rates of 15 and 60 Mhz; DMA transfer from the PC to the SBC across the i960 bridge; and minimal feature extraction (FEX) in the power PC (PPC). The feature extraction performed is a simple copy of the input data. As can be see the ROM performance is more than adequate to satisfy the throughput requirements at the nominal operating point.

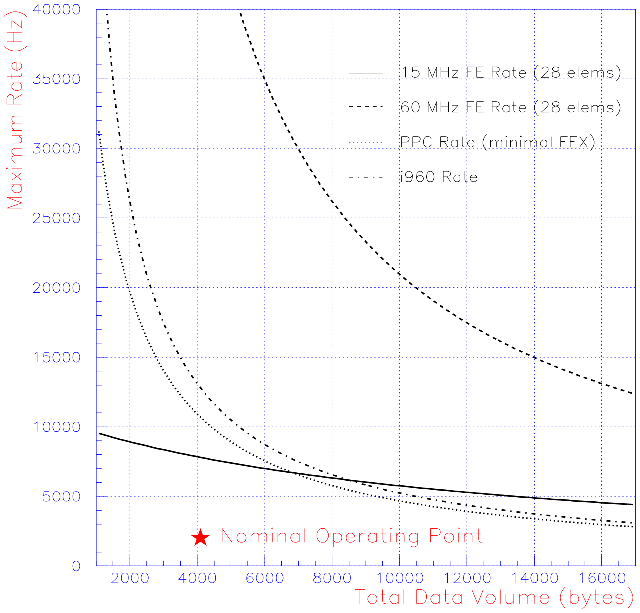

Figure 3: ROM Data Transfer Performance.

V. Event Builder

Event assembly at BaBar occurs in two stages - the Crate (or *Fragment*) level and the *Event* level. At the *Fragment* level, an entire crate's contributions are assembled in the Slot 1 ROM. In the final system, this build will occur over the VME backplane. Currently, however, this build is being performed over ethernet. This stage of the event build is the bottleneck in the current system.

At the *Event* level, the Fragment level contributions from all crates are assembled on a Unix box. Up to 32 event levels are supported, with Fragment level contributions being vectored to the correct Event level on the basis of the *timestamp* on the event. After an entire event is assembled by any Event level, it is passed to a Level 3 process (one per Event level). All the Level 3 results are sent to a single Logging Manager which the writes the output to disk.

Currently, the system can accept 1.3 kHz of L1Accepts without incurring deadtime. We expect the Fragment level VME backplane build to increase the performance to the design value of 2.0 kHz.

VI. Conclusion

The BaBar data acquisition system is currently taking colliding beam data at a Level 1 trigger rate of 500Hz, and writing Level 3 output to disk at a rate of a few Hz. We have shown that the data acquisition platform is capable of delivering Level 1 trigger data to Level 3 at 1.3 kHz in the current implementation, and we believe that the VME backplane build will support a Level 1 rate of 2.0 kHz. The backplane build, together with improvements in the Level 3 algorithms, should allow us to satisfy our design for a 2.0 kHz Level 1 rate with a 100 Hz Level 3 output rate with negligible deadtime.

VII. Acknowledgments

We gratefully acknowledge Department of Energy contracts DE-AC03-76SF00515 and DE-FG02-91ER-40664 for supporting this work.

[1] BaBar Technical Design Report, SLAC-R-457 (1995).

[2] R. Claus, *et al.*, “Development of a Data Acquisition System for the BaBar CP Violation Experiment”, this conference.

[3] I. Scott, *et al.*, “BaBar Data Acquisition”, Proceedings of the CHEP 1998 Conference.

[4] P. Grosso, *et al.*, “The BaBar Fast Control System”, Proceedings of the CHEP 1998 Conference.

[5] MVME2306 Single Board Computers are manufactured by the Motorola Computer Group division of Motorola, Inc., 2900 South Diablo Way, Tempe, AZ 85282.

[6] The VxWorks RTOS and Tornado Development interface are products of Wind River Systems, Inc., 1010 Atlantic Ave., Alameda, CA 94501-1153.