

A TOMOGRAPH VMEbus PARALLEL PROCESSING DATA ACQUISITION SYSTEM

N.A. Wilkinson*, M.S. Atkins[†], and J.G. Rogers*

*TRIUMF, 4004 Wesbrook Mall, Vancouver, B.C., Canada V6T 2A3

[†]School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada, V5A 1S6**Abstract**

This paper describes a VME based data acquisition system suitable for the development of Positron Volume Imaging tomographs which use 3-D data for improved image resolution over slice-oriented tomographs. The data acquisition must be flexible enough to accommodate several 3-D reconstruction algorithms; hence, a software-based system is most suitable. Furthermore, because of the increased dimensions and resolution of volume imaging tomographs, the raw data event rate is greater than that of slice-oriented machines.

These dual requirements are met by our data acquisition system. Flexibility is achieved through an array of processors connected over a VMEbus, operating asynchronously and in parallel. High raw data throughput is achieved using a dedicated high speed data transfer device available for the VMEbus. The device can attain a raw data rate of 2.5 million coincidence events per second for raw events which are 64 bits wide. Real-time data acquisition and pre-processing requirements can be met by about forty 20 MHz Motorola 68020/68881 processors.

1 Introduction

Historically, positron emission tomographs have been slice oriented (SO) machines which use detector and machine geometry to limit the accepted emission lines to a narrow angle perpendicular to the axis of the tomograph. Such a geometry enables the use of two dimensional image reconstruction algorithms on the data from a given slice. These algorithms are now well established and may be partially implemented in hardware. A three dimensional image is approximated by piling the distinct slices one on top of the other. Unfortunately the axial resolution of such machines is different from (and usually worse than) the resolution in the two other dimensions and, furthermore, the presence of septa reduces machine sensitivity by limiting detector acceptance to a narrow axial angle.

To overcome these disadvantages, Positron Volume Imaging (PVI) machines increase the axial acceptance angle. Some SO machines may be converted to PVI machines by allowing cross plane coincidences and by having removable septa[1]. However, this may be accompanied by a loss of resolution. Other PVI machines are designed to operate without septa[2,3,4]. A major feature of PVI machines is their ability to achieve true 3-D image reconstruction.

The tomograph data acquisition system is intimately linked to the image reconstruction algorithm used, as the data produced by the data acquisition system often incorporates the first stage of image reconstruction, namely, the generation of two dimensional sinograms for SO machines or data sets of higher dimensionality for PVI machines. Several 3-D image reconstruction algorithms have been proposed[5,6,7,8]. Which algorithm will provide the best image is still an active research area. Thus, PVI machines are still evolving; they may need to accommodate a variety of image reconstruction algorithms including two and three dimensional reconstruction and, possibly, event-by-event backprojection[6,7]. The data acquisition system of a PVI machine must be flexible enough to accommodate several reconstruction methods and, perhaps, very different machine geometries. This implies that the transition from hardware to software based processes should occur as close to the front end of the tomograph as is possible without limiting machine performance.

Therefore, until the data acquisition and image reconstruction technology is well established, PVI machines will require a data acquisition system with flexible processes for acquisition and image reconstruction.

The additional (axial) dimension in the raw data of PVI machines places increased demands on the PVI data acquisition system; for the same event rate, the raw data rate is higher. In addition, the increased resolution of PVI machines necessitates more voxels per 3-D image. PVI machines are expected to have significant scatter contamination and therefore more events may need to be acquired per voxel, compounding the increased raw data rate requirement on any PVI data acquisition system.

Our data acquisition system is designed to accommodate the dual requirements of PVI: flexible, software-based processing of raw data and high data transfer rates. This system consists of a high performance software based data transformation engine (DTE) which takes as input the raw data from a tomograph (two generalized detectors in coincidence, wobble position, physiological data, etc) and produces as output a list of histogram coordinates and/or a histogram.

In such a DTE, there are two possible limitations to event throughput: the maximum data transfer rate and the maximum event processing rate. In overcoming these limitations, we have chosen a parallel array of commercially available processor boards connected over a VMEbus. The choice of a parallel processing DTE is a consequence of the fact that no single cost effective processor can accommodate the minimum processing required, namely, the production of histogram coordinates from the raw data at the required event throughput.

Parallel processing also makes the DTE scalable. This means that, provided the system is not data transfer limited, throughput is an almost linear function of the number of parallel processors in the engine.

The data transfer limitation is overcome by the use of dedicated high speed data transfer devices (the Ironics IV-3272 Full Speed Data Transporter[9]) and high speed disk controllers (the Ciprico RF3200[10] and the Interphase V/SMD4200[11,12]) available for VMEbus. The IV-3272 is particularly applicable to the tomograph DTE due to its FIFO based high speed i/o bus. If a raw coincidence event is 64 bits wide, one IV-3272 is capable of a peak event acquisition rate of 5 million events per second and a sustained rate of 2.5 million events per second. These rates are well above the maximum rate to be expected in present-day tomographs.

2 System Environment

The DTE will perform in a variety of modes such as raw list mode, processed list mode and histogram mode. The DTE is inherently flexible. Thus, formerly incompatible modes, such as simultaneous list and histogram mode in real time, may be mixed.

In list mode, the raw and/or processed event list is written in chronological order directly to a large store. In histogram mode, a time independent histogram is constructed in real time. The real time production of histograms historically necessitated the use of special purpose hardware; however our design achieves an acceptable event throughput using software.

The essential functions of the designed data acquisition system are illustrated in Figure 1. The tomograph front end hardware supplies the coincidence event words to the data acquisition function 'Transfer Raw Data'. This process distributes the PET events to the input stores 'Raw Data' of n parallel processors which process the events into output stores 'Processed Data'. The event record is compressed during the processing. Raw list mode may be implemented by using the process 'Transfer Raw Data' to write the raw events directly to a list mode store.

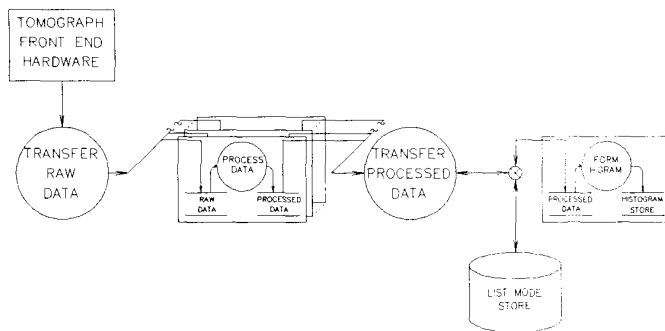


Figure 1: Data Acquisition System Context Diagram

The function ‘Process Data’ is allocated to the parallel array of processors. This function is shown in detail Figure 2. The function *decode event* takes as input the raw event record containing the detector coordinates of each end of the annihilation line, time varying tomograph parameters, such as gantry position and physiological data, and fixed tomograph parameters, such as lookup tables for detector and wobble positions. For each of the pair of gamma rays, ‘decode event’ produces the corresponding detector position in the tomograph ring $(x_d, y_d, z_d)_i, i = 1, 2$. Two identical transformation functions *xform data* each take as input one of $(x_d, y_d, z_d)_i$ and, using the tomograph parameters¹ transform these coordinates to coordinates in a fixed frame of reference. The transformation produces as

processed event lists must be written to the list mode store in the same order that the raw event lists were written to the parallel raw data stores. A detailed discussion of this mode is found in [13].

For a PVI tomograph using a particular three dimensional image reconstruction algorithm[5], the four dimensions of the histogram are $\theta, \phi, L_x,$ and L_y . The angles θ and ϕ identify the orientation of the projection plane with respect to the fixed frame of reference (x, y, z) of the tomograph. The coordinates, (L_x, L_y) are the coordinates of the point of intersection of the annihilation line with its projection plane, in the coordinate system of the projection plane.

For an SO tomograph or a PVI tomograph using a two dimensional image reconstruction algorithm, the histogram coordinates are

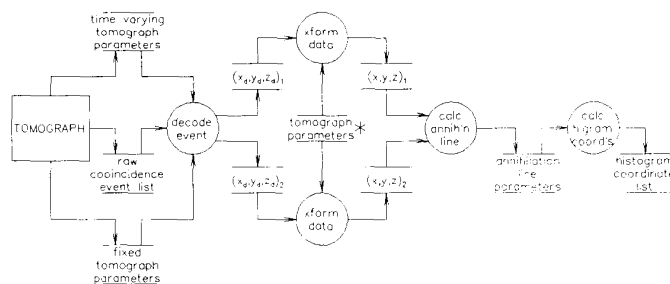


Figure 2: Data Transformation Engine

output the coordinates $P_1 = (x, y, z)_1$ and $P_2 = (x, y, z)_2$ of the two ends of the annihilation line in the fixed frame of reference established for the complete tomograph. The function *calc annih'n line* (calculate annihilation line) then uses the coordinates of the two ends of the annihilation line, $P_1 P_2$ to calculate the annihilation line parameters. Finally, the function *calc h'gram coords* (calculate histogram coordinates) uses the annihilation line parameters to produce a histogram coordinate which is added to the list in the output store *histogram coordinate list*. The histogram may be either the sinogram, the four dimensional projection data set[5,8] or some other histogram[6,7].

In Figure 1 the function ‘Transfer Processed Data’ takes as input the processed event lists in the output stores of the parallel ‘Process Data’ processes and transfers the processed data to a list mode store or to one of several input stores of the histogramming process ‘Form H’gram’. If simultaneous list and histogram mode is selected, when the histogramming of a processed data list is complete, the function ‘Transfer Processed Data’ transfers the processed data list from the input store of the histogram process to a list mode store. A major difference between list mode and histogramming mode is the fact that, if chronological order of the events is to be preserved, the

the (r, θ) values of the sinogram within a given slice. The slice number forms the third coordinate. The computational burden in producing these sinogram coordinates is considerably less than that associated with the production of the projection data set required for three dimensional image reconstruction. Thus, the DTE rate handling ability is expected to be considerably higher for SO than for PVI data for a given implementation.

3 System Architecture

As previously mentioned, Figure 1 shows the essential functions of the DAS. Figure 3 shows the DAS hardware.

3.1 Software

The software is configured as one master process controlling data transfers and data processing, with n_{fep} slave front end processors (FEPs) processing data asynchronously and in parallel.

FEPs logically share memory, but there are definite performance advantages to accessing local memory (which can be as large as 16 Mbytes).

While processing continues, data can be transferred in and out of a FEP’s local memory by DMA data transporters (see Figure 3). Thus, in order to maximise the time available for processing, double

¹wobble, rotation, or clamshell position; and detector position. This information is contained in the store ‘tomograph parameters’. This store represents both the fixed and the time varying tomograph parameters shown elsewhere in the diagram. The asterisk indicates that it is a duplicate of one or more store(s) shown elsewhere in the diagram.

buffering is used for each FEP. The master FEP controls the buffer transfers, using a master list. The master/slave software is described in detail in [13].

There are several logically concurrent threads of control in the master FEP. In order to reflect these parallel activities, the software in the master FEP is designed as a multi-tasking system, with three main tasks: Control Data Processing, Control Input Transfer (raw events), and Control Output Transfer (processed events). The master FEP software can run as an application program above a multi-tasking real time kernel such as pSOS[14]; however, in our development environment the master runs as a process above the UNIX operating system. All the code is written in the high level language C for portability; the compiler's optimising features were used to improve the data processing times.

3.2 Hardware Data Transport

The Ironics IV-3272 full speed data transporter is particularly well suited for use in the DTE. It provides a well defined 32 bit i/o bus upon which a maximum of 14 data transporters may be resident. Each data transporter is interfaced to the bus by an on board 512 by 32 bit FIFO and is capable of i/o data transfer rates of 40 Mbytes/second[15]. Thus, the data transporter is capable of accepting tomograph data at peak rates of 40 Mbytes/second. In transferring data from the i/o bus to VMEbus, the data transporter can maintain a sustained data transfer rate of over 24 Mbytes/second and peak data transfer rates of over 30 Mbytes/second[16] into the local 100 nanosecond dynamic random access memory of an Ironics processor board² using static column block transfer mode[17]. An additional useful feature of the IV-3272 is the on-board TMS 320xx digital signal processing chip which may be programmed[9] to pre-process the data in real time before transfer over the VMEbus.

3.3 Front End Processors

The three control functions of the master FEP have been allocated to the single processor which is labelled 'DAS CTRLR' in Figure 3.

The function 'Process Data' is allocated to each of the parallel array of slave FEPs. In order to maintain PVI event throughput with a manageable number of slave FEPs, the data transformations required will probably necessitate the use of third generation RISC processors such as the Advanced Micro Devices AMD29000 or the Motorola 88000. However, pending the purchase of VME boards based on these RISC processors, the data processing code is being

²the Ironics IV-3201A

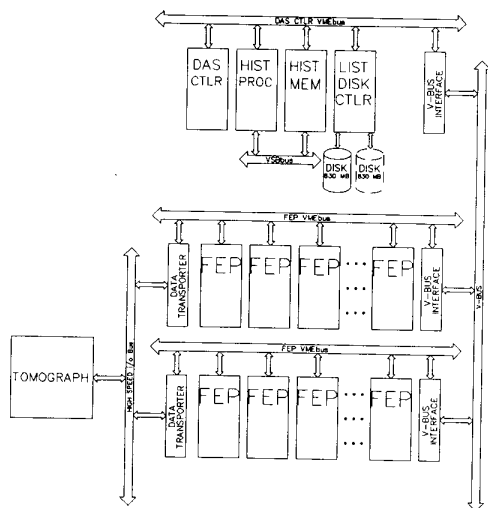


Figure 3: Data Acquisition System Block Diagram

developed on a Motorola 68020 based VME board. For this reason, the data processing code is written in C so it is transportable, standalone and independent of the software and hardware environment of the target processor.

4 Development System

A prototype data acquisition system based on a single VME backplane has been assembled and is shown in Figure 4. The main components of this system are a 16 MHz Motorola 68020/68881/MMU UNIX engine (the IV-3204), two standalone 20 MHz Motorola 68020/68881 processors (the IV-3204A and the IV-3201A), a data transporter (the IV-3272) and a 16 Mbyte VMEbus memory card (the MM6230[18]). An interface has been constructed[19] which connects the i/o bus of the data transporter to the ECL bus of the Lecroy 4300 Fast Encoding and Readout ADCs in CAMAC[20]. The prototype system is networked to the TRIUMF VAX cluster using an Ethernet.

A PVI data transformation algorithm[5] has been implemented on one of the standalone processors in order to measure the processing time. This data transformation takes as input a list of simulated raw tomograph coincidence events produced by a Monte Carlo code[2] and produces as output a list of four dimensional projection set coordinates.³ The processing time on 200,000 events was measured as 350 microseconds per event.

5 Performance Analysis

The performance of the DTE has been modelled in order to estimate the expected event rate as determined by the data transfer rate, the parallel data processing rate, and control/communication overheads.

A significant parameter of the DTE is the effective processing time per event, t_{ep} ⁴. For n_{fep} parallel FEPs, the effective DTE processing time is:

$$t_{ep} = \frac{t_w + t_p + t_r}{n_{fep}} \quad (1)$$

where t_w is the effective time taken to write one raw event into the FEP memory, t_r is the effective time taken to read one processed event from the FEP memory and t_p is the FEP cpu time taken to process one event, measured as 350 microseconds per event on our development system. Based on published performance figures, the processing time for this algorithm is expected to improve to $t_p = 70$ microseconds on the 25 MHz Ironics IV-9001 VMEbus Single Board Super Computer[21] which uses the AMD29000 RISC processor with an AMD29027 floating point coprocessor.

³When this software was transferred to a VAX computer, only one modification was required for it to compile and execute: a floating point '1' had to be changed to floating point '1.0'.

⁴the per event data transfer time is a small fraction of the PVI processing time, but may become significant for other applications.

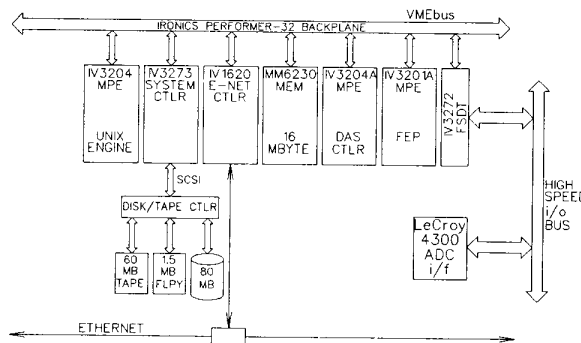


Figure 4: Development System

A model of the DTE performance is derived in [22]. This model estimates the event throughput as a function of raw and processed event size, raw and processed event block size, bus arbitration time (including the effect of a device's position in the arbitration daisy chain), effective VMEbus read/write cycle times, Vertical bus arbitration time, Vertical bus signal propagation delay and control/communications overheads. Most of these factors are amortized over the transfer of a block of events and have little effect on the available bandwidth. However, VMEbus effective read cycle time, VMEbus effective write cycle time and Vertical bus propagation delays may affect each data transfer cycle and therefore one or more of these factors form the primary limitation to data transfer rate.

The DTE histogram mode event rate with a single histogramming engine on the DAS CTRL VMEbus, R_h , is estimated as:

$$R_h = \frac{1}{t_c + t_{vi} + t_{ep} + t_{vo}} \quad (2)$$

and the DTE histogram mode event rate with one histogramming engine per FEP backplane, R_{ph} , is estimated as:

$$R_{ph} = \frac{1}{t_c + t_{vi} + t_{ep} + t_{vvo}} \quad (3)$$

where t_c is the control/communication overhead per event, t_{vi} is the effective raw event transfer time from the data transporter to a FEP, t_{vo} is the effective processed event transfer time from a FEP to a single histogramming engine located on the DAS CTRL backplane and t_{vvo} is the effective processed event transfer time from a FEP to a histogramming engine located on the same backplane.

For an Ironics IV-3272 Full Speed Data Transporter using static column block transfer mode to transfer blocks of 64 bit raw events and 32 bit processed events, the model obtains the following values:

$$t_{vi} = \frac{340}{n_{bp}} \text{ nsec/event} \quad (4)$$

$$t_{vvo} = \frac{370}{n_{bp}} \text{ nsec/event} \quad (5)$$

$$t_{vo} = 560 \text{ nsec/event} \quad (6)$$

$$t_c = 31 \text{ nsec/event} \quad (7)$$

where n_{bp} is the number of FEP backplanes.

Since the calculation of t_{vo} includes a Vertical bus cycle on each data transfer, it is much larger than t_{vvo} . If multiple FEP backplanes are employed, it is advantageous to place a histogramming engine on each FEP backplane as this allows the parallel transfer of processed events to histogramming engines. Thus, in implementing high data throughput, it is preferable to have one histogramming engine per FEP backplane with the separate histograms (one for each FEP backplane) being combined when the data rate drops during a run or upon completion of a run.

In Figure 5, R_h is plotted for a three FEP backplane system using the PVI algorithm value of $t_p = 350 \mu\text{sec/event}$ measured on the 20 MHz M68020/M68881 standalone processor. R_h is also plotted for a two FEP backplane system using the PVI algorithm value of $t_p = 70 \mu\text{seconds/event}$ projected for the AMD29000/AMD29027 RISC processor.

Since SO processing algorithms require less processing time than PVI algorithms, Figure 5 also includes plots of R_{ph} for processing times $t_p = 20 \mu\text{sec/event}$ and $t_p = 50 \mu\text{sec/event}$.

Figure 5 shows that, for this particular PVI processing algorithm[5], 40 Motorola 68020/68881 FEPs would maintain a PVI coincidence event rate of about $R_h = 100$ kevents/sec, whereas 30 AMD 29000/29027 RISC front end processors would maintain a PVI coincidence event rate of about $R_h = 350$ kevents/sec. For $t_p = 50 \mu\text{sec}$, 30 FEPs would maintain a coincidence event rate of $R_{ph} = 480$ kevents/sec and, for $t_p = 20 \mu\text{sec}$, 30 FEPs would maintain a coincidence event rate of $R_{ph} = 950$ kevents/sec.

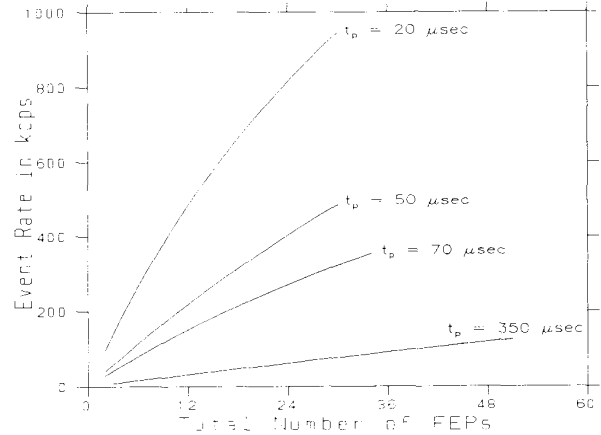


Figure 5: Data Transformation Engine Simulated Performance

6 Concluding Remarks

This is a parallel real-time system design with many asynchronous events and activities and capable of processing hundreds of thousands of events every second. Until recently, this could only be done in hardware.

The system offers several attractive features to the tomograph designer:

- software controlled processes are moved closer to the front end of the system. This provides flexibility in data structure and hardware configuration as compared to hardware based data acquisition systems.
- the flexible input event list data structure allows optional auxiliary data records to be inserted into the coincidence event list and varying sizes of raw and processed event records.
- the real time parallel processing of the raw events to processed events reduces the list mode storage requirement and data transfer rate by a factor of two without degrading event throughput.
- the parallel architecture is open-ended and scalable. This enables the use of software and hardware development systems based on a few modules which can, when required, be scaled up to obtain the desired event acquisition rate by adding more modules.
- the software architecture is portable. Thus, the systems engineer can choose among a broad range of commercially available modules.

References

- [1] T J Spinks, T Jones, M C Gilardi, and J D Heather. Physical performance of the latest generation of commercial positron scanner. *IEEE Transactions on Nuclear Science*, NS-35(1):721-725, 1988.
- [2] J G Rogers, R Harrop, P E Kinahan, N A Wilkinson, P W Doherty, and D P Saylor. Conceptual design of a whole body PET machine. *IEEE Transactions on Nuclear Science*, NS-35(1):680-684, 1988.
- [3] G Muehllenhner, J S Karp, D A Mankoff, D Beerbohm, and C E Ordonez. Design and performance of a new positron tomograph. *IEEE Transactions on Nuclear Science*, NS-35(1):670-674, 1988.
- [4] C A Burham, D E Kaufman, D A Chesler, C W Stearns, D R Wolfson, and G L Brownell. Cylindrical pet detector design. *IEEE Transactions on Nuclear Science*, NS-35(1):675-679, 1988.

- [5] P E Kinahan, J G Rogers, R Harrop, and R R Johnson. Three-dimensional image reconstruction in object space. *IEEE Transactions on Nuclear Science*, NS-35(1):635-638, 1988.
- [6] M E Daube-Witherspoon and G Muehlelehner. Treatment of axial data in three-dimensional PET. *Journal of Nuclear Medicine*, 28(11):1717-1724, 1987.
- [7] M Defrise, S Kuijk, and F Deconinck. A new three dimensional reconstruction method for positron cameras using plane detectors. *Physics in Medicine and Biology*, 33(1):43-51, 1988.
- [8] J G Colsher. Fully three-dimensional positron emission tomography. *Physics in Medicine and Biology*, 25(1):103-105, 1980.
- [9] Ionics Incorporated. *IV-3272 DMA Controller/Digital Signal Processor User's Manual*, February 1988.
- [10] Ciprico, Incorporated. *Rimfire 3200 Intelligent VMEbus SMD-E Disk Controller*, 1988. (Manufacturer's Literature).
- [11] Interphase Corporation. *V/SME 4200 Cheetah High-performance VMEbus Storage Module Device (SMD) Disk Controller Evaluation Reference Guide*, 1986.
- [12] D Allmond. DMA controller saves bus bandwidth from drowning. *ESD: The Electronic System Design Magazine*, pages 79-83, August 1987.
- [13] N A Wilkinson, M S Atkins, and J G Rogers. A real time parallel processing data acquisition system. In *Ninth Real-Time Systems Symposium*. Huntsville, Alabama, December 6-8, 1988.
- [14] Software Components Group, Incorporated. *pSOS-68K Real-Time, Multi-processing Operating System Kernel User's Manual*, May 1987.
- [15] Ionics Incorporated. *IV-3272-PIO Parallel Interface Daughter Board for the IV-3272 Full Speed Data Transporter*, April 1988.
- [16] Ionics Incorporated. *IV-3201A VMEbus Multiprocessing Engine User's Manual*, February 1988.
- [17] Ionics Incorporated. *Static Column Block Transfer Mode (SCBLT)*, January 1988. (technical summary).
- [18] Micro Memory Incorporated. *MM6230D User's Guide*, June 1987.
- [19] N A Wilkinson, B E Evans, and D Paranchyk. *FTOOLS: High Speed Data Acquisition on VMEbus*. TRIUMF Tomograph Group Technical Report, August 1988.
- [20] LeCroy Research Systems Corporation. *CAMAC Model 4300B, 16 Channel Fast Encoding and Readout ADC (FERA) User's Manual*, February 1986.
- [21] Ionics Incorporated. *IV-9001 VMEbus Single Board Super Computer User's Manual (Preliminary)*, September 1988.
- [22] N A Wilkinson. *A Performance Model of a Parallel Processing Data Acquisition System using VMEbus*. TRIUMF Design Note, November 1988.