Fermi National Accelerator Laboratory

FERMILAB-Conf-89/259

Portable Software for Distributed Readout Controllers and Event Builders in FASTBUS and VME*

Ruth Pordes, David Berg, Eileen Berman, Mark Bernett, David Brown, Penelope Constanta-Fanourakis, Terry Dorries, Margaret Haire, Umesh Joshi, Ken Kaczar, Bryan Mackinnon, Carmenita Moore, Tom Nicinski, Gene Oleynik, Don Petravick, Gary Sergey, Dave Slimmer, Jonathan Streets, Margaret Votava, Vicky White
Fermi National Accelerator Laboratory
P.O. Box 500
Batavia, Illinois 60510

December 1989

* Submitted to the 1989 IEEE Nuclear Science Symposium, San Francisco, California, January 15-19, 1990

Operated by Universities Research Association Inc. under contract with the United States Department of Energy

Portable Software for Distributed Readout Controllers and Event Builders in FASTBUS and VME[1]

Ruth Pordes, David Berg, Eileen Berman, Mark Bernett, David Brown, Penelope Constanta-Fanourakis, Terry Dorries, Margaret Haire, Umesh Joshi, Ken Kaczar, Bryan Mackinnon, Carmenita Moore, Tom Nicinski, Gene Oleynik, Don Petravick, Gary Sergey, Dave Slimmer, Jonathan Streets, Margaret Votava, Vicky White

Online and Data Acquisition Software Groups
Fermi National Accelerator Laboratory
Batavia, Illinois 60510

Abstract

We report on software developed as part of the PAN-DA system to support the functions of front end readout controllers and event builders in multiprocessor, multilevel, distributed data acquisition systems. For the next generation data acquisition system we have undertaken to design and implement software tools that are easily transportable to new modules. The first implementation of this software is for Motorola 68K series processor boards in FASTBUS and VME and will be used in the Fermilab accelerator run at the beginning of 1990. We use a Real Time Kernel Operating System. The software provides general connectivity tools for control, diagnosis and monitoring.

A. Purpose and Scope

In response to experiments' needs for decreased data readout dead-time and increased event acquisition and logging rates, the number of intelligent modules included in the data acquisition architecture is multiplying. As the intelligence of these modules increases, software development and support is needed, not only to understand and diagnose their actions, but as an integral part of the data acquisition function they will perform. This paper reports on a data acquisition system (known as PAN-DA)[1] that has been developed in conjunction with support for selected such modules, to provide full software support for each module independently, and additionally as part of an integrated data acquisition environment. The design and implementation of the software has been carefully engineered to allow support for new modules to be rapidly and easily provided.

During the last fixed target run at Fermilab, experiments typically included a few intelligent modules (z8000, m68000 based) as part of their front end data acquisition system. Online systems contained a few computers - pdp-11s, single VAX 11/780 or uVAX. 1/2 Mbyte/sec data throughput from front end to logging on 9 track tape was required [2]. In the fixed target run starting in February 1990, several experiments will acquire data at 1-5 Mbytes/sec steady state. Their data acquisition architecture includes many intelligent modules (M68020, DSP based). Software filtering algorithms will be performed on the data before it is logged. These throughput rates require high speed logging to multiple I/O devices in parallel, to prevent this being the bottleneck in the system. Because of the spill nature of the particle beam, parallel event buffers are used to increase the total amount of data that can be logged and analyzed. Online systems of 10-20 workstations will monitor and control the data acquisition.

We have responded to these trends by developing support for a set of hardware modules which address specific needs of experiments, and which together provide an integrated solution to these expanded requirements. The system supports data acquisitions systems whose front ends are in CAMAC, FASTBUS and/or VME.

This software and hardware environment provides for

- o high efficiency data readout on the main data path;
- o access to a sub-set or all of the event data at each stage of the data acquisition pipeline;
- o monitoring and distribution of the data;
- o integrated control and synchronization of the data acquisition system;
- o parallelism in the data flow, to reduce dead-time and increase throughput;
- o central management of the data buffers in distributed processor nodes;
- o system diagnostics and debugging tools;

o uniform communication and control across hardware platforms.

The hardware supported as part of the first implementation provides for building of the events in FASTBUS, processing and logging of the data from VME, and distribution of the event data over TCP/IP to VAX/VMS and UNIX workstations.

The software developed has been designed to be general in nature but easily massaged to allow each experiment to tailor it to their particular requirements. It has been implemented to make it easily portable from one hardware platform to another. Several Fermilab fixed target experiments (E687, E791, E771, E789, E706) are planning to use parts or all of this software in the 1990 run (Figure 1, Figure 2)

E687 Topology Using GPM

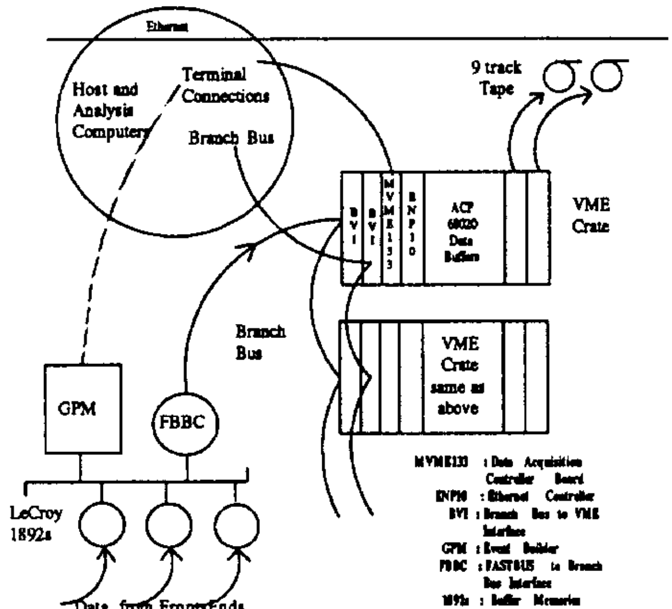

Figure 1

B. Supported Hardware Platforms

The following modules are currently supported:

- o Motorola MVME133A processor board. This is a 68020 based CPU board and includes a real time clock, VME Slave and Master interface, and support for VME interrupts.
- o the FASTBUS General Purpose Master (GPM), a FASTBUS 68000 based readout controller developed at Cern [3],
- o and the FASTBUS Smart Crate controller (FSCC), a FASTBUS 68020 readout controller developed at Fermilab [4].
- o ACP 1 68020 - the Fermilab developed 68020 processor board, with 2 or 6 Mbytes of RAM [5]

In the near future we will port the relevant software to the MVME147 68030 based VME processor board, and to the FASTBUS CERN Host Interface (CHI). We are currently developing support the ACP 2 (R3000) risc based VME processor board [6]. As a measure of the time required to provide software access to a new controller board, the underlying system software was ported on the FASTBUS Smart Crate Controller with less than two weeks effort. (Figure 3)

E771 Topology Using FSCC

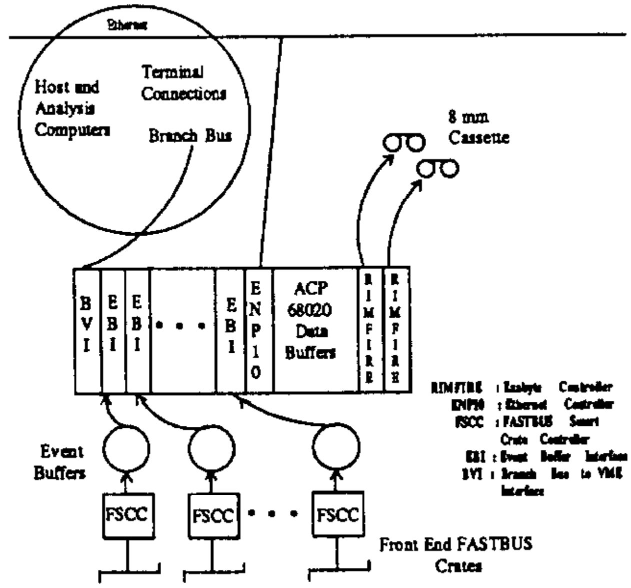

Figure 2

C. Software Environment

PAN-DA software applications are based on the existence of a multi-tasking real-time operating system. For the 68xxx boards we use pSOS [7], a fast, efficient real time kernel of Unix flavour. All the application software developed relies in a modular and minor way on the operating system environment; requiring really only such system services as an "event signal" facility, timeout support, and some way for multiple processes to share data structures. The software is designed to allow it to be easily migrated to UNIX. This has already begun and confirmed our claim that the software can be transferred easily to new hardware platforms which support a multi-tasking operating system. As an aid to program and system debugging we have added extensions to the psos "environment", which are available across all the 68xxx platforms we support [8].

FASTBUS Smart Crate Controller

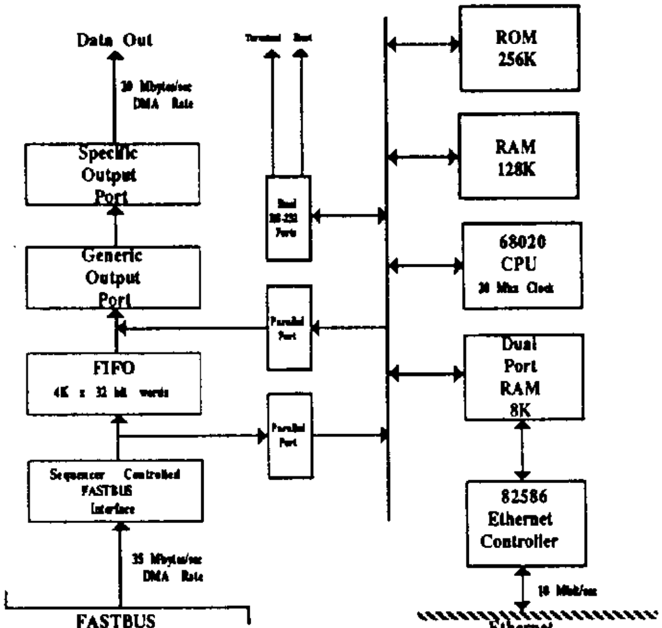

Figure 3

D. Event Readout Software

For the FASTBUS readout controllers, the GPM and FSCC, implementations of the FASTBUS standard routines [9] have been provided. In addition, a set of low overhead, function specific macros for accessing the main data bus, have been developed. These macros have the same call interface as the standard routines, and thus can be used to replace them in an application with no code changes required.

Data acquisition control software has been developed to provide an initialization and control framework into which the experimenter can include his own specific data readout code. This philosophy allows the user to tailor the actual data readout to be as efficient as possible. This software, FEREAD, is available for the GPM and has been ported with no effort to the MVME133 (Figure 4) [10]. It will be easily portable to the FSCC.

Readout Controller Software Structure

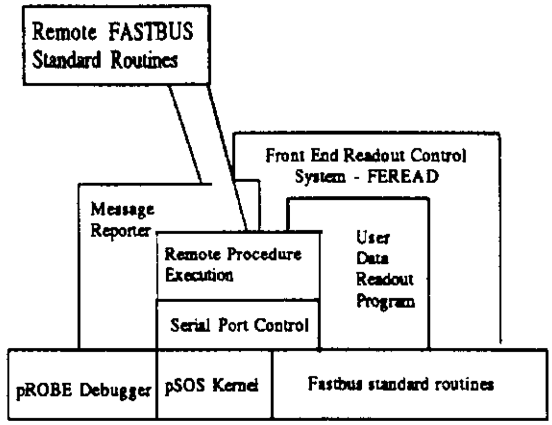

Figure 4

E. System Communication and Integration

Communication of control and data information between front end readout controllers and the host computers is an essential part of any online system. PAN-DA includes software to provide for such communication between modules.

The MVME133A and the FASTBUS Smart Crate Controller (FSCC) have access to ethernet. The MVME133 uses a companion commercial Vme Ethernet controller board (CMC ENP-10) with associated TCP/IP firmware and user interface software. The FSCC includes an on-board Intel 52586 based ethernet interface. We are porting the Berkeley TCP/IP package to this module paying careful attention to providing the mechanisms to allow it to be easily ported to any new module [11].

Some modules, such as the GPM, only have RS232 terminal lines for control. A connection oriented communications protocol over RS232 serial line is provided as part of the general system support tools. Any new module which has a terminal connection, can be quickly brought online and made accessible from the software on the back end hosts.

A sophisticated Remote Procedure eXecution package has been implemented for both Ethernet (TCP/IP) and Terminal Line communications [12].

This tool provides a framework where programs are easily developed whose actions are to be distributed across several CPUs. Part of RPX includes a VAX/VMS based compiler, where a text file description of the program subroutines, together with a description of their parameters, and platform on which they will actually be executed, can be used to automatically generate the underlying communication and data conversion code to link the subroutines and pass the required data between the CPUs. RPX is a major tool in integrating the many modules and sub-systems into a single data acquisition system (Figure 5).

In addition RPX gives us significant hardware platform and operating system independence for the applications coded using it. The use of "sockets" as the interface layer between RPX and the communications layer has already enabled us to easily port code to a UNIX based MIPS R120 system. This experience proved to us that our implementation is "UNIX compatible" . We expect, therefore, that as new RISC CPU based front end controllers and processor modules arrive on the scene we will be able to quickly provide software for use on these boards.

RPX Execution Flow

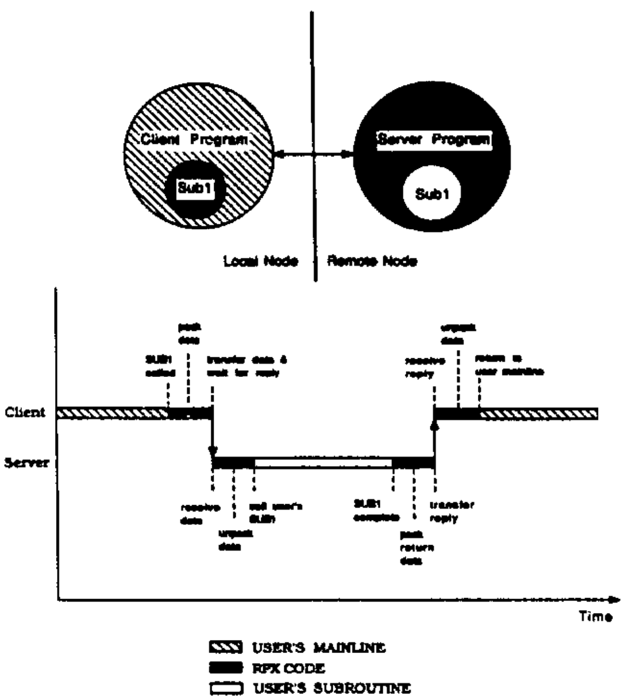

Figure 5

F. General Processor Master (GPM) Support

So far, the GPM has been used at Fermilab as a fastbus readout controller and event builder.

The GPM is being used to read and build event data from FASTBUS memories in a local crate and then write that data to the next level of the data acquisition system. The architecture we have implemented provides for the GPM to push the events through the Fermilab ACP Branch Bus to VME Crates of buffer memories and processor boards. In this architecture, where the data must pass over the backplane twice, data throughputs of up to 4-5 Mbytes/sec steady state have been sustained.

Additional software provides for data acquisition control of the GPM and to allow the experimenter access to the event data for formatting and monitoring as needed.

E706 Topology Using the GPM

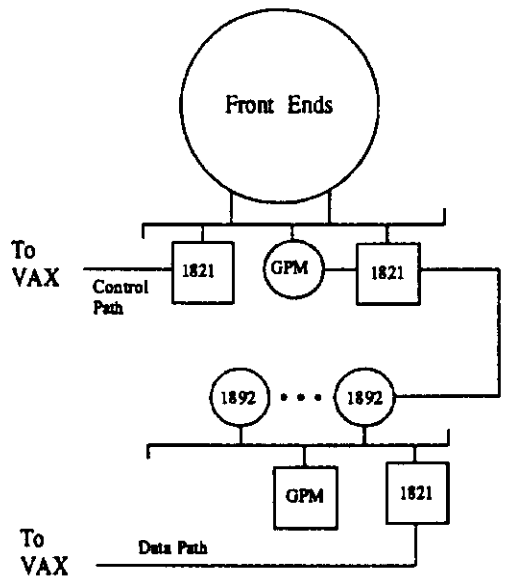

Figure 6

The GPM is also used as the "brain" in conjunction with the LeCroy 1821 Segment Manager as the readout controller (Figure 6) [13]. Although the 1821 moves data efficiently from FASTBUS to VAX Q-Bus, Unibus or to data buffers in CAMAC, FASTBUS or VME, it has very limited programmability and processing capability. Front panel signals available on the GPM and 1821 are used to synchronize between the two cooperating controllers. The software developed includes the FASTBUS Standard routines and inline macro equivalents; software for handling the interrupt ports; code to allow the GPM to be used as a FASTBUS slave - data buffered in the GPM is read out by the 1821 - and special purpose microcode in the 1821 [14].

As part of the system support provided for the GPM, an RPX implementation of the fastbus standard routines is underway. VAX/VMS or Unix based programs will be able to perform fastbus operations for front end monitoring and diagnostics through the GPMs terminal port.

G. MVME 133A Support

We have developed system support tools to provide support for the MVME133a 68020 based processor board running under pSOS. This includes a device driver and associated socket routine library for communication over ethernet using the ENP-10 ethernet controller board. RPX is implemented over both the ENP10 and RS232 ports of the MVME133. Application software provides for the MVME133 to control the data flow and to control the distribution event data from VME buffers to any cpu supporting TCP/IP.

Software Components of MVME133 Data Acquisition Board

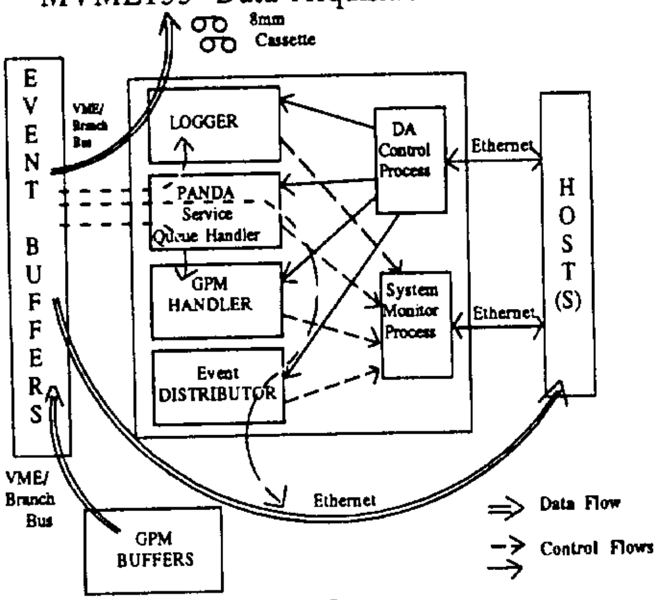

Figure 7

The MVME133 provides centralized data handling and control for multiple "intelligent event buffers in VME" [15]. In the first implementation, these are provided by ACP 1 68020 processor boards, running the LUNI operating system.

Software packages to support the Ciprico 9-Track tape (tapemaster and Rimfire Exabyte adaptor cards has been developed for use on the MVME133A [16]. Data acquisition logging applications are also provided for both these devices (Figure 7).

H. Fastbus Smart Crate Controller

The 68020 based FASTBUS Smart Crate Controller has been developed at Fermilab to provide high speed readout and control of TDCs, ADCs and the new Silicon Strip Detector systems [17]. It provides an enhancement over the GPM in that it has a high speed 32-bit parallel output port in addition to the FASTBUS crate interface, and supports communications over Ethernet. The data pipeline - from FASTBUS through an on-board 512 word FIFO through the output port to upper level event buffers - can act without CPU intervention. This allows for high speed readout. A second data fifo in parallel with the main data path, allows for some data to be made available to the CPU without compromising the readout time. The CPU is then free to perform monitoring and diagnostics on the data, and handle the transfer of information to the upper level host.

The FSCC software has been developed by porting a large portion of the GPM software. FASTBUS Standard routines and macros, remote FASTBUS operations using RPX, message reporting, and readout control are available.

I. Summary

The first implementation of the PAN-DA system was successfully demonstrated in November. It met the throughput requirements of the system and will be used as a system for Experiment 687 in their February 1990 run. Initial experience with transporting the software to new hardware environments indicates that we have met our goals of providing software that is easily ported and maintained.

J. References

[1] D. Petravick et al, "The PAN-DA Data Acquisition System", IEEE Transactions on Nuclear Science, Vol.NS-36, No.5, October 1989.

[2] V.White, et al, "The VAXONLINE Software System at Fermilab" IEEE Transactions on Nuclear Science, Vol.NS-34, No.4, August 1987.

[3] The CERN Host Interface, R. Mclaren et al, IEEE Transactions on Nuclear Science, Vol.NS-35, No.1, February 1988.

[4] FASTBUS Smart Crate Controller, Design Specification, October 12th 1989. Internal Fermilab Document.

[5] I. Gaines et al. "Use of the Fermilab Advanced Computer Program Multi-Microprocessor as an On-Line Trigger Processor", IEEE Transactions on Nuclear Science, Vol. NS-32, No. 4, August 1985.

[6] J. Deppe et al, "ACP/R3000 Processors in Data Acquisition Systems" J. Deppe, et al, IEEE Transactions on Nuclear Science, Vol.NS-36, No.5, October 1989.

[7] Trademark of Software Components Group, Inc.

[8] D.Berg, et al. "A Real Time Integrated Environment for Motorola 680xx-based VME and FASTBUS Modules" IEEE Transactions on Nuclear Science, Vol.NS-36, No.5, October 1989.

[9] FASTBUS Standard Routines User Guide for the GPM, PN266, T. Dorries, T. Kozlowski et al. Internal Fermilab Document. IEEE Transactions on Nuclear Science, Vol.NS-36, No.5, October 1989.

[10] T.Dorries, M.Haire, C.Moore, R.Pordes, M.Votava, "FEREAD Front End Readout Software for the Fermilab PAN-DA Data Acquisition System" IEEE Transactions on Nuclear Science, Vol.NS-36, No.5, October 1989.

[11] M.Bernett, G.Oleynik, "Uniform Communications Software Using TCP/IP" IEEE Transactions on Nuclear Science, Vol.NS-36, No.5, October 1989.

[12] E.Berman, D.Petravick, G.Sergey, " "Remote Procedure Execution Software for Distributed Systems" IEEE Transactions on Nuclear Science, Vol.NS-36, No.5, October 1989.

[13] LeCroy FASTBUS Model 1821, LeCroy catalog.

[14] R.Pordes et al, "List Processing Software for the LeCroy 1821 Segment Manager Interface", IEEE Transactions on Nuclear Science, Vol.NS-34, No.4, August 1987.

[15] PBS - PANDA Buffer Services, DS190, D. Berg et al. Internal Fermilab Document.

[16] S.Bracker, V.White, et al "Exabyte Helical Scan Devices at Fermilab" IEEE Transactions on Nuclear Science, Vol.NS-36, No.5, October 1989.

[17] See accompanying paper, "Silicon Strip High Rate FASTBUS Readout System." C. Swoboda et al.

---

1 sponsored by DOE contract No. DE-AC02-76CH03000