IEEE Transactions on Nuclear Science, Vol. NS-26, No. 4, August 1979

# The Crystal Ball Data Acquisition System

R. Chestnut, C. Kiesling, E. Bloom, F. Bulos,
J. Gaiser, G. Godfrey, M. Oreglia
SLAC Stanford University

R. Partridge, C. Peck, F. Porter
California Institute of Technology

D. Aschman, M. Cavali-Sforza, D. Coyne, H. Sadrozinski,
Princeton University

W. Kollmann, M. Richardson, K. Strauch,
Harvard University

R. Hofstadter, I. Kirkbride, H. Kolanoski,
A. Liberman, J. O'Reilly, J. Tompkins,
Stanford University, HEPL

T. Burnett
University of Washington, Seattle

## ABSTRACT

The data acquisition system for the Crystal Ball project at SLAC is described. A PDP-11/t55 using RSX-11M connected to the SLAC Triplex is the basis of the system. A "physics pipeline" allows physicists to write their own equipment-monitoring or physics tasks which require event sampling. As well, an interactive analysis package (MULTI) is in the pipeline. Histogram collection and display on the PDP are implemented using the Triplex histogramming package. Various interactive event displays are also implemented.

## INTRODUCTION

The Crystal Ball is a non-magnetic detector system with a large solid angle acceptance. It emphasizes the complete detection and precise measurement of energy depositions of all particles produced in positron-electron annihilation events. The detector has five major components:

1. A compact 672-segment NaI detector of 16 radiation lengths which is almost spherical and covers 94% of the total solid angle.

2. A set of cylindrical magneto-strictive spark chambers and multi-wire proportional chambers around the beam pipe to define charged particle trajectories.

3. End caps of magneto-strictive spark chambers, and mildly segmented (60 segments) NaI crystals closing the solid angle to 98% of the full sphere.

4. A multicounter luminosity monitor to measure the beam luminosity to about 2%.

5. A muon-hadron selector consisting of proportional tube arrays sandwiched between several layers of steel plates covering about 15% of the solid angle.

The detector system was designed and built by the Caltech, Harvard, Princeton, Stanford, SLAC collaboration and is now running this experiment at SPEAR (Stanford Positron Electron Asymmetric Rings).

An online-computer system was planned as an integral part of the experiment. The design goals of the data acquisition system were:

Flexible computer access to the hardware using CAMAC,

Read-out and logging of data onto tape,

Provide a diagnostic tool so that any physicist in the group could easily monitor some portion of the equipment,

Deliver an accurate overview of the status of the detector and electronics at any time,

Do as much physics analysis online as possible to efficiently monitor the experiment's progress.

## HARDWARE

A PDP-11/t55 computer with the full compliment of core, including 64kBytes of fast core and a fast floating point arithmetic unit was chosen to meet these goals. This configuration is ten percent faster than the similarly equipped PDP-11/70, according to Digital Equipment Corporation benchmarks. Peripheral devices include two 1600 b.p.i magnetic tape drives, an RK07 (28 Megabyte), an RK06 (14 Megabyte) and two RK05 disk drives, a Versatec printer/plotter, three Tektronix 4013 displays and an IBM System/7 which serves as a link to the

SLAC Triplex (two IBM 370/168 and one IBM 360/91 computers loosely coupled). The external electronics are connected to the online computer via CAMAC. The CAMAC controllers are mounted in a Unicrate (a modified CAMAC crate with a DEC Unibus back plane), connected to the UNIBUS of the PDP-11 and communicate with the 13 CAMAC crates imbedded in the electronics. The controllers in use are ARBOLA (6), which allows all legal CAMAC commands, LABRI (1), which services the SLAC priority encoder module (2), and MAESTRO, which services the special spark chamber readouts. As well the SPEAR CRT CAMAC memory is used for graphics.

## SOFTWARE

### System

The DEC system RSX-11M v3.1 (4) is the basis of the data acquisition system. This is a priority-driven, task-oriented system. In our system, all external devices are accessed via loadable drivers, including the sophisticated CAMAC interface and the System/7 fast data link. Memory partitions have been so defined that the RSX-11M system itself, the Fortran resident library, all drivers, necessary systems partitions and common blocks occupy the lower 64K words of memory. Fixed data-acquisition tasks (those always waiting for an interrupt) occupy a further 4K, leaving 56K for active data acquisition tasks. The lower 32K words of memory are fast core; here reside the system, the fortran resident library, the camac driver, and the data I/O task. Deadtime minimization was the primary concern when the decision was made on how to parcel out the fast core. With few exceptions, all data acquisition tasks are written in Fortran. The DEC Fortran-4 Plus compiler generates nicely optimized code, so that further optimization becomes unprofitable. As well, program compatibility with subroutines developed for offline analysis on the SLAC Triplex is desirable. Program development can take place during data acquisition and degrades the effectiveness of the latter only marginally if the priority of the system utilities lies below that of the data acquisition system.

### CAMAC Driver

All CAMAC functions are included in the driver, which adheres religiously to DEC specifications for an I/O driver. The QIO operations included are classified as follows:

1) Non-interrupt data and command transfer.
Programmed Data Transfer(PDT) and Direct Memory Access(DMA)
I/O packets are queued and the task waits on completion;
The task is generally checkpointable;
The PDT may be a long list of CAMAC commands.

2) Connect to Interrupt (CCI)
Issue one I/O packet to get permanently connected to interrupt.
Task must henceforth only wait for flag;
Task is non-checkpointable.

3) Connect to Interrupt (ICN)
Issue one packet to get permanently connected to interrupt
Task must henceforth only wait for flag;
Task is checkpointable (IOC=0);
(slower than CCI)

4) Connect to Interrupt and CAMAC data and/or command transfer (CPT,CDM)
Perform action an interrupt, set flag;
Task is non-checkpointable;
Two packet technique.

5)Special Calibration QIO (burst QIO)
One I/O packet performs 'n' DMA's
Can service 2 kHz interrupt rate.

6)Unsolicited interrupt killer
clear and disable goofy lams
(needed because of Hardware latch, priority encoder)

The 'Connect to Interrupt' functions are necessary as well as cute. The mean time from interrupt to service after having connected to an interrupt is 150 micro-seconds, as opposed to 1.5 milli-seconds for an I/O packet to be issued. A given task may connect to up to 12 interrupts at one time.

### CBOLS

#### System Structure.

The data acquisition system itself (CBOLS - Crystal Ball OnLine System) is, in effect, a multi-level dispatcher oriented system. The Input/Output runs at very high priority (245 of a maximum 250) using a double buffer to overlap input and output. This task reads the approximately 1500 16-bit parameters from CAMAC, and writes them to tape and/or the System/7. Large blocks of data are passed via common blocks; global flags are used for task synchronization.

#### User Interface.

A 16-word 'button board', consisting of a collection of switches, thumb wheels, and LAM-producing buttons is used to control the experiment. Here the experimental configuration, I/O configuration, pipeline configuration, type of event display and histogram display selection are determined, as well as functions such as START RUN, END RUN, STOP, CONTINUE, etc.

#### Pipeline.

The pipeline operates on a sampling basis and consists of a pipeline driver and a series of analysis tasks. The first pipeline task converts the digital data to energies; then if the data represents a physics event (as opposed to a Xenon-flasher calibration event or an event triggered by the luminosity counters), it finds connected energy regions and tags tracks as charged or uncharged. The pipeline driver then sends non-physics data to the proper monitoring task (if running), and physics data sequentially to selectable tasks (PHYS01, PHYS02 ...), which are physicist-written and can be diagnostic tasks, test tasks, or physics tasks. They can also be aborted, re-run, replaced or may crash without affecting the integrity of the system. Each task in the pipeline sets a global flag to indicate to the pipeline driver that its function is complete. Should the task crash or otherwise 'time out', a control task, which is scheduled to run 10 seconds after a given task is posted by the pipeline driver, runs and sets the proper flags, preventing a deadlock situation. If the given pipeline task runs successfully, the request to run the control task is cancelled. This mechanism does add a bit of overhead; it can be shut off if the system is running stably.

At the end of the pipeline the MULTI system, which is described in another paper at this meeting, and an event display driver are implemented. MULTI can be used to do complex analysis via user provided subroutines, but our use is limited to looking at components of the system which seem to be acting up at any one time, taking advantage of the interactive nature of MULTI.

This pipeline philosophy has proven extremely flexible and almost indispensable for development work. The testing of new routines is non-destructive and new ideas are quickly implemented. Our stable running mode has become one where only one or two well-tested physics tasks run which fit together in available core, minimizing disk I/O and system overhead. With such a configuration we get a total sampling rate of around 44% at an event rate of 2.5/second.

Event Displays.

The display driver at the end of the pipeline dispatches one of eight different display tasks. The data necessary for the display tasks is copied from the general common area and saved on a disk file, quickly decoupling the display subsystem from the pipeline. The event displays can be taken in and out of the pipeline or 'held' for a given event, which can then be examined in more detail. Any of four displays can be routed to either the SLAC scope or to the Versatec printer/plotter. The available displays are:

1) X-Z and X-Y projections of the aparatus and all tracks.

2) A rough overview of energy in all crystals (flattened orange peel).

3) A blown-up view of one track or set of crystals.

4) A view of the end caps.

All of the graphics programs in the system use the SLAC Unified Graphics system (7), wherein graphics elements are first prepared in a device independent manner. Only upon output are these graphics elements transformed into device specific commands. Thus it is possible to generate a picture on our PDP-11 and send the elements via the fast data link to the Triplex, where the picture could be drawn on some other device, for example, microfiche. The implementation of new devices involves only a minimal effort.

Histogramming.

The HPAK and DPAK packages (3), developed at SLAC, are used for all histogramming. Any task can define spectrum accumulation areas in its own task space. Simple subroutine calls define and provide access to the spectra. Every 'n'th event, where 'n' is a definition parameter, the spectra are written to a disk file, where they are accessible by the display package. The display routines fetch a spectrum by number (each task must use different numbers) and display it on one of several devices (Tektronix 4013, SLAC scope, or Versatec), again using Unified Graphics. The flexibility provided by this modular structure makes the system extremely useful.

Monitor Programs.

Several tasks are run in the RSX-11M reschedule mode to monitor various aspects of the electronics, keep track of the beam energy, to run the Xenon flashers, which serve to monitor the crystals in the ball, etc.

Realtime Link

An IBM System/7 provides the interface to the SLAC Triplex Realtime System (5), a multitasking network which routes "messages" and data between user-defined "nodes", which can be Fortran programs running in the Triplex or tasks in the System/7 or PDP-11. The realtime system has demonstrated its usefulness often, especially when complicated analysis is required immediately, during an energy scan. The system can also be used to transfer programs or other data from one computer to the other. The data link is capable of transferring up to 200 kByte/sec.

## SUCCESSES TO DATE AND FUTURE PLANS

The Crystal Ball experiment was installed at SPEAR in October, 1978. One week later the first data tape was written. All experimenters now work easily with the system.

Several important physics results were obtained from the initial data (at Psi and Psi-prime energies).

Our future plans include obtaining a VAX which we would use as a backend machine, as well as for preliminary data reduction. This would produce statistically better immediate results as well as relieve the load at the SLAC Triplex.

## BIBLIOGRAPHY

1) G.I. Oxoby, A Fast LAM Handler for the PDP-11
C.B. Note no. 114, June 28, 1976
SLAC, Stanford

2) D. Horelich, "A Simple, Versatile CAMAC Crate Controller and Interrupt Priority Encoding Module"
IEEE Trans. Nucl. Sci.
NS22, 517(1974)

3) DPAK and HPAK, A Versatile Display and Histogramming Package,
C.A. Logg, A.M. Boyarski, A.J. Cook, R.L.A. Cottrell,
SLAC Report No. 196
June 1976

4) RSX-11M v3.1 Vol 1A
System Reference Information
Digital Equipment Corporation

5) Ilse Denecke, Introduction to SLAC's Real-Time System Sept. 1974
SLAC/SCIP Menlo Park, Calif

6) G. Oxoby and J. Bernstein,
"ARBOLA"-SLAC Branch Driver for PDP-11,
Crystal Ball Note no. 113
January 31, 1977

7) Robert Beach, The SLAC Unified Graphics System Programming Manual
SLAC, Stanford, Calif.
Jan. 1978