PNL-SA-21926

ACQUIRE: A DATA ACQUISITION SYSTEM FOR CAMAC ON SUN WORKSTATIONS

R. T. Kouzes
M. M. Lowry

June 1993

Presented at the
8th Conference on Real-Time
Computer Applications in Nuclear
Particle & Plasma Physics
June 8-11, 1993
Vancouver, British Columbia

Prepared for
the U.S. Department of Energy
Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Richland, Washington 99352

ACQUIRE: A Data Acquisition System for CAMAC on SUN Workstations

Richard T. Kouzes
Pacific Northwest Laboratory,† MS K1-87, PO Box 999, Richland, WA 99352
M.M. Lowry
Princeton University, PO Box 708, Princeton, NJ 08544

Abstract

The data acquisition software package ACQUIRE has been used for many years by the Princeton University Cyclotron Laboratory for nuclear physics research applications. This code has been ported to the SUN Sparc workstation and is fully functional, including block data transfers using an in crate Event Handler. A SCSI interface to CAMAC is utilized, and the device handling software has been developed in such a way that little modification was needed in the ACQUIRE code for the SUN implementation. The Higz X windows graphics package from CERN is used for data display.

I. INTRODUCTION

Data acquisition for high-rate, multiparameter physics experiments has, until recently, relied on minicomputer systems to obtain the performance and flexibility necessary to meet the experimental needs. The price performance of RISC workstations makes them attractive for real-time applications, even though the Unix operating system is not optimal for this application. The emerging versions of Unix that have Posix real-time extensions will make future real-time applications much faster.

The data acquisition software package ACQUIRE [1-4] has been used for many years by the Princeton University Cyclotron Laboratory for nuclear physics research applications. ACQUIRE utilizes CAMAC data acquisition modules, including the option of intelligence in the crate to gather the data for DMA transfers to the host computer. ACQUIRE was originally developed for Harris computers, migrated to 16 bit Data General Eclipse computers and then to 32 bit Data General systems under the AOS/VS operating system. An implementaion was also made on IBM/RT computers under the AIX operating system.[5]

The ACQUIRE code was developed for nuclear physics applications, characterized by relatively high continuous data rates of a few parameters with real-time sorting of the multiparameter events. The general nature of this code gives it possible applicability in other environments such as in molecular chemistry measurements. ACQUIRE was designed with the philosophy that it operate on standard computers with standard operating systems and that it provide for virtually all experimental needs without the user developing any code. This means that only one code must be maintained, enhanced, and debugged. It has moderate to high data throughput capability and provides data display and analysis capability. The code is written in Fortran 77.

ACQUIRE has been ported to SUN Sparc workstations and is fully functional. For this implementation, a SCSI interface to CAMAC is utilized. Both the Kinetic Systems and Jorway SCSI to CAMAC crate controllers are supported. The software that has been developed includes a generic SCSI device interface (sxm.c) to the operating system driver and a subroutine interface from the user application to sxm (Jamac or Kamac depending on the controller used). The device handling software has been developed in such a way that little modification was needed in the ACQUIRE code for the SUN implementation. ACQUIRE was upgraded in a number of ways to enhance its performance for the SUN implementation.

The Higz X windows graphics package from CERN is used for data display, which required modifications to the code since the previous platform utilized a Tektronics terminal emulation for graphical display. Higz is used because it is a publicly available library that allowed for an easier translation from the old graphics calls to the new ones than would a direct X implementation. The only compromise came in implementing the graphical cursor input mouse clicks, where an extra mouse action is now required when selecting from a graphical menu. A small library of subroutines was written to provide a Plot10-like calling set to the Higz routines, which provided for rapid implementation of the ACQUIRE code requirements.

A number of other changes to handle multitasking aspects of the code were required. The fork system call provides the mechanism to spawn processes used as timing, event recording, and polling loops. Common block data is shared through calls to the mmap system call, which also requires aligning the blocks at load time. Process communication is carried out through memory flags, as well as the signal, kill, pauses, and wait system calls. System calls are easily made from Fortran.

The system runs under the Sparc standard SUN OS operating system. The SUN has a highly useable development environment in the Openlook windows interface. We expect that the performance of multiprocessing will be improved when the Posix real-time extensions become available in the future Solaris operating system on the SUN.

II. ACQUIRE OPERATION

ACQUIRE assigns two megawords of array space for up to 40 histograms (one and two dimensional) from up to 64 data sources in CAMAC (e.g., ADCs, TDCs, scalers, transient recorders, histogramming memories). Multiple condition cuts can be placed on data histogramming. Data values can be mathematically manipulated and combined to produce new pseudodata values for sorting, such as for position determination from resistive division counters or position correction of scintilator signals. There are several default modes of data sorting that set up the array parameters and conditions for commonly used operations. The *Singles* operation mode of sorting is used for a mixture of one and two-dimensional arrays, with a default of eight 1024 word arrays associated with ADC sources one to eight. The *Spectrograph* mode of sorting was designed for use with the QDDD magnetic spectrograph, and can handle resistive division detectors. This mode defaults to a particle ID, plus eight one-dimensional histograms associated with two-dimensional particle cut windows. Figure 1 shows a particle ID typically used for spectrograph mode operation. A mode for solid state telescopes is also provided. Additional data taking modes have been added for special experiments such as parity violating beta decay studies.

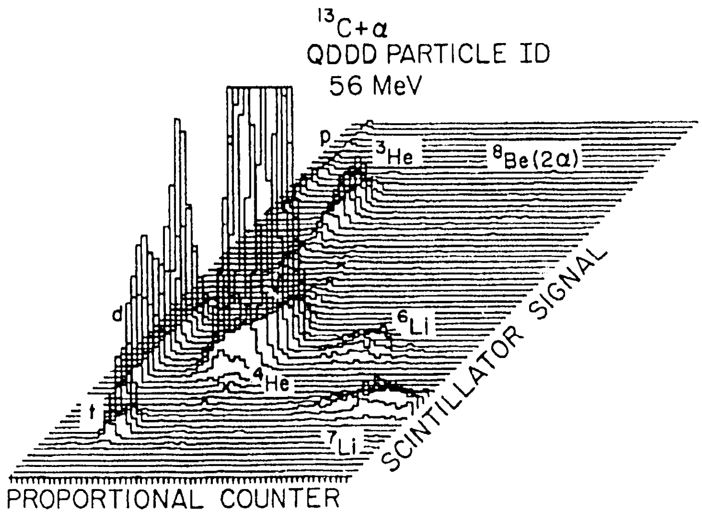

Figure 1: A two-dimensional particle identifier as displayed by ACQUIRE.

The program is command driven (over 70 commands are recognized), with a set of default parameters provided for simple setup of data acquisition. Table I shows a typical command sequence.

Table I: A typical command sequence for taking data with ACQUIRE.

Program	User	Comment
COMMAND:	singles	Set singles data taking mode
COMMAND:	dp,acqdap.413	Set the DAP file name
COMMAND:	acquire	Start data acquisition
COMMAND:	scope,1	Display histogram 1 in a graphics window
COMMAND:	stop	Stop data acquisition
COMMAND:	ps,1	Output a postscript file of the histogram 1
COMMAND:	ma,junk	Assign disk file "junk" as the output (magtape) file
COMMAND:	dump	Write all the histograms and parameters to the disk

ACQUIRE can control and receive data from any CAMAC module. The description of the CAMAC modules used and their function codes are contained in a text file that the user prepares, known as a DAP file. This resource file provides all the information needed by the program for initiating data acquisition. For low speed requirements, the CAMAC controller is polled by the code for data pending. For high-speed applications, an Oak Ridge Event Handler is used to gather the data from various modules and place it into a FIFO buffer in the crate for block transfer to the host. The code that runs in the Event Handler is built by the ACQUIRE application and downloaded when data acquisition is initiated. We plan to support a list processor module as an alternative for in-crate control.

Figure 2 shows the overall structure of ACQUIRE. The code consists of a command interpreter that calls the appropriate subroutine for action. When data acquisition is initiated, a process is spawned to handle the polling and DMA actions. Other processes may also be spawned for event mode recording, timed acquisition, automatic data backup, and display update. Event data can be read and sorted in the same manner as online sorting, and a revised event tape can be written.

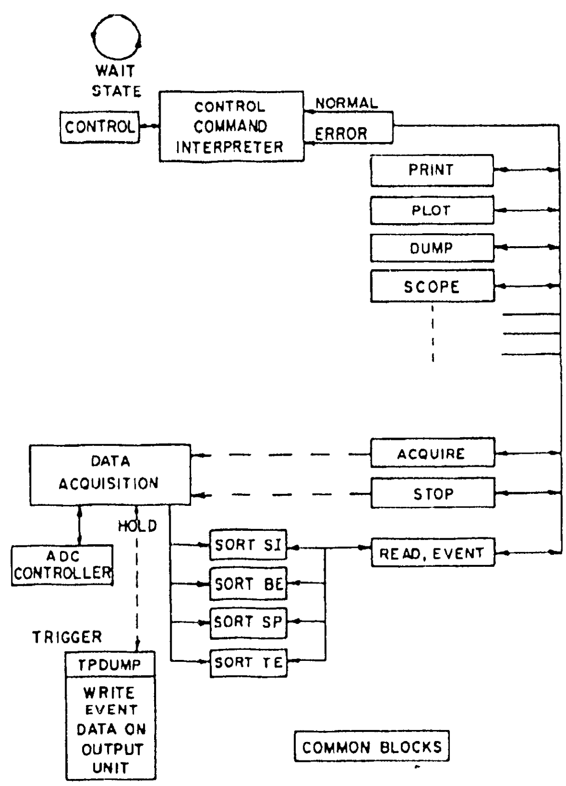

Figure 2: Acquire subroutine and data flow structure.

III. SCSI to CAMAC Implementation

In order to implement ACQUIRE on a SUN SPARCstation, it is necessary to provide communication with a CAMAC crate controller. The integral SCSI Bus Adapter provides a convenient solution to this requirement since at least two manufacturers offer SCSI-bus-based CAMAC crate controllers. Most peripherals (disks and tapes) are mounted via this bus so SUN requires adherence to the SUN Common SCSI Architecture (SCSA) programming interface, which enforces a strict separation and encapsulation of the various tasks needed to communicate over the SCSI bus. The interface paradigm is that of a user program requesting I/O of a "target driver" that prepares a SCSI command packet and DMA vector (if required). This packet is then passed to the "host driver" supplied in the operating system, which obtains the necessary DMA resources and accesses the bus adapter to transport the command packet (and receive or send data) to the "target", the CAMAC controller in this case. Unfortunately SUN does not implement the Asynchronous Event Notification (AEN) part of the SCSI-2 protocol; thus, no crate-initiated communication (for a LAM) can occur. For a nonstandard device like a CAMAC crate controller, both a user program and a system-kernel-resident target driver must be provided and we have developed both. One final wrinkle is the question of adding the target driver to the system kernel. Under SunOS 4.1.x, there are in general two mechanisms: 1) the driver is added by rebuilding the kernel and then re-booting with the new kernel or 2) the driver is loaded into a running system and takes the place of a virtual driver. Although the SCSA does not provide for the second alternative, we have successfully implemented this procedure for our target driver.

The user program (ACQUIRE) communicates with the target driver (SXM) solely via the *ioctl()* system call. Five functions are available:

- *SXM_DMAMAX* returns the maximum size DMA transfer that the host adapter can perform. The system limit is usually more restrictive.
- *SXM_RESET* forces either a SCSI bus or a SCSI device reset.
- *SXM_RD* performs a SCSI command with data read in.
- *SXM_WRT* performs a SCSI command with data written out.
- *SXM_CMD* performs a SCSI command without data transfer.

The last three commands take as an input the address of a structure containing the SCSI command packet, an array for the return of target driver and host adapter diagnostic information, the length and address of any target device generated diagnostics (a Request Sense), and the length and address of the data buffer. This very generic architecture allows both of the tested crate controller types to be driven by the same driver. A simple busy status lockout allows only one process at a time to access a particular crate. Should the status byte returned by the given SCSI command have the CHECK CONDITION bit set, the target driver will perform a second command, a REQUEST SENSE, to obtain crate controller diagnostics (such as bad Q). This must be done immediately to prevent another process from interposing a command before the REQUEST SENSE and invalidating the diagnostics. This is the only form of error recovery attempted by the target driver. It is up to the user program to take appropriate action based on the returned diagnostic values because both the interpretation of the diagnostics and the action depend on the type of crate controller. Similarly the SCSI command packet for a given NAF is also a function of manufacturer and is left to the user program.

Although the SCSI-2 bus is capable of transfers at up to 10 Megabytes per second (maximum of 2 or 5 MB/s with the controllers considered here), the significant overhead of the protocol means these high rates are only available for large block transfers. For many real-time applications, a processor in the CAMAC crate filling a large buffer will be required to achieve acceptable dead time. Utilizing such an event handler filling a 4 kilo-word FIFO, we have obtained data taking speeds of 40 kHz for a single-parameter acquisition and 20 kHz for four parameter data acquisition. This is a higher throughput rate than that obtained with the previous Data General implementation, and is expected to improve as the OS and application software evolve.

IV. CONCLUSIONS

The present system will be used as the Princeton Cyclotron Lab data acquisition system for nuclear physics experiments to replace the Data General sytem that has been in use for many years. ACQUIRE has the appropriate data taking modes for this application and can be enhanced as future needs arise.

The ACQUIRE system will also play a role in the Molecular Sciences Research Center at Pacific Northwest Laboratory. As part of the Environmental and Molecular Sciences Laboratory now under development at PNL, this software has been brought up for testing of CAMAC-based systems and as a prototype for future Unix-based data acquisition. The prototyping effort will continue as the EMSL instrumentation takes on its final form.

ACQUIRE has provided for almost all of the data acquisition needs at the Princeton Cyclotron Laboratory for over 15 years, and has migrated across several platforms. The latest implementation on SUN workstations will continue its usefulness for nuclear physics, as well as in molecular science applications.

V. REFERENCES

[1] R.T. Kouzes, "Minicomputer Data Acquisition System for Nuclear Physics Applications," NIM **155** (1978) 261.
[2] R.T. Kouzes, M. Miller, "The Princeton Cyclotron Data Acquisition System," NS **26** (1979) 4378.
[3] R.T. Kouzes, P.C. Nelson, "An application of Microprogramming to Nuclear Physics Data Acquisition," NS **27** (1980) 1363.
[4] R.T. Kouzes, M. Karlsson, "Data General 32 Bit Computer Based Acquisition System at the Princeton Cyclotron," NS **32** (1985) 1363.
[5] R.T. Kouzes, M.M. Lowry, "IBM/RT Based CAMAC Data Acquisition Station," NS **34** (1987) 924.

†Pacific Northwest Laboratory is operated for the U.S. Department of Energy by Battelle Memorial Institute under contract DE-AC06-76RLO 1830.

END

DATE
FILMED
9 / 30 / 93