CONF-890545--7

CONF-890545--7

DE89 015697

DATA ACQUISITION FOR FNAL E665

D. F. Geesaman, M. C. Green*, S. Kaufman, S. Tentindo-Repond†
Argonne National Laboratory, Argonne IL 60439
J. F. Bartlett, H. L. Melanson, D. Petravick
Fermi National Accelerator Laboratory, Batavia, IL 60510
D. G. Michael
Harvard University, Cambridge MA 02138
D. McLeod
University of Illinois at Chicago, Chicago, IL 60680
M. Vidal
Max-Planck-Institut fuer Physik und Astrophysik, Muenchen 40, FRG

Received by
JUL 0 8 1989

Abstract

The data acquisition system for FNAL E665, an experiment to study deep inelastic muon scattering from nucleons and nuclei, is described. The system is built with the FNAL VAXONLINE and RSX DA building blocks. The structure, capabilities and limitations for data flow, control and monitoring are discussed.

I. INTRODUCTION

E665 is a fixed target experiment at FNAL constructed to study deep inelastic muon scattering at the highest possible energies to provide the best spatial resolution in studies of nucleons and nuclei. The E665 collaboration involves thirteen institutions and over eighty physicists. In this paper, the data acquisition system developed for E665 is described. This system relies heavily on the FNAL VAXONLINE building blocks.[1] As such, it serves as an excellent example of the application of the VAXONLINE system to a running experiment. In addition, the actual design implementation for data acquisition and monitoring illustrates several features which are important for the success of the experiment.

II. E665 APPARATUS

A new muon beam line (NM) was constructed at FNAL for this experiment. [2] The fixed target proton beam is extracted from the Tevatron in a 20 second spill every 57 seconds. This spill structure sets the parameters for the buffering required in the data acquisition system. In 1987, the beam line was typically operated to produce $2 * 10^7$ muons/spill with an average energy of 490 GeV, and it is capable of order of magnitude higher fluxes, particularly at lower muon energies. A detailed description of the apparatus is available in ref [3]. The key features of the experiment are: 1) Two large (4 T-m and 7 T-m) superconducting magnets for momentum analysis. 2) A streamer chamber surrounding the target to detect low energy fragments. 3) 75 tracking chambers for momentum analysis 4) Hadron particle identification over most of the kinematic range with a time of flight system, two threshold Cerenkov counters and a ring imaging Cerenkov counter. 5) An electromagnetic calorimeter for neutral energy detection 6) A muon detection system following 3 m of steel absorber providing a 3 mr trigger. 7) A beam veto small angle trigger capable of triggering to scattering angles of 0.3 mr. Most of the vertex detector system between the two magnets was employed in the NA-9 experiment at CERN [4] and was moved to FNAL in 1984 with existing CAMAC read-out systems.

III. GOALS FOR DATA ACQUISITION

The goals set in 1984 for data acquistion which governed the design of the data acquisition system were:

1) Acquire 2000 events per spill. Each event was expected to be 2000-5000 bytes in length.
2) Maintain a computer live time of 80%.
3) Incorporate existing CAMAC interfaces.
4) Provide a system to incorporate monitoring at all levels.
5) Provide a reliable system.
6) Provide a system which is easily modified to incorporate new equipment.

In practice the average event size reached 10000 bytes. While the anticipated time-averaged throughput of 130 kbytes/s was exceeded, the number of events per spill was compromised. The performance is discussed in more detail in Section IX.

IV. FNAL TOOLS

The online computing group at FNAL, led by V. White, provides a powerful suite of software building blocks for a data acquisition system. The major product, the VAXONLINE system was developed concurrently with the construction of E665, and this experiment with other FNAL fixed target experiments provided the demanding customers which gave direction to the final development of VAXONLINE. E665 attempts to make maximum use of these products in their generic forms. A brief summary of these building blocks is given here:

RSX DA — RSX DA is a set of RSX-11M tasks to control data acquisition and logging for in-spill events on PDP 11 systems.[5]

Connected Machines — The CD software controls 16 bit parallel transfer of data between PDP11 , VAX and μVAX processors via DR11-W UNIBUS and DRV11-W Q-bus controllers (CDlinks).[6] This provides the major path for data flow between processors.

DAQ — DAQ is a VMS shared event buffer and user interface developed for CDF. This provides the central framework of event buffering and distribution within a single VMS machine.[7]

EVENT_BUILDER — EVENT_BUILDER provides the central data input for VMS components. The EVENT_BUILDER accepts subevents from several external sources, including CDlinks and FASTBUS, concatenates subevents into complete events and enters these events into the DAQ pool.[8]

OUTPUT — OUTPUT logs the data in the DAQ pool to tape or disk.[9]

BUFFER_MANAGER — BUFFER_MANAGER distributes data from one VMS machine to another machine.[10]

RUN_CONTROL — RUN_CONTROL provides the control and coordination of tasks on several front end PDP 11's.[11]

COURIER — COURIER is a central message display and logging facility and a user interface.[12]

VAXONLINE also contains a number of subroutine libraries which provide essential interface elements. This includes a general command/menu interface MENCOM [13] which provides the user environment for the building blocks. Many of the features of VAXONLINE are discussed in further detail in ref. [1].

V. HARDWARE ARCHITECTURE

The hardware architecture of the data acquisition system is shown in Figure 1. Since a significant fraction of the experimental apparatus had existing CAMAC digitization, brute force parallelism was employed to achieve the required read-out speed. Each of the three PDP 11/34 front end processors (BISON 61, BISON 64 and BISON 65) possessed two parallel CAMAC branches, each controlled by a JORWAY 411 branch driver. During data acquisition, simultaneous direct memory access (DMA) transfers from CAMAC occur on the two branches. One branch on each PDP contains only fixed length data blocks to ensure a fixed offset in the event for data from the variable and fixed length data blocks on the second branch. Data interrupts are received by a special purpose DR-11C interface, the FNAL Bison Box. Only two such interrupts are defined, a SPILL interrupt signaling a begin or end of the extracted beam spill, and an EVENT interrupt.

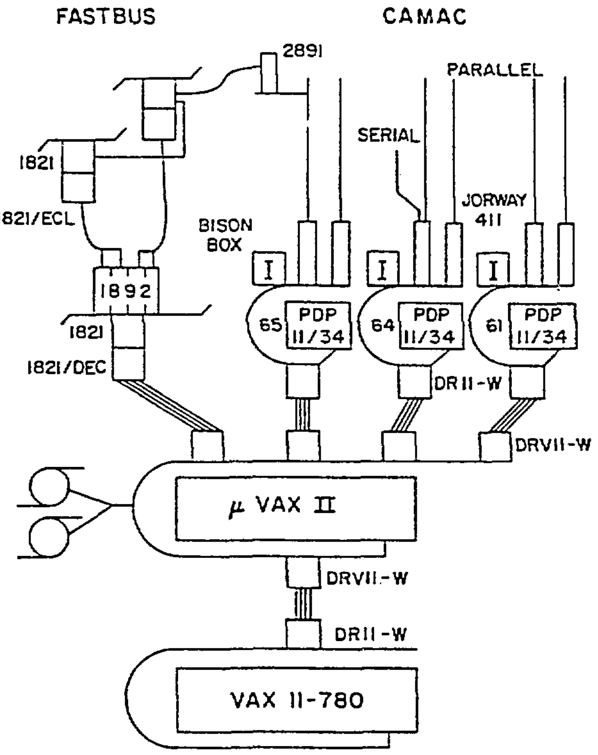

Figure 1: Hardware Architecture

Each of the PDP 11/34's possessed 2Mbytes (BISON 61 and 64) or 1Mbyte (BISON 65) of Konelar UNIBUS bank switchable bulk memory. These serve as the data buffers for the PDP 11's and were capable of holding about 65-90% of the data from one complete spill. The data are then transferred by 16 bit parallel DR11-W controllers to the μVAX II .

BISON 64 also possesses one serial CAMAC branch for control functions at remote locations. The serial branch is also used to read scalers at the end of each spill.

The electromagnetic calorimeter required a much faster data acquisition and buffering path as its 5 kbytes of data per event accounted for $\sim$ 50% of the entire event length. The calorimeter analogue-to-digital converters (LRS1885's) are housed in two FASTBUS crates. The two crates are read out in parallel using LRS1821 segment managers into LRS1892 4Mbyte memories with a total data buffer size of 8 Mbytes per branch. A third 1821 segment manager with an 1821/DEC personality card is used to transfer blocks of events from the memories via a parallel interface (MDB-DRV11-W) to the μVAX II .

The μVAX II serves as the central computer for data acquisition and control. It receives data from the four front ends, (three PDP 11/34's and FASTBUS) and provides the CPU engine to build events, write data to 6250 b.p.i. magnetic tape, distribute events for analysis and serve control functions. Two Storage Technology 2925 6250 b.p.i. tape drives are used for data logging. A VAX 11/780 provides the analysis power for on-line monitoring and message logging and the central location for software development.

VI. SOFTWARE ARCHITECTURE

Many of the key features of the architecture can be illustrated by following an event through the data acquisition system. (Figure 2) At the beginning of each spill, a begin of spill interrupt alerts each PDP 11/34 to prepare for data acquisition. The RSX DA system consists of five coordinated tasks, one of which, SYNCH, synchronizes the action of the main data acquistion task, DA with the spill structure. DA blocks other tasks from access to CAMAC during the spill interval, issues begin-of-spill CAMAC commands to ensure that the hardware is in a known state and waits for interrupts. On the receipt of an event interrupt, data acquisition commences. The event read-out can be specified by a list processing interpreter or a higher performance assembly language routine. The list processing option is controlled by a simple text file which is easily modified by users. Execution of the assembly language directed-read-out is one list element. List processing is extremely valuable in the equipment development and debugging phase of the experiment. Primary data acquisition uses the assembly language directed-read-out with typical timings illustrated in Table I. Coordination of the read-out on the two branches is organized by standard jump tables. The data are read out by direct memory access control of the branch drivers into the Konelar bank-switchable bulk memory.

The computer busy consisted of the logical .OR. of the busy signals of each front end. The average dead time per event was 3.5 ms.

Asynchronously, as CPU time was available, or when the bulk memories became full, the LOG task sends 8-12 kbyte blocks of event data to the μVAX II .

For the calorimeter, the two front-end 1821's act as free-running processors during the spill. Special data acquisition microcode [16] was written for these 1821's to allow for response to experiment interrupts and to perform all operations needed for read-out without any input from the PDP's. An .OR. of 1821 busy signals is provided to the experiment to inhibit triggers during read-out or if the 1892 memories are full. The present 1885 ADC's require $\sim 750\mu s$ for digitization after which read-out proceeds at a rate of 420ns per word for a total FASTBUS deadtime of around 1.4ms. CAMAC read-out time for the same data (with two branches) would have been over 4 ms. During interspill periods, the ADC's are read out using the same 1821's but via CAMAC (LRS2891) into one of the PDP's for monitoring.

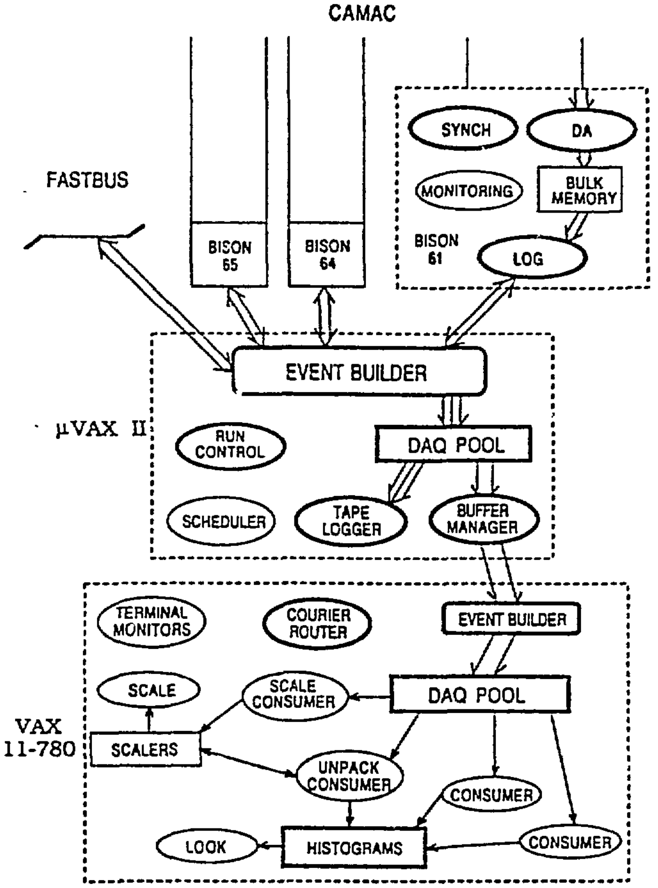

Figure 2: Software Architecture

Table I

PROCESSOR	Length (bytes)	Dead Time
BISON 61	2200	3.5 ms
BISON 64	1560	3.0 ms
BISON 65	960	2.0 ms
FASTBUS	5184	1.4 ms
TOTAL FRONT END		3.5 ms
μVAX II	10040	
Throughput	1350 events/spill 23.7 events/sec	

At the end of each accelerator spill, end of spill CAMAC operations are performed on the PDP 11/34's and then the CAMAC branches are released for monitoring tasks. The primary DA task is usually checkpointed and data logging continues during most of the interspill interval.

The heart of the VMS data acquisition software consists of the EVENT_BUILDER which receives all data and builds complete events and the DAQ shared common which serves as a data buffer and distribution framework. The EVENT_BUILDER on the μVAX II receives unsolicited subevents over the CD links from any of the PDP 11/34's as well as from itself and the VAX 11/780. FASTBUS is accessed more directly by the EVENT_BUILDER using a combination of special microcode for the 1821 and code written in a special list-processing language developed by the Fermilab Computing Department [16] [17]. When buffer space is available on the μVAX II, the event builder "requests" data from FASTBUS by executing a QIO in which the "FASTBUS List" will be executed. Along with the 1821 microcode, the FASTBUS List contains all of the 1821 instructions which are necessary to read events from the 1892 memories and keep track of memory pointers and resetting. Buffers had a capacity of 30 kbytes which allowed for up to 6 events per buffer. Events from the two halves are concatenated into a single event by the 1821 prior to transfer to the μVAX II. With each buffer completely filled, 6.3 ms of μVAX II CPU time were required for each FASTBUS subevent transferred.

From internal header information in each subevent, the EVENT_BUILDER decides if a subevent should be concatenated into a four source event, or built as an independent event. A special hardware input register, which contains the event number and real time information, provides the basis for determining the related subevents. A limited capability is available for error correction of events with missing subevents or unusual event numbers due to hardware problems. The ability to incorporate data into the event stream from any source was an essential component of the monitoring strategy. Finally, the EVENT_BUILDER stores blocks of events into the DAQ shared common.

The DAQ system provides a framework to identify and distribute events with the enumeration of a list of event requirements . The OUTPUT process receives all the events in the DAQ pool and writes 9600 byte blocks to 6250 BPI tape drives, automatically switching between two drives as tapes are filled. Another event consumer process, the BUFFER_MANAGER, obtains selected events including all begin- and end-of-spill events, monitoring events, and as cpu time allows, data events and transmits these over the parallel link to the VAX 11-780 for analysis.

The VMS machines are essentially isolated from the spill structure of the accelerator by the large front end buffering.

The same software framework is implemented on the VAX 11-780 with an EVENT_BUILDER to receive the events from the BUFFER_MANAGER and enter them into the DAQ pool. A subset of a large number(~20) of event-consuming monitoring tasks perform on-line monitoring and analysis. Each consumer is able to request only those events which contained the type of information it desires.

VII. CONTROL AND COMMUNICATIONS

The μVAX II serves as the control center for the experiment. The smallest unit of data sets, the run, is defined completely in terms of the logging of data to tape. Runs are not allowed to span tapes, though one tape can hold several runs. Data acquisition is not halted on the front end machines between runs and in automatic tape switching mode, no data is lost between runs. Care is taken to make the data self-normalizing.

A RUN BLOCK is defined as the interval between the start of data acquisition on the front ends and the termination of data acquisition on the front ends. Under ideal conditions with all equipment operating properly, a run block can last several hours. At the start of a RUN BLOCK, hardware initialization and synchronization could take several minutes.

The control and synchronization to start and stop RUN BLOCKS is provided by the RUN_CONTROL process on the μVAX II and associated run control servers on each of the PDP 11/34's. RUN_CONTROL checks that the EVENT_BUILDER is active, notifies OUTPUT to start tape logging with the appropriate run number and via the servers, activates and synchronizes the BEGINR tasks on the PDP 11/34's to start data acquisition. Similar functions occur at the end of a RUN BLOCK. The communication with the RSX servers takes place over the CD links.

Message distribution is another vital communications function. Text messages are distributed to a central message router COURIER, on the VAX 11-780 which displays the messages in one of 20 message windows in dedicated terminals and logs the messages to a central log file. DECNET communication with the router is used on the VMS machines. Communications with the PDP 11's and any other intelligence uses RS-232 terminal lines. Terminal line input is received by detached TERMINAL_MON software processes and sent via DECNET to the router. This distribution system proved sufficiently general to be of use to other experiments.

VII. DATA STRUCTURES

In addition to the DAQ event pool on each VMS machine, there are four other data structures used for communication within the computers. On each PDP 11/34, a 2kbyte shared common is available for communication between the data acquisition software and monitoring tasks. Users must communicate with this shared common through interface routines to isolate different detector groups. The VAXONLINE software uses an indexed sequential file and a suite of interface routines to provide a common repository of run information and conditions.[15] This database is DECNET accessible from any VMS node in the system. Histograms from monitoring tasks are stored in group global sections and available for display by a common display program. Similarly, numerical information and hardware and software scalers are stored in global sections for general access. In particular, the hardware scalers read out at the end of each spill, and associated derived information is stored in a SYSTEM global section by a dedicated consumer process and available to all users.

VIII.MONITORING

Since this was the first use of this assembled apparatus in an experiment, it was essential to have a general monitoring strategy which permitted easy modification and independent operation. Several general rules served to guide the monitoring.

1) All communication with CAMAC is on the PDP 11/34's or through auxiliary crate controllers and stand alone processors. On the PDP 11/34's only access via QIO (2.5 ms per operation) operations with DOE standard Fortran CAMAC interface routines[14] is permitted. Since QIO's to CAMAC are blocked during the spill interval, users generally do not have to be concerned with synchronization with the spill. The price paid for this isolation is the relatively slow access to CAMAC. QIO's were developed to implement CAMAC operation lists to provide some gain in performance.(see below)

2) All monitoring communication with FASTBUS occurred via an LRS 2891 CAMAC interface module. Many CAMAC commands are required for reading data. In order to achieve a sufficiently short read-out time, it was necessary to make use of the list-processing option in the Fermilab CAMAC driver for the PDP. This allowed for execution of a complete read-out cycle in 10 ms as compared to about 60 ms without list processing. Although slower than might be optimal, this was acceptable for monitoring purposes.

3) From the PDP 11/34's only two means are available to communicate with the rest of the data acquisition system: a) With ASCII messages via terminal lines. b) By formulating information into an event structure and dispatching these events to the μVAX II . A standard subroutine call is provided for this purpose. The number of monitor subevents is limited(administratively) to less than 100 per spill per front end. This does not prove to be a significant limitation.

Users generally developed a number of small RSX tasks to perform control and monitoring functions such as to load trigger matrices and to strobe wire chambers and light-emitting diodes for wire map and gain information. These tasks can be run stand alone, started by BEGIN-RUN-BLOCK ,END-RUN-BLOCK or INTERSPILL RSX command files or time scheduled by a VMS SCHEDULER running on the μVAX II which communicated with the front ends via the RUN_CONTROL servers. This procedure allows an essential degree of flexibility and ease of development as the character of the experiment changes from the debugging stage to production data acquisition. The message system and the event stream are always available for monitoring tasks even when normal data acquisition is not in progress. The inherent limitations in RSX task size and the CAMAC QIO access time led some detector groups to provide stand alone processers for monitoring. These usually communicate with the main data stream via CAMAC memories and simple RSX read-out tasks.

All other monitoring was performed by independent event-consuming tasks on the VAX 11-780. A template consumer is provided to encourage a uniform interface to users and a single interface routine is provided to locate data from any CAMAC DMA within the event. Each consumer is able to store histograms and scalers in group global sections for display by separate processes. Several of these monitoring consumers were run continuously; others were started in a regular rotation by shift personnel to check individual detector performance.

The most important monitoring processes are:

1) A consumer (UNPACK_CONSUMER) which continuously checks the event structure, including the number and length of each CAMAC DMA operation, the consistency of header information and the general distribution of events through the software chain.

2) An online event display which provides a sophisticated graphical display of decoded information from an event.

3) A consumer to maintain and update the system scaler information including time average beam characteristics and event rates.

Since a number of consumer processes were generating histograms and statistical information, separate utilities were developed to display the information generated by any monitoring process. LOOK is an interactive shell for HBOOK[19] and HPLOT[20] to display and output histogram information based on a subset of the GHIS system [18]. LOOK can access online histograms or disk files. Similarly, SCALE is a screen definition and display utility to display pages of numerical information. With these two utilities, a user can survey several monitoring processes, and compare the current status with standard results.

IX. PERFORMANCE - LIMITATIONS

The performance characteristics of elements of the data acquisition system and the total throughput are given in Table I. The time-averaged maximum throughput was 240 kilobytes per second. This was limited primarily by the CPU speed of the concatenation μVAX II , not by the front end dead time and buffering. At typical data rates of 1200 events per spill, the average live time is 80%. The μVAX II is being replaced by a faster CPU for the next run cycle. However at factors of two higher throughput, tape logging and perhaps Q-bus bandwidth are expected to become limiting.

The primary limitation of the present system lies in the restricted communication between the VMS machines and the front ends. This makes recovery from errors difficult. In this sense, the concatenation μVAX II was special since communication via the CD links was possible. However, analysis processes running on the VAX 11/780 were not able to communicate with the PDP 11 tasks that could actually access CAMAC. The data stream provided an excellent one way path of communication, but only one way. A more general two-way communication, control and file transfer procedure could be implemented via the CD protocols with some effort. A more satisfactory solution might be to replace the PDP 11/34's with machines with larger addressing space and DECNET communications.

A concatenation and matching scheme such as the one used here places a premium on uniqueness of the trigger. Extra triggers, caused perhaps by noise or crosstalk, can easily produce multiple subevents with the same hardware event numbers, a condition beyond the scope of the EVENT_BUILDER

X. SUMMARY

The data acquisition system developed for FNAL E665 is a good example of assembling a system from the VAXONLINE building blocks. The actual code development required to put together a working system with these tools is modest, and consists primarily of experiment specific read-out lists, command files, and additional data structures for monitoring processes. This is not to minimize the immense effort involved in constructing meaningful monitoring and analysis processes to set up the experiment and verify that it is working properly.

The isolation of user monitoring and the enforced development of stand-alone monitoring tasks make this system especially appropriate for a new and developing experiment. It allows users to work independently to bring pieces of the apparatus online and to diagnose problems. The data stream provides an efficient means of communication for monitoring tasks between the otherwise loosely coupled front end and analysis machines. This technique also ensures that monitoring information is always archived onto data tapes.

This work is supported in part by the Department of Energy, Contract W-31-109-ENG-38 (Argonne National Laboratory), Contract EY-76-C-02-3000 (Fermi National Accelerator Laboratory), Contract DE-AC02-76ER03064 (Harvard University) and the National Science Foundation Contract PHY88-11164 (University of Illinois at Chicago)

References

[1] V. White, P. Heinicke, E. Berman, P. Constanta-Fanourakis, B. MacKinnon, C. Moore, T. Nicinski, D. Petravick, R. Pordes and L. Quigg,"The VAXONLINE Software System at Fermilab", IEEE Transactions on Nuclear Science, Vol NS-34, 763, (1987)

[2] A. Malensek and J. Morfin, "The Tevatron Muon Beam", FNAL TM-1193, (1983)

[3] M. R. Adams et al., "A Spectrometer for Muon Scattering at the Tevatron", submitted to Nuclear Instruments and Methods

[4] J. P. Albanese et al., "The Vertex and Large Angle Detectors of a Spectrometer System for High Energy Muon Physics", Nucl. Inst. V212, 111 (1983)

[5] D.M.Berg, L.Quigg, and P. Heinicke, "A High-Speed Data Acquisition System for PDP-11", IEEE Transactions on Nuclear Science, Vol. NS-32, 1368 (1985).

[6] V. White and D. M. Berg, "CDPACK",FNAL PN-159, (1984)

[7] D. Quarrie,"Data Acquisition System Programmer's Reference Manual" FNAL PN-251 (1985)

[8] P. Constanta-Fanourakis, P. Heinicke, T. Nicinski, D. Petravick, L. Quigg and V. White, "VAXONLINE Event Builder User's Guide" , FNAL PN-278, (1987)

[9] D. Petravick, "OUTPUT Component of VAXONLINE Software User's Guide", FNAL PN-289, (1986)

[10] D. Petravick and G. Oleynik,"Buffer Manager User's Guide", FNAL PN-382, (1988)

[11] E. Berman, "VMS/RSX Run Control User's Guide", FNAL PN-271, (1987)

[12] B. MacKinnon, "VAXONLINE Message COURIER System", FNAL PN-311, (1987)

[13] P. Constanta-Fanouraikis, D. Petravick and V. White, "A Menu Oriented Command Interface Package", FNAL PN-254, (1987)

[14] D. Berg,"DOE Standard Routines for CAMAC under RSX-11M", FNAL PN-245, (1985)

[15] E. Berman, P. Heinicke and L. Quigg, "Guide to Status Manager Database and Routines", FNAL PN-304, (1987)

[16] D.G. Michael, "The FASTBUS Data Acquisition System for E665", FNAL PN-378, (1989).

[17] T. Dorries, C. Moore and R. Pordes FNAL FBN-039.4,FBN-047,FBN-042,FBN-048, (1987).

[18] J. Martin, "GHIS", Indiana University, unpublished.

[19] R. Brun, I. Ivanchenko and P. Palazzi,"HBOOK", CERN DD-EE-81.1, (1981)

[20] R. Brun, and H. Watkins, "HPLOT", CERN DD-EE-80-2, (1982)

* Present Address: LeCroy Research Systems, Spring Valley, NY 10977

† Present Address: Northern Illinois University, Dekalb, IL 60115